Afstudeerverslag

Making Offers Intelligent

De Haagse Hogeschool HBO-ICT SE AFST P4-P1
Unchained Carrot
Den Haag
Oktober 2020

Eerste Contactpersoon UCC
De heer Walter Thoen
Functie: CTO

Eerste contactpersoon HHS
De heer Ed van Doorn
Functie: Begeleidend examinator

Tweede Contactpersoon UCC
De heer Mark Noorlander
Functie: CEO

Tweede contactpersoon HHS
De heer Maarten Thissen
Functie: Expert examinator

Sohaib Ahmad

Voorwoord

Dit is het afstudeerverslag ‘Making Offers Intelligent’. Ik heb dit verslag geschreven in het kader van het afstuderen aan de opleiding Software Engineering aan De Haagse Hogeschool. Ik heb van 11 mei tot 5 oktober bij Unchained Carrot gewerkt aan een project om aan te tonen dat er een mogelijkheid is, om met Machine Learning de effectiviteit te verhogen van e-mail marketing aanbiedingen.

Hierbij wil ik mijn begeleiders bedanken. Dit zijn Walter Thoen en Mark Noorlander vanuit Unchained Carrot en Ed van Doorn en Maarten Thissen vanuit De Haagse Hogeschool. Ook wil ik de bedrijven die mee hebben gedaan aan het experiment bedanken voor hun medewerking.

Den Haag, Augustus 2020

Sohaib Ahmad

Referaat

Dit verslag beschrijft het proces dat ik gedurende 17 weken heb doorlopen tijdens het uitvoeren van mijn afstudeeropdracht bij Unchained Carrot vanuit huis. De afstudeerperiode is op 11 mei 2020 gestart en de einddatum is 5 oktober.

Het verslag behandelt het onderzoeken en ontwikkelen van een proof-of-concept die het mogelijk maakt om met Machine Learning effectievere e-mail marketing promoties te houden.

Descriptoren:

- Machine Learning
- XGBoost
- Python
- Node.js
- AWS
- E-mail Marketing
- Conversie
- ActiveCampaign
- Mailchimp
- AWS SageMaker

Begrippenlijst

Conversie: een meetbare actie die wordt gezien als waardevol. Bijvoorbeeld, het doen van een aankoop. In dit verslag wordt meestal het beantwoorden van een enquête via e-mail bedoeld.

Conversiepercentage: het percentage van de conversies in verhouding tot het totaal aantal ontvangers. Als 100 mensen een enquête ontvangen en 10 geven een antwoord, dan is het conversiepercentage 10%.

Kosten per conversie: de kosten die gemaakt zijn voordat er een conversie heeft plaatsgevonden. Als er voor €500 ergens reclame wordt gemaakt, en dit levert 10 conversies op, dan zijn de kosten per conversie €50

Openingspercentage: het percentage van de ontvangers van een e-mail die deze hebben geopend. Als 100 mensen een e-mail hebben ontvangen en 50 openen deze, dan is het openingspercentage 50%.

Feature: ook wel variabele genoemd. Input voor het Machine Learning model. Het is een kolom in de dataset. Voorbeelden: kleur, lengte, leeftijd.

Dataset: in dit verslag wordt hiermee een tabel met data bedoeld.

Trainingsset: de dataset waarmee het Machine Learning model traint of in andere woorden: leert.

Test set: de dataset die het Machine Learning model niet heeft gezien tijdens het leren. Deze wordt gebruikt om de kwaliteit van de voorspellingen van het model te meten.

ML model: afkorting voor Machine Learning model.

Jupyter Notebook: de software waarin ik het Machine Learning model heb geprogrammeerd.

AWS SageMaker: een omhulsel voor Jupyter Notebooks. Ik heb AWS SageMaker gebruikt om Jupyter Notebooks te creëren.

AWS Lambda: een service van AWS die het mogelijk maakt om online code te runnen zonder servers.

E-mail marketing provider: een software oplossing die bedrijven de mogelijkheid geeft om e-mails te versturen naar haar klanten.

UNCHAINED CARROT

Inhoudsopgave

1 Inleiding

In het laatste jaar van de opleiding HBO-ICT Software Engineering heb ik een meesterproef gekregen om te testen of ik klaar ben om mijn vaardigheden te gebruiken op de werkvloer. Deze meesterproef is de afstudeeropdracht. Ik heb gewerkt aan mijn afstudeeropdracht in periode 4 en 1, dus van 11 mei 2020 tot 5 oktober 2020 bij het bedrijf Unchained Carrot. Unchained Carrot is gevestigd in Rotterdam, Rijswijk en Milan (Italië) en maakt SaaS oplossingen voor marketing.

Mijn opdracht bestond uit twee verschillende onderdelen: onderzoek en software ontwikkelen. Ik ben begonnen met het onderzoeken hoe het mogelijk is om met Machine Learning de effectiviteit van e-mail marketing promoties te verhogen. Vervolgens heb ik hier een proof-of-concept voor ontwikkeld die de theorie en hypotheses toetst in de praktijk.

Mijn bedrijfsmentor was Walter Thoen. Hij is de CTO van Unchained Carrot en heeft al meer dan 20 jaar ervaring op het vakgebied. Naast Walter, ben ik ook begeleid door de CEO van Unchained Carrot Mark Noorlander. Mark is een multi-ondernemer en heeft meerdere succesvolle projecten en bedrijven opgericht in Nederland en Italië.

Mijn begeleidende examinator vanuit de Haagse Hogeschool was meneer Ed van Doorn. Meneer Ed van Doorn is afkomstig uit de opleiding en is mijn eerste aanspreekpunt. Ik ben door meneer Ed van Doorn begeleid bij het beoordeelbaar maken van mijn afstudeerwerk.

Mijn expert examinator was meneer Maarten Thissen. Meneer Maarten Thissen is in staat een deskundig oordeel te vellen over mijn HBO vaardigheden. De expert examinator is verantwoordelijk voor de kwaliteit van de beoordeling van mijn afstudeerwerk en heeft daarom meer afstand tot mij gehouden.

Het verslag begint met de context omschrijving waarin ik meer informatie geef over het bedrijf waar ik de opdracht heb uitgevoerd. Daarna heb ik het over mijn initiële aanpak, gevolgd door mijn daadwerkelijk werkzaamheden. Ten slotte reflecteer en evalueer ik op het uitgevoerde proces en de opgeleverde producten.

2 Unchained Carrot

In dit hoofdstuk geef ik de lezer een beeld van de organisatie waar ik mijn afstudeeropdracht heb uitgevoerd.

2.1 Missie

De missie van Unchained Carrot is om marketeers de mogelijkheid te bieden om real-time hyper-gepersonaliseerde beloningen te maken.

2.2 Omgeving

Unchained Carrot is een relatief nieuw softwarebedrijf opgericht in 2019 door het team van het softwarebedrijf Flashboys. Er zijn minstens twaalf personen actief in Unchained Carrot. De onderneming is gevestigd in het dorp Zuidland en is actief vanuit verschillende locaties, zoals Rotterdam, Rijswijk en Milaan (Italië). Het bedrijf werkt naast vanuit kantoor veel remote. Met de huidige Covid-19 situatie wordt vrijwel al het werk vanuit huis verricht en dit blijft voorlopig nog zo.

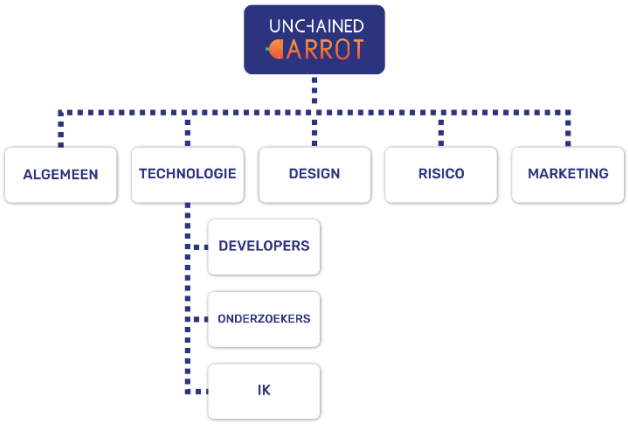
2.3 Beschrijving

Unchained Carrot biedt SaaS oplossingen voor reward-based marketing. Er wordt gebruik gemaakt van gedecentraliseerde technologieën om marketingactiviteiten effectiever te maken. Unchained Carrot verandert marketing mechanismen door bedrijven in staat te stellen real-time, hyper-gepersonaliseerde beloningen en promoties te creëren. Het is de volgende stap in marketing en de volgende generatie marketingoplossingen. De oplossingen worden gecreëerd met behulp van blockchain-experts, data-analisten en marketeers.

2.4 Bedrijfsmodellering

De stand van zaken rond bedrijfsmodellering bij Unchained Carrot zijn voornamelijk informeel. Processen worden vaak mondeling via een telefoontje of via WhatsApp overgebracht, maar er worden ook formele tools, zoals Azure DevOps gebruikt.
De stand van zaken zijn vooral formeel met remote werkers in bijvoorbeeld Ghana en met klanten.

Unchained Carrot heeft een hiërarchisch structuur. Er is een managementteam. Daaronder de developers, researchers en ik. In de organisatie zijn er geen silo's. De verschillende afdelingen, zoals Design en Development werken nauw met elkaar en met mij samen.



Figuur 1: Organogram Unchained Carrot

3 Making Offers Intelligent

In dit hoofdstuk geef ik de lezer een beeld van mijn opdracht. Ik beschrijf alles wat de lezer nodig heeft of moet weten om de opdracht te begrijpen en maak duidelijk hoe mijn opdracht samenhangt met de doelen van de organisatie waar ik mijn opdracht heb uitgevoerd.

3.1 Probleemstelling

De afgelopen jaren wordt er steeds meer gesproken over Machine Learning in verschillende industrieën. In de marketingwereld is het zo een populair onderwerp dat Gartner "Artificial Intelligence for Marketing" op de peak heeft geplaatst in de 2019 hype cyclus. Bedrijven die actief zijn in de marketingwereld kunnen dit niet negeren. Unchained Carrot is een van deze bedrijven. Naar aanleiding hiervan is Unchained Carrot op zoek naar manieren om AI te gebruiken in haar SaaS-oplossingen.

Echter, is dit niet makkelijk. Het is bijna onmogelijk om goede voorbeelden te vinden over hoe andere organisaties daadwerkelijk Machine Learning gebruiken voor marketing. Velen artikelen leggen uit wat ze doen, maar niet hoe en zijn in feite een marketingmiddel voor deze organisaties. Unchained Carrot heeft nu de volgende vraag die zij beantwoord wil zien hebben:

Hoe kan Machine Learning de optimale prikkel per klant voorspellen en de effectiviteit van e-mail marketing promoties verhogen?

In het algemeen geven bedrijven tijdens een promotie iedereen dezelfde beloning. Het probleem hierbij is dat het niet bekend is of deze beloning wel genoeg is om te leiden tot een aankoop en of misschien de klant toch al van plan was om een aankoop te plaatsen. Er wordt dan zomaar bijvoorbeeld een korting van 10% gegeven en omzet misgelopen. Een enorme korting, zoals 90% is een zeer sterke prikkel dat waarschijnlijk zal leiden tot een aankoop, maar het bedrijf maakt dan weinig of geen winst. In dit voorbeeld gaat het over kortingspercentages, maar de beloningen kunnen ook bijvoorbeeld punten of een donatie naar een goed doel zijn.

Er is een optimum dat de meeste winst oplevert. Om dit optimum te vinden moet er data vanuit verschillende bronnen worden verzameld en geanalyseerd.

Dit kan gedaan worden door een mens, maar er is een limiet aan hoeveel en hoe snel patronen door een persoon in data kunnen worden gevonden. Een machine kan dit veel snellerkg. Het zijn in feite calculaties op de data. Een rekenmachine doet het al sneller dan een mens.

3.2 Doelstelling

Het doel van deze opdracht is om antwoord te geven op de vraag: hoe kan Machine Learning de optimale prikkel per klant voorspellen en de

effectiviteit van e-mail marketing promoties verhogen? Dit ga ik doen met behulp van deskresearch, gevolgd met het ontwikkelen van een proof-of-concept. Het doel van deskresearch is om met bestaande literatuur te onderzoeken wat andere gedaan hebben met soortgelijke problemen. Het resultaat gaat invloed hebben op mijn aanpak. Het doel van het proof-of-concept is om het minimale systeem te bouwen die de hoofdvraag toetst in de praktijk. De proof-of-concept is dus een software oplossing die gebruik maakt van Machine Learning om de optimale prikkel per klant te bepalen en hiermee de effectiviteit van promoties verhoogt. Het is het minimale wat ontwikkeld moet worden om de hoofdvraag te kunnen toetsen.

3.3 Resultaat

Wanneer de afstudeeropdracht met succes is afgerond, krijgt Unchained Carrot een onderzoeksrapport en een proof-of-concept van het systeem en kan deze gebruiken om haar SaaS-oplossingen te verbeteren en meer waarde te leveren aan haar klanten. Het proof-of-concept bestaat uit het ML model en de omliggende software-integraties die ik ervoor ga bouwen.

3.4 Afbakening

Tijdens mijn opdracht ga ik een proof-of-concept ontwikkelen. Dit houdt in dat de software oplossing tot zo ver is ontwikkeld dat het de hoofdvraag kan toetsen. Als ik met het systeem, de hoofdvraag kan toetsen, dan heb ik voldaan aan de eisen van de opdrachtgevers en ben ik klaar met de opdracht.

4 Oriëntatie

Voordat ik daadwerkelijk kon beginnen aan mijn opdracht moest ik mij eerst oriënteren op het bedrijf, de initiële requirements achterhalen en mijn plan van aanpak opstellen. Deze fase noem ik de oriëntatiefase.

4.1 Oriëntatieweek

Voordat ik ben begonnen met het opstellen van het plan van aanpak ben ik een week aan de slag gegaan met een aantal oriënterende activiteiten. Hierin had ik de huidige situatie, probleem- en doelstelling, en de requirements helder in kaart gebracht. Het doel van de oriëntatieweek was om genoeg kennis op te doen om een krachtig plan van aanpak op te kunnen stellen.

Ik had een presentatie voorbereid over mijn opdracht en mijn visie ervoor en deze gepresenteerd aan het technische team van Unchained Carrot (Technologie afdeling, zie Figuur 1: Organogram Unchained Carrot). Dit team bestaat uit mijn twee begeleiders, onderzoekers met een PhD en developers. Na de presentatie hadden we gediscussieerd over de requirements van mijn opdracht en hier kom ik later op terug (zie hoofdstuk 6 voor de requirements).

4.2 Wijze van communicatie

Aangezien ik vanuit huis werkte, hadden we afgesproken om dagelijks een update te sturen via WhatsApp om af te stemmen waarmee iedereen bezig was. Daarnaast hadden we afgesproken om op maandag en donderdag een meeting van een uur te houden met het technische team. In deze meetings zal o.a. de voortgang van mijn opdracht worden behandeld.

Voor mijn vrije dagen had ik afgesproken om deze in lijn te nemen met de vrije dagen van mijn begeleiders. Als bijvoorbeeld mijn begeleider in augustus een weekje op vakantie zal gaan, dan zal ik mijn vakantie ook in dezelfde week plannen. Echter, had niemand plannen. Vandaar dat ik al mijn vrije dagen helemaal aan het einde had ingepland. Dit hield in dat ik ongeveer een maand voor de inleverdatum van 5 oktober al klaar zal zijn met mijn opdracht.

4.3 Plan van aanpak

Nadat ik mijn oriënterende activiteiten had afgerond was ik begonnen aan mijn plan van aanpak. Het doel van het plan van aanpak was om de uitvoering van mijn project te plannen. Het moest structuur en visie geven aan de opdracht. Om dit plan op te stellen had ik gebruik gemaakt van de visie document template die beschikbaar was gesteld door de Haagse Hogeschool. Ik had hiervoor gekozen, aangezien dit tijdens de opleiding voor software projecten was gebruikt.

Echter, was dit alleen niet genoeg, omdat mijn opdracht meer inhield dan software ontwikkelen. Het was een onderzoeksopdracht waar het

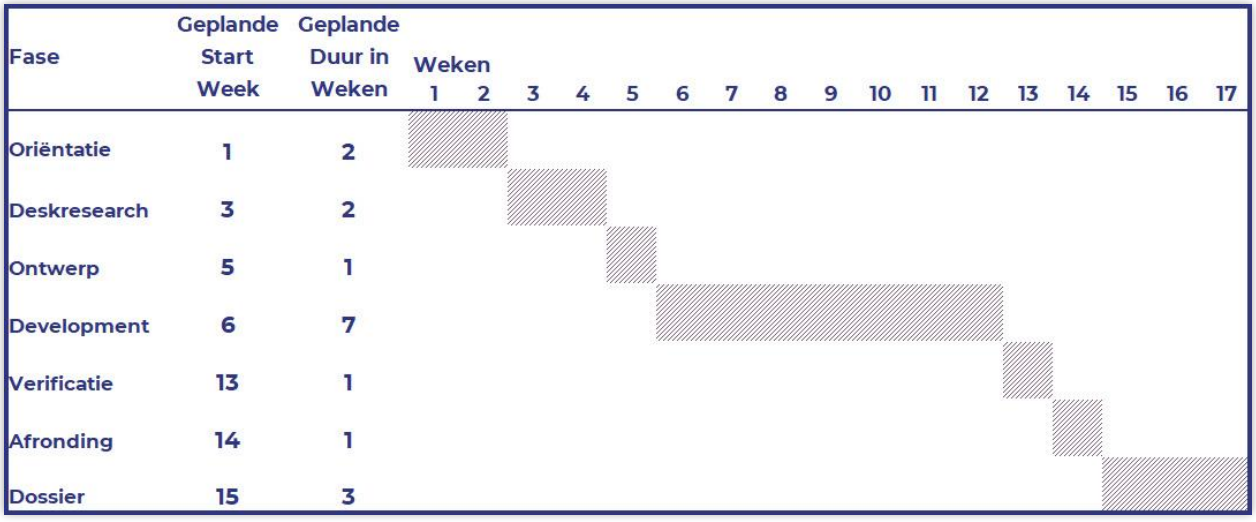
ontwikkelen van software een onderdeel van was. Daarom was ik verder gaan zoeken naar andere technieken en sjablonen.

Voor mijn derdejaars stage had ik gebruik gemaakt van een sjabloon voor het opstellen van een plan van aanpak dat was beschikbaar gesteld door Scribbr. Deze was destijds goedgekeurd door mijn begeleiders, vandaar dat ik deze weer had gebruikt.

4.3.1 Planning

In de globale planning voor het project had ik het project opgedeeld in 6 fases: oriëntatie, deskresearch, ontwerp, development, verificatie, en afronding. Deze fasering was gebaseerd op de Watervalmethode, zoals behandeld in blok 11 IT Operations. Ik had aanpassingen aan de fasering gemaakt, zodat het beter bij mijn opdracht paste. Waarom ik voor de Watervalmethode had gekozen is te lezen in paragraaf 8.1.1 Software development methode. Als laatste had ik nog 3 weken gereserveerd om te werken aan mijn afstudeerdossier. Om structuur in het verslag te behouden heeft elke fase (behalve "Dossier") een apart hoofdstuk.

Voor het opstellen van de globale planning van mijn project had ik gebruik gemaakt van een Gantt chart en dit had ik gedaan in Excel. Van de verschillende methodes die ik had onderzocht om projecten te plannen vond ik een Gantt chart het meest overzichtelijk. Ik had verschillende software oplossingen voor het maken van Gantt charts onderzocht en had uiteindelijk gekozen voor Excel. Dit had ik gedaan, omdat het gratis en makkelijk te gebruiken was.



Figuur 2: Gantt chart met de fasen van het project

In hoofdstuk 8 Development wordt de fasering voor de development-fase verder uitgewerkt. Het volledige plan van aanpak is te vinden in de "Plan of Action" bijlage.

4.3.2 Risicoanalyse

Voor het opstellen van de risicoanalyse had ik de methode gebruikt die tijdens de opleiding in blok 11 was behandeld. Dit had ik gedaan, aangezien ik deze al meerdere keren succesvol had gebruikt.

Bedreiging	***Kans***	***Impact***	***Risico***	***Tegenmaatregel***	***Risico eigenaar***
Technisch					
TR1: *De juiste apparatuur ontbreekt*	*1*	*4*	*4*	***Reactief:*** *Gebruik maken van Cloud oplossingen*	*Ik*
Organisatorisch					
OR1: *Ik heb niet genoeg kennis*	*3*	*5*	*15*	***Preventief:*** *Oriënteren* ***Reactief:*** *Kennis aanvullen, bijvoorbeeld vragen aan collega's.*	*Ik*
OR2: *Er wordt geen proces gevolgd*	*3*	*3*	*9*	***Preventief:*** *Kies een proces dat past bij het project* ***Reactief:*** *evalueer en pas de werkwijze aan*	*Ik*
OR3: *Niet genoeg data*	*3*	*4*	*12*	***Reactief:*** *Verzamel meer data*	*Ik*
OR4: *Slechte communicatie*	*3*	*4*	*12*	***Preventief:*** *Afspraken maken*	*Ik*
Functioneel					
FU1: *Het is niet duidelijk wat het systeem moet kunnen*	*2*	*4*	*8*	***Preventief:*** *Interviews houden* ***Reactief:*** *Reviews met stakeholders*	*Ik*

Figuur 3: Risicoanalyse

5 Deskresearch

Deskresearch was de fase waarin in op zoek was gegaan naar bestaande informatie. Ik had literatuur onderzocht, video's bekeken en online cursussen gevolgd met betrekking tot Machine Learning en marketing. Het doel van deze fase was om met bestaande literatuur te onderzoeken wat andere gedaan hadden met soortgelijke problemen, en mij voor te bereiden op de development-fase.

5.1 Onderzoeksvragen

De hoofdvraag had ik gekregen van het afstudeerbedrijf. Voor het formuleren van mijn deelvragen had ik gebruik gemaakt van de literatuur die we tijdens de opleiding in het blok Exploring The World of Research hadden behandeld. Aangezien het een bron van school was, wist ik zeker dat het betrouwbaar en op HBO niveau was. Dit zal ook tijd besparen, die ik anders had moeten besteden aan het kritisch onderzoeken van andere methodes. Gezien de korte tijd, ging mijn voorkeur naar de eerste optie.

Hoofdvraag

Hoe kan Machine Learning de optimale prikkel per klant voorspellen en de effectiviteit van e-mail marketing promoties verhogen?

Deelvragen

1. Wat is Machine Learning?
2. Hoe kan Machine Learning worden geïntegreerd in traditionele software oplossingen?
3. Hoe kan de conversie van e-mail marketing worden verhoogd zonder Machine Learning?
4. Welke en hoeveel data moet worden verzameld?
5. Welke algoritmes moeten worden gebruikt voor het Machine Learning-model?

In vraag 3 komt het begrip conversie naar voren. Een conversie is een meetbare actie die wordt gezien als waardevol. Bijvoorbeeld, het doen van een aankoop. In dit verslag wordt meestal het beantwoorden van een enquête via e-mail bedoeld.

5.2 Onderzoeksaanpak

Tijdens de deskresearch had ik voornamelijk kwalitatief onderzoek uitgevoerd. De meeste deelvragen waren beschrijvend van aard en konden worden beantwoord met woorden. Deze had ik beantwoord met beschrijvend onderzoek. Ik had literatuur, cursussen, bestaande datasets en video's bestudeerd, en hiermee de onderzoeksvragen beantwoord. Deelvraag 4 kon worden beantwoord met cijfers. Deze had ik geprobeerd te beantwoorden met kwantitatief onderzoek waar ik op zoek was gegaan naar bestaande datasets die werden gebruikt voor Machine Learning.

5.2.1 Literatuur onderzoek

Literatuuronderzoek had ik verricht om theoretische kennis op te doen en begrip te krijgen op Machine Learning en marketing. Ik had literatuuronderzoek op een gestructureerde wijze uitgevoerd. Voor deze aanpak had ik gebruik gemaakt van de literatuur die tijdens de opleiding in het blok Exploring The World of Research was bestudeerd.

Kernwoord selectie

De eerste stap was om een aantal kernwoorden te selecteren die ik in de volgende stap zal gebruiken om bronnen te zoeken. De kernwoorden kwamen uit mijn probleemstelling.

Op zoek naar bronnen

Ik had daarna Google gebruikt om zoveel mogelijk bruikbare bronnen te zoeken. Ik had voor Google gekozen, omdat Google de meeste resultaten vond vergeleken met Bing, Yahoo en DukcDuckGo. De bronnen die ik via Google had gevonden, konden worden onderverdeeld in de volgende 3 categorieën: artikelen, video's en cursussen.

Bron selectie

De volgende stap was om mijn selectie van bronnen te verkleinen naar de meest relevante. Hiervoor had ik onder andere de volgende criteria gebruikt:

- De bron moet niet ouder dan 2015 zijn. Aangezien Machine Learning zo snel innoveert, is het van cruciaal belang dat alleen recent gepubliceerde bevindingen worden geraadpleegd om de validiteit van het onderzoek te verzekeren.
- Het doel van de bron moet zijn om de lezer te informeren over het onderwerp. Veel artikelen op het gebied van Machine Learning en marketing hebben als doel om te verkopen. Het artikel gaat bijvoorbeeld over een Machine Learning oplossing en is geschreven door het bedrijf die deze oplossing aanbiedt. Hoogstwaarschijnlijk heeft het geen neutrale positie en is geschreven voor eigen belang. Het artikel is minder betrouwbaar.
- De bron moet zijn geschreven door een deskundige. Het moet bijvoorbeeld geen artikel zijn die is geschreven door een algemene journalist, aangezien het dan niet op de details ingaat.

Data verwerking

De volgende stap was om de informatie te verwerken. Van de artikelen en video's had ik samenvattingen gemaakt en deze verwerkt in mijn onderzoeksrapport. De cursussen had ik bestudeerd en de opdrachten ervan uitgevoerd. Ik had deze verschillende categorieën van bronnen niet los van elkaar verwerkt, maar tegelijkertijd. Ik had bijvoorbeeld de dag ingedeeld in drie delen, in de eerste verwerkte ik de artikelen, in de tweede de video's en in de derde de cursussen. Met deze manier kon ik sneller verbanden leggen tussen de verschillende bevindingen.

5.3 Conclusie

In deze paragraaf beantwoord ik kort de deelvragen en tenslotte de hoofdvraag. De volledige antwoorden zijn te vinden in de "Desk Research" bijlage.

Wat is Machine Learning?

Machine Learning is het proces waarbij machines het vermogen krijgen om te leren en zich aan te passen vanuit ervaring (Google, 2020).

Hoe kan Machine Learning worden geïntegreerd in traditionele software oplossingen?

Machine Learning modellen kunnen via API endpoints worden geïntegreerd met andere software oplossingen. Via een endpoint kan andere software een voorspelling opvragen aan een ML model.

Hoe kan de conversie van e-mailmarketing worden verhoogd zonder Machine Learning?

Een van de meest veel belovende methodes om de conversie van e-mailmarketing te verhogen zonder Machine Learning is het personaliseren van e-mails (Smart Insights, 2019; Prakash, 2020; Mialki, 2020). De hoogste converterende e-mails zijn zo gepersonaliseerd dat het 1-op-1 e-mails lijken. Ze zijn zo opgesteld dat het lijkt dat ze door een medewerker, zoals de directeur van het versturende bedrijf, speciaal en alleen voor een bepaalde klant zijn geschreven (Smart Insights, 2019).

Het personaliseren van e-mails kan op meerdere manieren gedaan worden. Een marketing e-mail is opgebouwd uit 3 delen: het onderwerp, het verhaal (tekst in de e-mail) en het aanbod (een aanbod zoals een korting). Ieder moet aanwezig zijn voor een succesvolle campagne. (Brunson, 2018). Deze onderdelen kunnen gepersonaliseerd worden. In het onderwerp of het verhaal kan bijvoorbeeld de naam van de klant worden vermeld. Het aanbod kan bijvoorbeeld gebaseerd zijn op de historische aankopen van de klant. Het tijdstip van verzenden kan ook worden gepersonaliseerd, door deze af te stemmen op eerder gemeten data van de klant.

Welke en hoeveel data moet worden verzameld?

Er zijn twee vormen van Machine Learning die relevant zijn voor deze opdracht: Supervised Learning en Unsupervised Learning. Voor Supervised Learning is gelabelde data nodig. Dit houdt in dat de mogelijke opties van de te voorspellen uitkomst al bekend zijn. Voor Unsupervised Learning is dit niet nodig. Een voorbeeld van Supervised Learning is om dieren te categoriseren onder bijvoorbeeld vissen, vogels en insecten. Bij Unsupervised Learning zijn deze uitkomsten niet bekend, en maakt het ML model vanzelf de verschillende categorieën. De uitkomsten zijn dan bijvoorbeeld groep 1, groep 2, groep 3 en zo verder. Het nadeel is dat het nog steeds niet bekend is, tot welke van de gelabelde groepen (vissen, vogels en insecten) een bepaald dier hoort en deze label moet dan zelf aan de groep worden

toegekend. Een mogelijk pluspunt is dat het model categorieën die niet bekend waren kan ontdekken.

Er is weinig data die noodzakelijk is om een ML model te trainen. Voor Supervised Learning zijn 2 (waarvan 1 de uitkomst is) eigenschappen en voor Unsupervised Learning 1 eigenschap al genoeg. Of de voorspellingen nauwkeurig gaan zijn, is een andere vraag. Om de juiste eigenschappen te bepalen voor nauwkeurige voorspellingen moet een data analyse worden uitgevoerd wat een kernonderdeel is van de Machine Learning Development Lifecycle. De bijlagen "XGBoost Mailchimp Feature Importance" en "XGBoost Mailchimp Example Tree" weergeven de eigenschappen die het ML model belangrijk heeft gevonden voor het maken van voorspellingen. Deze bijlagen zijn opgesteld tijdens de werkzaamheden van paragraaf 8.2.5 ML model Iteratie #4.

Er zijn 3 categorieën van data die een kern rol kunnen spelen bij het personaliseren van beloningen. Eigenschappen van de klant zoals demografische data (Al-Zuabi, Jafar en Aljoumaa, 2019); eigenschappen van het bedrijf zoals hoe lang het bedrijf bestaat (Song & Yang, 2018); en eigenschappen van de beloning zoals het kortingspercentage (Song & Yang, 2018).

De data kan worden verzameld met directe methoden, zoals enquêtes en via indirecte methoden, zoals het bijhouden van de klantactiviteit op websites met behulp van "tracking pixels". Er kan bijvoorbeeld met een enquête achterhaald worden wat iemands favoriete product is, maar deze informatie kan ook worden achterhaald door de klantactiviteit te meten en analyseren. Er kan op de website worden bijgehouden dat iemand steeds een bepaalde pagina van een product bezoekt of dat iemand een bepaald product vaak koopt, dus waarschijnlijk is dit een favoriet. Daarnaast kunnen gegevens ook worden verkregen uit bestaande systemen van derden die een bedrijf al gebruikt, zoals een CRM-systeem.

De vraag "Hoeveel?" kon niet worden beantwoord met een getal. In het algemeen geldt dat meer data beter is (Google, 2020). Echter, is het niet de bedoeling om eeuwig lang bezig te zijn met dataverzameling. Een aangeraden aanpak is om een kleine hoeveelheid data te verzamelen en hiermee aan de slag te gaan (Brownlee, 2017). De resultaten die met behulp van deze data worden behaald, kunnen dan de vervolgstappen met betrekking tot dataverzameling bepalen.

Welke algoritmes moeten worden gebruikt voor het Machine Learning-model?

Er is geen algoritme waarvan van tevoren bekend is dat deze het beste resultaat gaat opleveren (Kharkovyna, 2019). Wel kan de selectie van algoritmes worden verkleind op basis van het probleem (Brownlee, 2019).

Een Linear Regression algoritme kan worden gebruikt bij een regressie probleem, zoals het voorspellen van de aandelenkoers. Decision Trees

kunnen worden gebruikt om classificatie problemen op te lossen, zoals het voorspellen van de kleur van een vogel.

Neurale netwerken werken het beste met grote hoeveelheden ongestructureerde data, zoals foto's en video's en presteren slechter dan traditionele algoritmes op kleinere gestructureerde gegevens (Brownlee, 2019).

Daarnaast kan de techniek genaamd Spot Checking worden gebruikt om tot een uiteindelijke algoritme te komen (Brownlee, 2020). In dit proces worden snel verschillende algoritmes getest om erachter te komen welke effectief zijn, zodat daarop gefocust kan worden.

Uiteindelijk had ik met behulp van Spot Checking gekozen voor de XGBoost (eXtreme Gradient Boosting) algoritme en hier is meer over te lezen in 8.2.5 ML model Iteratie #4.

Hoe kan Machine Learning de optimale prikkel per klant voorspellen en de effectiviteit van e-mail marketing promoties verhogen?

De effectiviteit van e-mail marketing promoties kan worden verhoogd door de e-mails te personaliseren. Met een hogere effectiviteit wordt het verhogen van het rendement op investering bedoeld. Het personaliseren van e-mails zal in theorie bijvoorbeeld een hogere conversie en of een lager kosten per conversie kunnen opleveren. De vraag is nu of Machine Learning de taak van personaliseren succesvol over kan nemen van een mens.

Er was een aantal manieren om e-mails te personaliseren, maar voor mijn opdracht waren de beloningen het meest relevant, aangezien Unchained Carrot zich bezig houdt met beloningen.

Er waren twee manieren die ik had bedacht om deze beloning te personaliseren met Machine Learning. De eerste was om een minimale beloning te bepalen waar vervolgens extra punten aan werden toegevoegd afhankelijk van de klant. Hier was nog input van een mens nodig voor het bepalen van de minimale beloning. Het voordeel is dat minder data nodig is.

De tweede optie was om het ML model helemaal zelf de beloning te laten kiezen afhankelijk van de klant. Het probleem hiermee is dat de beloningen al een keer in het verleden moeten zijn aangeboden. Dus er is meer data nodig wat niet altijd zo makkelijk is te verkrijgen.

5.4 Vervolgonderzoek

Ik had drie hypotheses geformuleerd op basis van de conclusie en theorie. Om deze te toetsen, was ik van plan een experiment uit te voeren waarvoor ik een software prototype zal moeten ontwikkelen. Dit wordt behandeld in de development-fase. De hypotheses zijn hieronder opgesomd:

- Het verhogen van de beloning leidt tot een verhoging van het aantal conversies en dit leidt tot een verhoging van de omzet, maar niet noodzakelijk de winst.
- Het gebruik van ML voorspelde beloningen leidt tot een verhoging van het aantal conversies en dit leidt tot een verhoging van de omzet.
- Het gebruik van ML voorspelde beloningen leidt tot een lagere kosten per conversie en dit leidt tot een verhoging van de winst.

6 Requirements

De requirements van dit project zijn achterhaald vanuit 2 bronnen. Er is een aantal requirements opgesteld als gevolg van de wekelijkse gesprekken met de stakeholders en er is een aantal requirements opgesteld als gevolg van deskresearch.

Ik had elke week twee meetings met mijn stakeholders. Tijdens deze meetings had ik elicitatie technieken gebruikt om de requirements te achterhalen.

Vervolgens zijn er nog requirements die ik had opgesteld met behulp van deskresearch. Dit zijn requirements, waaraan een Machine Learning systeem moet voldoen volgens de theorie. Bijvoorbeeld R17, als deze niet wordt voldaan, kan het model niet trainen en geeft het een error.

De requirements zijn geprioriteerd met behulp van de MoSCoW (Must, Should, Could Won't) methode. Van de methodes die ik had vergeleken, vond ik dat bij deze methode de betekenis van de prioriteit het duidelijkst was. Er wordt geen gebruik gemaakt van User Stories aangezien, er geen gebruikers centraal staan in dit project. Hieronder volgt een tabel met de requirements gegroepeerd op bron.

ID	Requirement	Prioriteit	Functioneel?	Bron
R1	Het systeem maakt gebruik van Machine Learning	Must	Nee	Gesprek stakeholders
R2	Het systeem maakt gebruik van AWS	Must	Nee	Gesprek stakeholders
R3	Het systeem maakt gebruik van de e-mail marketing kanaal	Must	Nee	Gesprek stakeholders
R4	Het systeem is schaalbaar	Should	Nee	Gesprek stakeholders
R5	Het systeem genereert beloningen real-time	Should	Ja	Gesprek stakeholders
R6	Het systeem geeft binnen 2 seconden een gegenereerde beloning weer	Should	Ja	Gesprek stakeholders
R7	De Reward Generator is ontwikkeld in Node.js	Must	Nee	Gesprek stakeholders
R8	De Reward Generator genereert SVG plaatjes	Should	Ja	Gesprek stakeholders
R9	Het systeem heeft een user interface	Won't	Nee	Gesprek stakeholders
R10	Het Machine Learning model kan integreren met andere software	Must	Nee	Gesprek stakeholders
R11	Het systeem kan beloningen in e-mails personaliseren	Must	Ja	Gesprek stakeholders
R12	Het systeem haalt de data op van ReviewByMe	Should	Ja	Gesprek stakeholders
R13	Het systeem haalt de data op van Mailchimp	Should	Ja	Gesprek stakeholders

R14	Het Machine Learning model is gebouwd in AWS SageMaker	Must	Nee	Deskresearch
R15	Het Machine Learning model traint met een gebalanceerd dataset	Should	Nee	Deskresearch
R16	Het Machine Learning model maakt gebruik van demografische gegevens	Should	Ja	Deskresearch
R17	Het Machine Learning model verandert non-numerieke waardes naar numeriek	Must	Ja	Deskresearch
R18	Het Machine Learning model normaliseert numerieke waardes in de dataset	Should	Ja	Deskresearch
R19	Het Machine Learning model plaatst numerieke waardes in de dataset in bins	Should	Ja	Deskresearch
R20	Het Machine Learning model heeft een hogere accuracy score dan een heuristische oplossing	Should	Nee	Deskresearch

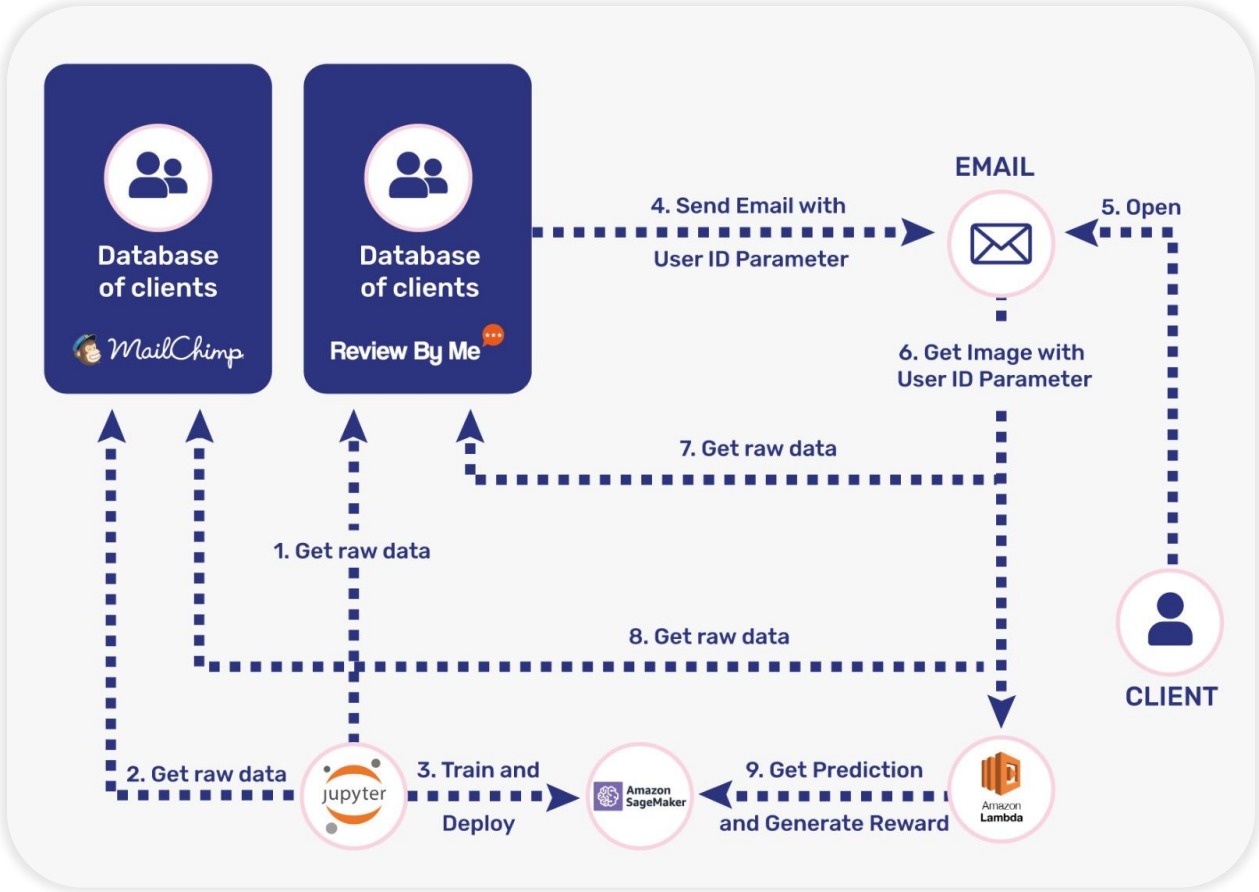
Figuur 4: Tabel met de requirements

De betekenissen van de requirements R17, R18, R19 en R20 worden duidelijk met de context in hoofdstuk 8 Development.

7 Ontwerp

Dit was de fase waarin ik het ontwerp van het systeem had opgesteld. Het gaf structuur en visie aan de development-fase.

Tijdens de opleiding hadden we twee verschillende methodes van ontwerpen behandeld. Klassendiagrammen met UML en het opstellen van een architectuur in blok 12 Architectuur en Security. Voor het opstellen van het ontwerp had ik gebruik gemaakt van de theorie van blok 12. Een UML klassendiagram is meer geschikt voor een systeem dat uit klassen bestaat. Ik was niet van plan een systeem te bouwen dat bestond uit verschillende klassen, maar een systeem dat geplaatst zal worden in een bestaande architectuur en zal communiceren met andere systemen en componenten. Hier was de methode van blok 12 beter geschikt voor. Hieronder is een versimpelde versie van de architectuur te zien. Een uitgebreidere versie is te vinden in de "Architecture Advanced" bijlage.



Figuur 5: Versimpelde architectuur

Het proces begint bij de Jupyter Notebook. Hier wordt de ruwe data van ReviewByMe en Mailchimp opgehaald en vervolgens getransformeerd tot

een dataset waarmee het model gebouwd en getraind wordt. Na dit proces wordt het model gedeployed op een AWS SageMaker endpoint.

Daarna wordt er een e-mail naar een klant gestuurd. Deze e-mail bevat de endpoint URL van de AWS Lambda Reward Generator. Aan deze URL wordt de ID van de klant meegegeven als parameter. Wanneer de klant de e-mail opent, haalt de Reward Generator Lambda functie de ruwe data van deze klant op met calls naar ReviewByMe en Mailchimp. Deze ruwe data worden getransformeerd tot een dataset, zodat ze leesbaar zijn voor het gedeployed ML model. Vervolgens stuurt de Reward Generator Lambda functie deze data naar het gedeployed SageMaker model. Deze model geeft de voorspelling als antwoord. De Reward Generator bepaald dan de beloning met behulp van deze voorspelling en genereert hier een afbeelding voor. Deze afbeelding wordt dan weergeven in de e-mail.

8 Development

In deze fase had ik een systeem gebouwd dat in staat was om de theorie en hypotheses die ik in het resultaat van mijn deskresearch had geformuleerd, te toetsen.

8.1 Aanpak

In deze paragraaf behandel ik de gekozen aanpak en planning voor de development-fase van het project.

8.1.1 Software development methode

Bij het kiezen van de software development methode voor de development-fase, had ik twee verschillende vormen van methodes overwogen: de Watervalmethode en een agile methode. Ik had deze twee overwogen, omdat ik er al ervaring mee had en gezien de korte tijd voor het project, hoefde ik niet tijd te besteden aan het leren van een nieuwe methode. Om te bepalen welke methodologie het meest voordelig zal zijn, had ik de voor- en nadelen van elk bekeken en vergeleken.

Agile

Agile is een iteratieve, team gebaseerde manier van ontwikkelen. De snelle levering van complete functionele componenten is een van de kernpunten van deze aanpak. In plaats van het maken van een langetermijnplanning, wordt er gewerkt met sprints (Lotza, 2018). De Agile methode heeft een aantal voordelen die mogelijk gunstig zijn voor mijn project. Agile maakt veranderingen in de ontwikkeling mogelijk, terwijl Waterval minder ruimte geeft om de requirements te wijzigen zodra de ontwikkeling is begonnen (Lotza, 2018). Met Agile kan sneller een werkende minimale versie, met beperkte functionaliteit, van de software worden gerealiseerd (Lotza, 2018).

Echter, heeft de Agile methode ook nadelen. Namelijk dat het hoge team coördinatie en aanwezigheid van de klant vereist (Kukhnavets, 2019). Als deze coördinatie en aanwezigheid laag is, kan het leiden tot problemen. Daarnaast is het moeilijk om in te schatten wat realiseerbaar is in een sprint als er bijvoorbeeld gewerkt wordt met een onbekende technologie.

Waterval

De Watervalmethode is een wat ouder lineaire manier van softwareontwikkeling dat bestaat uit sequentiële fases. Elke fase is een afzonderlijk stadium van softwareontwikkeling, en begint pas als het vorige stadium is afgerond (Lotza, 2018). De Watervalmethode heeft een aantal voordelen die mogelijk gunstig zijn voor mijn project. Er wordt vroeg afgesproken wat er gedaan moet worden (Lotza, 2018). De voortgang is makkelijker te meten (Lotza, 2018). Er kan aan verschillende componenten tegelijk worden gewerkt (Lotza, 2018). Er is minder aanwezigheid van de klant vereist (Lotza, 2018).

Echter, heeft de Watervalmethode ook nadelen. Opdrachtgevers weten meestal, in het begin nog niet wat ze precies willen (Lotza, 2018). Hierdoor is het verzamelen van de requirements wat moeizamer.

Conclusie

De beste aanpak voor het project was een combinatie van Agile en Waterval waar het voornamelijk Waterval was met een aantal kenmerken van Agile eraan toegevoegd. Deze aanpak had ik gekozen omdat er voor het project weinig aanwezigheid van de klant vereist was; bij het kiezen van een aanpak voor de development-fase, was al bekend wat er gedaan moest worden: ik had een theorie die getoetst moest worden met het ontwikkelen van een stuk software dat bestaat uit verschillende componenten; de exacte requirements waren nog niet bekend, maar de belangrijkste wel. Ten slotte was het belangrijk om de voortgang goed te meten, gezien de beperkte tijd. Voor al deze redenen was de Watervalmethode beter geschikt.

Echter, wilde ik niet pas aan het einde van de timeline een werkende stuk software hebben en mogelijk een "big-bang" meemaken. Als het dan niet werkte, zal ik het mogelijk niet meer af krijgen wegens de beperkte tijd. Kenmerken van de Agile methode konden hier gebruikt worden om dit te voorkomen. Dit hield in dat ik zo snel mogelijk een werkende stuk software zal ontwikkelen en deze iteratief verbeteren.

8.1.2 Omgeving

Ik had van mijn stakeholders de requirement gekregen om gebruik te maken van AWS. AWS heeft een breed assortiment aan Machine Learning-diensten. Van vooraf opgeleide services tot tools waarmee nieuwe Machine Learning-modellen kunnen worden gebouwd. De meest interessante voor mij was AWS SageMaker. AWS SageMaker is een IDE die volgens Amazon alle tools die nodig zijn voor Machine Learning development samenbrengt. (Simon, 2019).

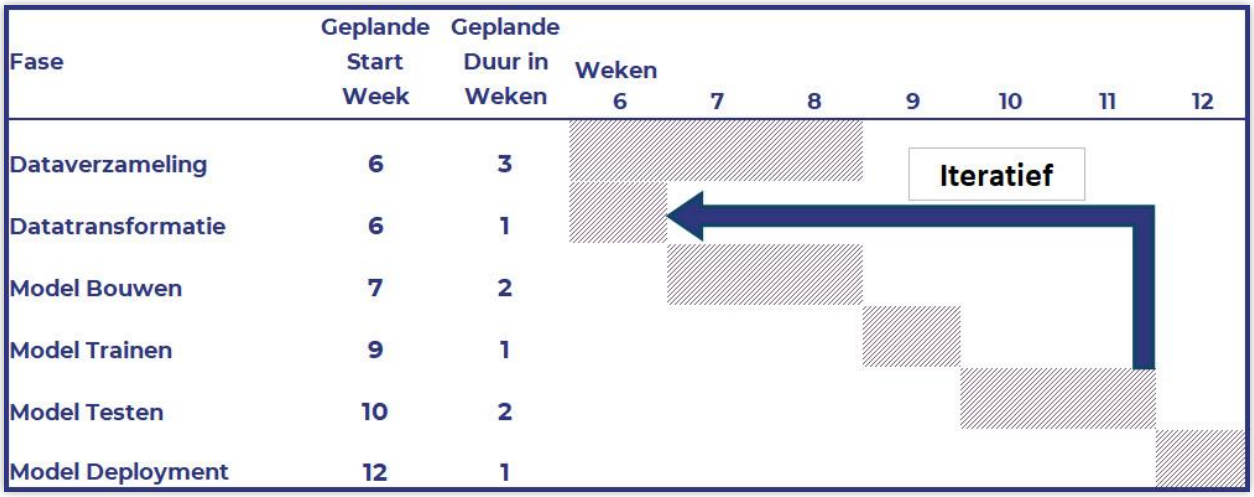
8.1.3 Fasering

Voor het opstellen van de fasering en het plan van aanpak voor de development-fase had ik een aantal bronnen onderzocht. Volgens deze bronnen bestaat het Machine Learning Development Lifecycle uit een aantal vaste stappen. Echter, worden deze per bron net wat anders vermeld. Bijvoorbeeld volgens Amazon (AWS Online Tech Talks, 2020) zijn er 9 stappen die onder te verdelen zijn in de volgende vier categorieën: Prepare Data, Build, Train & Tune, Deploy & Manage. Volgens Google (Ferlitsch, 2019) zijn er 11 stappen die onder te verdelen zijn in de volgende drie categorieën: Planning, Data Engineering, Modeling. Echter, is er bij Google geen consistentie en standaard voor deze fasering. Bij een Machine Learning cursus van Google (Google, 2019) worden er 5 categorieën behandeld in plaats van 3. Omdat ik tijdens de deskresearch, de cursus van Google had gevolgd, had ik gekozen om de 5 categorie fasering te hanteren. De fasering is

als volgt: probleem en oplossing, dataverzameling, datatransformatie, model training en ten slotte model deployment en gebruik.

Figuur 6: Het stappenplan van Google (Google, 2019) vertaald naar het Nederlands

Om structuur in het hoofdstuk te behouden, krijgt elke fase een aparte paragraaf, tenzij deze deel is van het iteratief proces. In dat geval krijgt elke iteratie een aparte paragraaf.



Figuur 7: Gantt chart van de development-fase

Het deployen van het model had ik voor het einde gelaten, aangezien dit niet van belang was zonder een kwalitatief model.

8.1.4 Probleem en oplossing

Het probleem en de oplossing waren grotendeels al geformuleerd in de voorgaande fases (hoofdstuk 3

Making Offers Intelligent). Echter, had de aanpak van Google een aantal vragen over het ML model die ik nog niet had beantwoord.

Wat moet het ML model doen?
Het ML model moet de optimale beloning voor een klant vinden.

Wat is het ideale resultaat?
Het ideale resultaat is om klanten beloningen te geven die net genoeg zijn om te leiden tot conversie en niet meer. Dit houdt in dat de kosten per conversie zo laag mogelijk worden gehouden en het aantal conversies zo hoog mogelijk.

Wanneer is het ML model geslaagd?

Een succes statistiek is de behaalde winst. Echter, is deze moeilijk of zelfs onmogelijk te meten. Statistieken die wel kunnen worden gemeten zijn het aantal conversies en de gemiddelde kosten per conversie.

Het model is een succes wanneer de gemiddelde kosten per conversie relatief minder stijgen dan het aantal conversies. Dus als de kosten per conversie zijn gestegen met 5% en het aantal conversies met 10%, dan stel ik het model als succesvol.

Dit kan ook de andere kant op. Het model is een succes wanneer het aantal conversies relatief minder daalt dan de gemiddelde kosten per conversie. Dus als het aantal conversies 5% daalt en de kosten per conversie met 10%, dan stel ik het model als succesvol.

Afhankelijk van het totaal aantal klanten, kan toeval invloed hebben op het resultaat. Ik ga uit van een verschil in percentage tussen de controle groep en experimentele groep van minimaal 5% om het model statistisch significant succesvol te verklaren.

Welke output moet het ML model produceren?
Het ML model kan in 2 fases worden onderverdeeld. De eerste is om een score te voorspellen en de tweede is om een beloning te voorspellen.

Het ML model moet de kans dat een klant een conversie aangaat voorspellen. Voor de klant zijn er twee mogelijke uitkomsten: de klant gaat wel een conversie aan of de klant gaat geen conversie aan. Dit is een classificatie probleem met twee opties (binaire classificatie). Een ML model voorspelt de waarschijnlijkheid voor beide opties. De waarschijnlijkheid van alle opties bij elkaar is altijd 1 (bijvoorbeeld [0,3][0,7]). Aangezien, er maar twee opties zijn bevat de score van één al alle informatie. Als de ene score 0,3 is, dan is de andere 0,7. Dus de output van het model is alleen de kans dat een klant wel een conversie aangaat. Hiervoor ga ik gebruik maken van Supervised Learning.

Het ML model moet een beloning voorspellen die een klant zal verleiden tot conversie. De lijst van beloningen waaruit gekozen kan worden bestaat uit een aantal beloningen. Dit is een classificatie probleem met meer dan twee opties (multi-class classificatie). Een ML model voorspelt de waarschijnlijkheid dat een klant zal converteren voor alle beloningen. De waarschijnlijkheid van alle opties bij elkaar is altijd 1 (bijvoorbeeld [0,2][0,5][0,15][0,10][0,05]). De output is de beloning met de hoogste score. Hiervoor ga ik gebruik maken van Supervised Learning.

Hoe gaan de ML-voorspellingen worden gebruikt?
De voorspellingen gaan worden gebruikt om een beloning voor klanten te genereren en deze aan te bieden via een e-mail.

Welke heuristische oplossing moet het ML model overtreffen?

Een heuristische oplossing is om de top 25% oudste klanten geen beloning te geven aangezien, deze waarschijnlijk loyaler zijn en bereid zijn meer uit te geven (Bernazzani, 2020).

De top 25% nieuwste klanten krijgen de grootste beloning, aangezien deze waarschijnlijk minder loyaal zijn en niet bereid zijn veel uit te geven (Bernazzani, 2020).

De klanten die niet in een van deze 2 categorieën vallen, krijgen een kleinere beloning die hetzelfde is voor iedereen in deze groep.

8.1.5 Dataverzameling

De volgende stap is het verzamelen van gelabelde data. Dit ga ik doen met behulp van ReviewByMe. Het is een tool van Unchained Carrot waarmee bedrijven onderaan e-mails enquêtes kunnen toevoegen. Als ontvangers van deze e-mails op deze enquêtes klikken, worden er gegevens van deze persoon verzameld, zoals het gekozen antwoord, IP-adres en apparaat informatie.

De ReviewByMe-service van Unchained Carrot is echter splinternieuw, dus er zijn nog niet veel klanten en data. Het idee is om op zoek te gaan naar bedrijven, die mee willen doen aan mijn opdracht. Dit kan echter een tijdje duren. Een tip van Brownlee is om een kleine batch data te verzamelen en aan de slag te gaan (Brownlee, 2017). Unchained Carrot had al een klant die gebruik maakte van RVBM, dus ik kon deze aanbeveling opvolgen.

Naast de gegevens die RVBM verzamelt, zal ik kijken welke gegevens kunnen worden verkregen uit Mailchimp. Dit is de e-mail marketing provider die wordt gebruikt door de klant waar ik het zojuist over had.

Mailchimp bevat onder andere de demografische gegevens van contacten. Demografische gegevens van klanten, zoals leeftijd en geslacht, kunnen een kern rol spelen bij het personaliseren van beloningen (Al-Zuabi, Jafar en Aljoumaa, 2019).

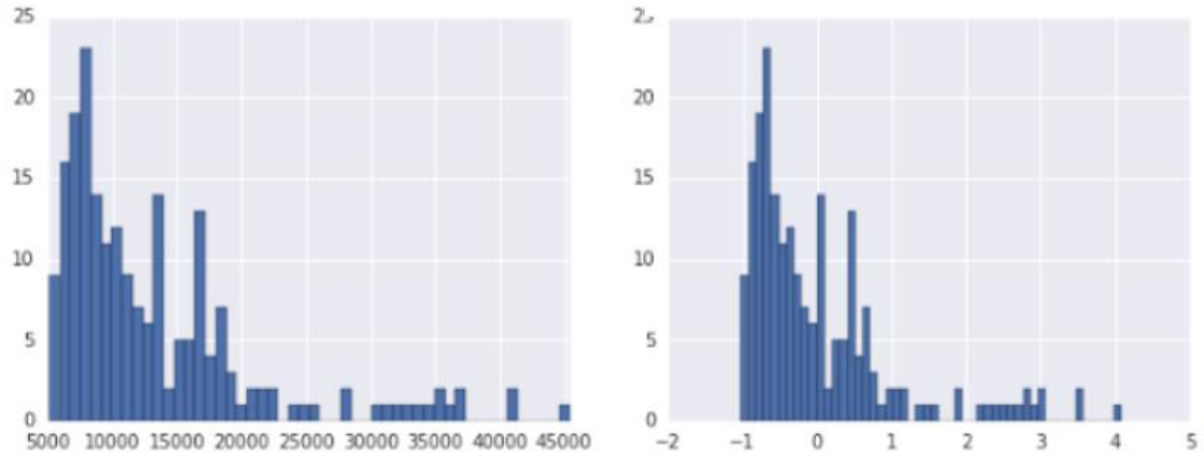
Voor het verzamelen van gelabelde data gaan de bedrijven drie e-mails versturen naar haar klanten. De eerste e-mail is een RVBM enquête zonder beloning. De tweede is een A/B-test, waarbij de helft van de klanten allemaal een enquête krijgt met dezelfde beloning. De andere helft ontvangt een e-mail zonder een beloning. De derde e-mail is een A/B/C-test. Een derde van de klanten ontvangt een enquête zonder beloning, een derde een enquête waarbij ze allemaal dezelfde beloning ontvangen. De klanten in het laatste deel ontvangen allemaal een aparte beloning die is gegenereerd door het ML model.

8.1.6 Datatransformatie

In deze fase ga ik de data analyseren en klaar maken voor Machine Learning algoritmes. Voordat data leesbaar is voor een Machine Learning algoritme, moet het voldoen aan een aantal eisen. De dataset

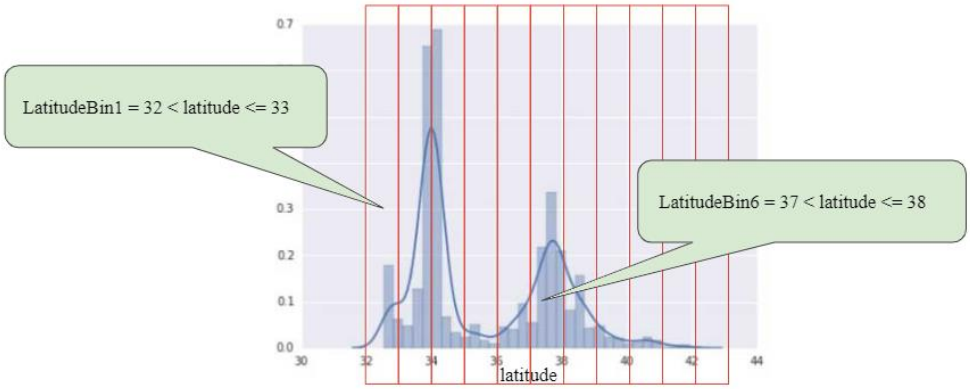
moet volledig numeriek zijn. Dit houdt in dat non-numerieke waardes, zoals kleuren, moeten worden geconverteerd naar numeriek. Een techniek die ik hiervoor ga gebruiken is one-hot-encoding. Elke kleur krijgt hier een aparte kolom. Bijvoorbeeld de waardes rood, groen, blauw voor de eigenschap kleur, krijgen alle een aparte kolom, zoals kleur_rood, kleur_groen, kleur_blauw en dan komt er een 0 of 1 te staan als waarde in deze kolommen als de eigenschap waar is of niet.

In een dataset met meerdere numerieke features, moeten de waardes dezelfde schaal hebben. Als er een feature is met waardes tussen de 10-20 en er is een feature met waardes tussen de 100-200, kunnen er problemen ontstaan. Bijvoorbeeld dat de feature met de grotere waardes meer invloed gaat hebben op de voorspellingen, omdat het grotere waardes bevat en niet omdat het daadwerkelijk een belangrijkere feature is. Hier kan normalisatie worden gebruikt om de 100-200 te schalen naar 10-20 zonder dat er waardes verloren gaan. Deze techniek ga ik gebruiken. Een voorbeeld van normalisatie is in de onderstaande afbeelding te zien. De waardes zijn van 5000-45000 genormaliseerd naar -2-5.



Figuur 8: Normalisatie gevisualiseerd (Google, 2020)

Een probleem met sommige numerieke waardes, zoals de leeftijd is dat een leeftijd van 56 niet veel meer informatie over een persoon bekend maakt dan een leeftijd van 55; en een leeftijd van 24 maakt niet veel meer informatie bekend dan een leeftijd van 25. Daarnaast kunnen er teveel unieke waardes zijn waardoor het model minder goed kan trainen. Deze losse waardes kunnen beter in groepen worden geplaatst. Binning is een techniek die ik hiervoor ga gebruiken. Nummers worden geplaatst in zogenaamde bins. Bijvoorbeeld de getallen 5,8,12,14,22,28 kunnen worden geplaatst in bins van 10: 0-10,10-20,20-30. Een voorbeeld van binning is in de onderstaande afbeelding te zien.



Figuur 9: Binning gevisualiseerd (Google, 2019)

8.1.7 Model training

Voor het trainen van het model ga ik gebruik maken van de Scikit-learn library. Scikit-learn is een Machine Learning library die vriendelijk is voor beginners. Het ondersteunt de meest populaire algoritmes, zoals Linear Regression, Logistic Regression en Random Forest. Scikit-learn wordt voornamelijk gebruikt voor kleinere projecten en is minder geschikt voor grote projecten waar er onder andere Neurale Netwerken worden gebruikt op grote hoeveelheden data (D., 2020).

Ik ga de techniek genaamd Spot Checking gebruiken om tot een uiteindelijke algoritme te komen (Brownlee, 2020). In dit proces worden snel verschillende algoritmes getest om erachter te komen welke effectief zijn, zodat daarop gefocust kan worden.

8.1.8 Model deployment en gebruik

Ik ga het model deployen op een AWS SageMaker endpoint. Dit is het meest vanzelfsprekend, aangezien ik het model ga ontwikkelen in de IDE AWS SageMaker. Door het model op een AWS SageMaker endpoint te deployen, is het gelijk beschikbaar voor alle andere componenten in de AWS architectuur.

Het model gaat gebruikt worden om persoonlijke aanbiedingen in e-mails te plaatsen. Dit gaat gedaan worden met een zogenaamde Reward Generator. Deze moet real-time zijn, dit houdt in dat het pas een beloning genereert, op het moment dat een e-mail geopend wordt. Het eerste waar ik aan dacht hoe dit te implementeren was met Javascript in een e-mail. Echter, is het niet mogelijk om Javascript in e-mails te stoppen wegens veiligheidsredenen (Chapman, 2019). Ik had onderzocht of er andere mogelijkheden zijn om real-time data in e-mails te plaatsen.

Mijn onderzoek had geleid tot de conclusie dat er maar 1 manier is om real-time data in e-mails te plaatsen. Dit is met behulp van plaatjes.

In e-mails, is de HTML image tag, de enige stuk content dat is toegestaan om een call te maken buiten de e-mail client.

Voor mijn opdracht is het idee om de source van de image te verwijzen naar een endpoint die real-time een reward genereert en een image van de reward als antwoord terug geeft. Er zijn een aantal manieren om dit te doen, zoals het converteren van HTML naar een plaatje, het plaatje helemaal genereren met code en derde partij oplossingen. Ik had met mijn stakeholders erover gesproken en de requirement die ik had gekregen was om gebruik te maken van (Scalable Vector Graphics) SVG plaatjes.

Een SVG plaatje is net zoals een HTML bestand, opgebouwd uit regels code. Als een SVG bestand geopend wordt in een code editor, lijkt het bijna op een HTML bestand. Dit betekent dat als er tekst in de afbeelding wordt weergeven, deze aan te passen is door in de code deze tekst aan te passen. Een designer kan een SVG bestand met een merge tag, zoals {AMOUNT} maken. Deze merge tag kan dan met code worden aangepast naar de gegenereerde waarde.

Figuur 10: Een voorbeeld van een SVG plaatje met een merge tag

Een van de manieren om deze input mee te geven is via de URL. Ik zal dan in de URL als parameter de klant mee moeten meegeven. Ik had een klein onderzoek gedaan, en kon concluderen dat de meeste e-mail marketing providers de mogelijkheid geven om gegevens van een klant, zoals de user ID in de e-mail en dus ook de URL van de Reward Generator afbeelding te plaatsen.

Deze manier had de minste LEAN-wastes van alle manieren die ik had kunnen vinden om met een e-mail marketing provider gegevens van een klant te sturen naar de Reward Generator. Een andere oplossing was om een gehele integratie tussen de Reward Generator en de e-mail marketing providers te bouwen. Het probleem hiermee was dat het voor elke e-mail marketing provider apart gebouwd zal moeten worden. Ook zal het mogelijk alleen een betere user experience voor gebruikers opleveren wat niet belangrijk was voor dit project, aangezien er maar 1 gebruiker was, ik. Vandaar dat ik had gekozen voor de eerste optie.

De Reward Generator ga ik deployen op AWS Lambda. Dit is een service van AWS die het mogelijk maakt om online schaalbare code te runnen zonder servers. Door gebruik te maken van AWS Lambda, is de functie

dus schaalbaar. Een van de andere requirements die ik van mijn stakeholders had gekregen is om de Reward Generator in Node.js te schrijven, zodat ik het in de huidige architectuur van Unchained Carrot kon deployen.

8.2 Werkzaamheden

In deze paragraaf behandel ik de daadwerkelijke werkzaamheden per fase.

8.2.1 Dataverzameling

Voor het trainen van het ML model was data nodig. Hiervoor had ik contact opgezocht met bedrijven om te vragen of ze wilden deelnemen aan mijn opdracht. Het proces van bedrijven contacteren voor dataverzameling was tijdens heel de development-fase ingang gebleven.

Mark, een van mijn begeleiders en de directeur van Unchained Carrot heeft een uitgebreid LinkedIn netwerk. Ik had voorgesteld om deze te gebruiken om de zoektocht naar bedrijven te starten. Ik had een bericht opgesteld en deze naar Mark gestuurd. Hij had deze onder zijn contacten verdeeld.

Het bericht was verspreid naar honderden contacten. Slechts 8 bedrijven hadden het formulier ingevuld. Deze 8 had ik vervolgens een uitnodiging gestuurd voor een Zoom call voor verdere instructies. Alleen 1 had deze geaccepteerd. Helaas kreeg ik na de Zoom call ook geen antwoord meer van deze.

Jammer genoeg had de LinkedIn netwerk van Mark niet geholpen met het vinden van bedrijven die bereid waren om mee te doen aan mijn opdracht. Voor mijn tweede poging voor het vinden van bedrijven had ik bedrijven direct benaderd via e-mail. Deze aanpak was succesvol voor mij tijdens de minor ondernemen, vandaar dat ik hiervoor had gekozen.

Uiteindelijk had ik met deze methode 406 bedrijven een e-mail gestuurd. Online had ik een database gevonden met website gegevens van bedrijven. Met een Python e-mail scraper die ik tijdens de minor had ontworpen, had ik de e-mails van deze bedrijven verzameld. Daarna had ik met SalesHandy, die ik ook tijdens de minor had gebruikt, elk van deze bedrijf een persoonlijk e-mail verstuurd. Met persoonlijk bedoel ik dat de bedrijfsnaam ergens in de e-mail wordt vermeld, zodat het lijkt dat de e-mail speciaal voor deze bedrijf is geschreven. Ik had de e-mail gepersonaliseerd, omdat ik tijdens mijn deskresearch (5.3) had ontdekt dat e-mails personaliseren, de kans op antwoord verhoogt (Smart Insights, 2019; Prakash, 2020; Mialki, 2020).

Ik was begonnen met het versturen van berichten naar Nederlandse bedrijven. Ik ging ervan uit dat het respons hiervan hoger zal zijn, aangezien mensen voorkeur hebben voor het bekende (Vries, Erasmus, & Gerber, 2017). Als Nederlands student een Nederlands bericht sturen naar Nederlandse bedrijven, bevat minder onbekende factoren voor het

ontvangende bedrijf dan als Nederlands student een Engels bericht sturen naar buitenlandse bedrijven.

De respons was inderdaad hoog. Alleen kreeg ik geen antwoord meer terug, nadat ik aangaf dat ik toegang zal krijgen tot gevoelige data. Hiervoor had ik een data sharing overeenkomst samengesteld met de hulp van mijn afstudeerbedrijf. Ik had deze aan de Nederlandse bedrijven voorgesteld en was ondertussen verder gegaan met het contacteren van bedrijven.

Na het contacteren van Nederlandse bedrijven, had ik bedrijven in de Verenigde Koninkrijk benadert en tenslotte bedrijven in de Verenigde Staten. Dit had ik gedaan, aangezien veel bedrijven in de database in deze landen waren gevestigd en we spraken een gemeenschappelijke taal waardoor er minder kans was op communicatieproblemen.

Van de bedrijven in de VS had ik nauwelijks antwoord gekregen. Met de bedrijven in de VK was dit anders. Ik kreeg heel wat antwoorden, maar minder dan de Nederlandse bedrijven. Uiteindelijk liep ik met de bedrijven in de VK tegen hetzelfde probleem aan. Wegens de betrekking van gevoelige data, wilden de bedrijven niet verder. De overeenkomst wilden ze ook niet tekenen, aangezien het was opgesteld volgens de Nederlandse wet. Een versie opstellen volgens de Engelse wet was geen optie meer, aangezien hier geen tijd meer voor was.

Uiteindelijk had ik toch nog antwoorden gekregen van twee Nederlandse bedrijven die ook nog eens bereid waren om de overeenkomst te tekenen. Echter, is dit pas gebeurd in een van de laatste weken van het project en hier kom ik in hoofdstuk 10 Afronding op terug.

8.2.2 ML model iteratie #1

Voor de eerste iteratie had ik nog geen bedrijven gevonden die bereid waren om deel te nemen aan mijn opdracht. Hierdoor was ik alvast gaan experimenteren met de data die Unchained Carrot al beschikbaar had. Het idee was om een model te bouwen dat voorspelde of iemand wel of niet op een ReviewByMe survey antwoord zal geven.

Het doel van het eerste model was om zo snel mogelijk een werkend model te krijgen. Hierdoor had ik minder tijd besteed aan het analyseren van de data.

Tijdens het analyseren en transformeren van de data had ik heel wat nieuwe informatie ontdekt. De belangrijkste bevinding was dat alle data die ReviewByMe verzamelde, ook was op te halen via de Mailchimp API. De Mailchimp API gaf zelfs toegang tot meer data. Dit betekende dat het niet nodig was om de gegevens via ReviewByMe op te halen.

Ik had de ruwe data opgehaald via de Mailchimp API, hier een dataset van gemaakt, deze opgeschoond en features ontworpen.

Daarna ben ik het model met de dataset gaan trainen. Hier liep ik tegen een probleem aan. De dataset was zo erg ongebalanceerd, dat het niet te gebruiken was. Het bevatte 937 klanten waarvan slechts 11 antwoord hadden gegeven op de ReviewByMe survey.

Deze verhouding veroorzaakte een aantal problemen. Het opsplitsen van de dataset was vrijwel onmogelijk. Voor het testen van een ML model moet een dataset worden opgesplitst in een set waarmee het model getraind wordt en een set waarmee het model getest wordt. Wegens de 937:11 verhouding, was het vrijwel onmogelijk om de dataset zo op te splitsen dat de opgesplitste groepen nog representatief van elkaar waren. Er was kans dat alle 11 die wel een antwoord hadden gegeven, allemaal in de test set terecht kwamen of andersom. Het model zal dan weinig of geen voorspellingskracht hebben gehad.

Het tweede probleem met deze ongebalanceerde dataset was dat de nauwkeurigheid van de voorspelling veel te hoog zal zijn. Als het model alles voorspelt had als "Geen antwoord", dan had het al 937 goede voorspellingen en was dit een score van 99%. Het klinkt goed maar hier had ik niks aan.

Om dit op te lossen, had ik een van de bronnen die ik had onderzocht tijdens mijn deskresearch, Machine Learning Mastery, erbij gepakt (Brownlee, 2020). Er werden hier 8 oplossingen genoemd om het probleem op te lossen. Echter, was er maar 1 die ik meteen kon gebruiken: De dataset resamplen. In mijn geval betekende dit dat ik op een planmatige manier een aantal mensen die geen antwoord hadden gegeven op de survey uit de dataset moest verwijderen. Mijn dataset bevatte alle klanten, ook de klanten die de e-mail met de survey niet hadden geopend. Deze konden weg, aangezien het onmogelijk was om antwoord te geven op de survey zonder de e-mail te openen.

Na deze oplossing toe te passen had ik een meer gebalanceerd dataset met een verhouding van 521:11. Dit was nog steeds niet genoeg.

Hierdoor had ik besloten het doel van het model aan te passen, zodat ik wel kon werken met een gebalanceerd dataset. Deze doel was het voorspellen of iemand wel of niet een e-mail gaat openen. Met dezelfde dataset hadden 433 van de 937 mensen de e-mail geopend. Dit was een verhouding waarmee ik wel mijn model kon trainen. Het uiteindelijke doel was wel om te voorspellen of iemand wel of niet antwoord gaat geven op een survey, maar totdat ik een bedrijf had gevonden die mij wilde helpen met het verzamelen van data, was voorlopig de beste optie om te voorspellen of iemand wel of niet een e-mail gaat openen, zodat ik het proces een keer kon doorlopen.

8.2.3 ML model Iteratie #2

Het doel van het tweede model was hetzelfde als het vorige model: zo snel mogelijk een werkend model krijgen. Bij deze model waren er geen problemen bij het splitsen van de dataset. Bij het trainen van het

model, had ik besloten om alleen 2 features als input te nemen: de leeftijd en geslacht. Dit had ik gedaan om mijn kans dat ik een werkend model kreeg, te verhogen. De gekozen algoritme was Logistic Regression, omdat deze paste bij het Machine Learning probleem (binaire classificatie, de uitkomst is wel of geen antwoord op de survey) en een van de simpelere algoritmes was om te implementeren en te begrijpen. Zo kon ik sneller een werkend model krijgen, wat het doel was van deze iteratie.

De uiteindelijke score van deze model was 54%. Dit was maar iets hoger dan het openingspercentage van de e-mail van $\frac{433}{936} * 100 = 46\%$. Het model was dus nog niet heel sterk.

De volgende stap was om deze score hoger te krijgen. Dit had ik gedaan door bestaande features, zoals hoe actief een klant is, de locatie van de klant en het gebruikte toestel voor het openen van e-mails aan het model toe te voegen. De features had ik gekozen met behulp van een data analyse die achterhaalde welke features de meeste voorspelkracht hadden.

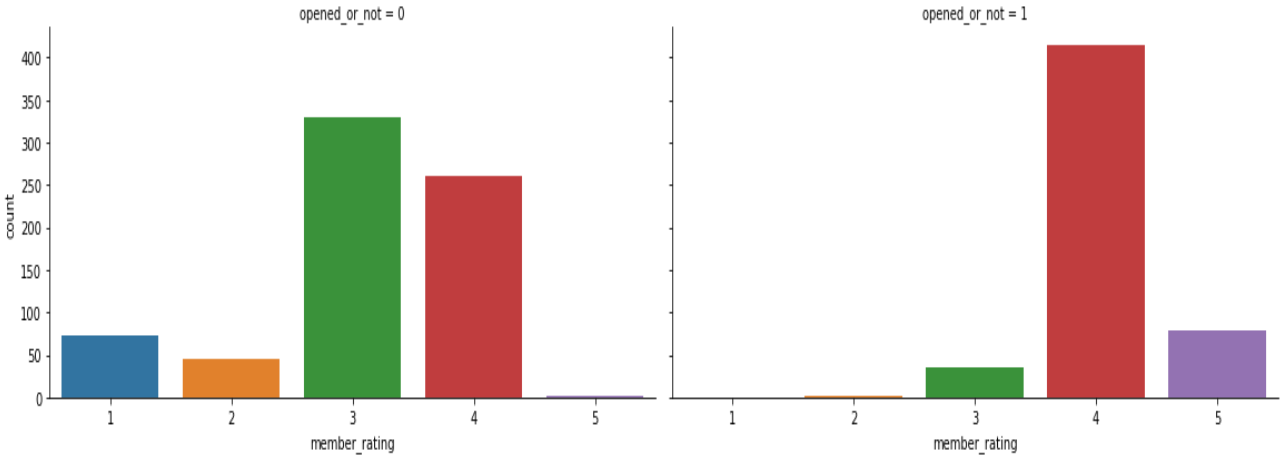
Uiteindelijk had ik een score van 87% behaald met de Logistic Regression algoritme. Ik had het model ook getest met de Random Forest algoritme en hiermee had ik een score van 88% behaald. Deze algoritme had ik als tweede gekozen, omdat het een ander soort algoritme was (boomstructuur), vaak werd gebruikt voor classificatie problemen en snel te implementeren was.

8.2.4 ML model Iteratie #3

Het doel van het derde model was om een zo sterk mogelijk model te krijgen. Dit zal ik doen door meer tijd te besteden aan het analyseren van de data, de juiste features kiezen en ontwerpen, en het beste algoritme vinden door verschillende algoritmes te checken. Het opschonen van de data had ik al gedaan in de voorgaande iteraties. Deze iteratie was begonnen met het analyseren van de data.

Een goede analyse kan veel bekendmaken over de data. Dit kan helpen met het bouwen van een sterk model. Er kan bijvoorbeeld achterhaald worden welke features veel of juist weinig voorspel kracht hebben.

Een van de features die ik geanalyseerd had was de member_rating. Deze was opgesteld door Mailchimp. In de onderstaande grafiek is te zien dat bijna niemand met een member_rating van 5 de e-mails niet opent. De member_rating is dus een feature met veel voorspelkracht.



Figuur 11: Grafiek van de member_rating feature opend_or_not=0 betekent niet geopend en opend_or_not=1 betekent wel geopend.

Voor het derde model had ik ook mijn eigen features ontworpen. Ik had bijvoorbeeld de e-mail domeinnaam extraheert uit de e-mail. De domeinnaam kan veel zeggen over een persoon (Schleckser, 2018). Als het bijvoorbeeld niet bekend was, zoals gmail.com of outlook.com, maar bijvoorbeeld unchainedcarrot.com, dan was de klant misschien een ondernemer. Als het bijvoorbeeld een wat oudere domeinnaam, zoals aol.com was, dan was deze persoon misschien wat ouder. Deze gegevens konden een rol spelen bij het personaliseren van de beloningen.

Nadat de data opgeschoond was, de features waren uitgekozen en de data leesbaar was voor ML-algoritmes, kon ik beginnen met het trainen van het model. Er was een ruime keuze aan algoritmes. Ik had een techniek genaamd Spot Checking (Brownlee, 2020) gebruikt om snel verschillende algoritmes te testen om erachter te komen welke algoritmes het meest effectief waren, zodat ik daarop verder kon bouwen.

De eerst stap was om 5-10 populaire algoritmes te kiezen die geschikt waren voor het Machine Learning probleem. Ik had voor de volgende algoritmes gekozen: Random Forest, Decision Tree, K-Nearest Neighbors, Logistic Regression, Linear Support Vector Classification, Support Vector Machines, Stochastic Gradient Descent, Perceptron en Naive Bayes.

In de onderstaande tabel zijn de scores die ik behaald had met de verschillende algoritmes te zien. De scores zijn nog wel gebaseerd op de data waarmee het model was getraind en niet op nieuwe onbekende data. Hierdoor zijn de scores enorm hoog. In het volgende model was ik van plan om te testen met nieuwe data. Het testen op nieuwe data zorgt voor een meer betrouwbare score.

Model	**Score**
Random Forest	100
Decision Tree	100

KNN	95
Logistic Regression	90
Linear SVC	89
Support Vector Machines	88
Stochastic Gradient Descent	78
Perceptron	75
Naive Bayes	61

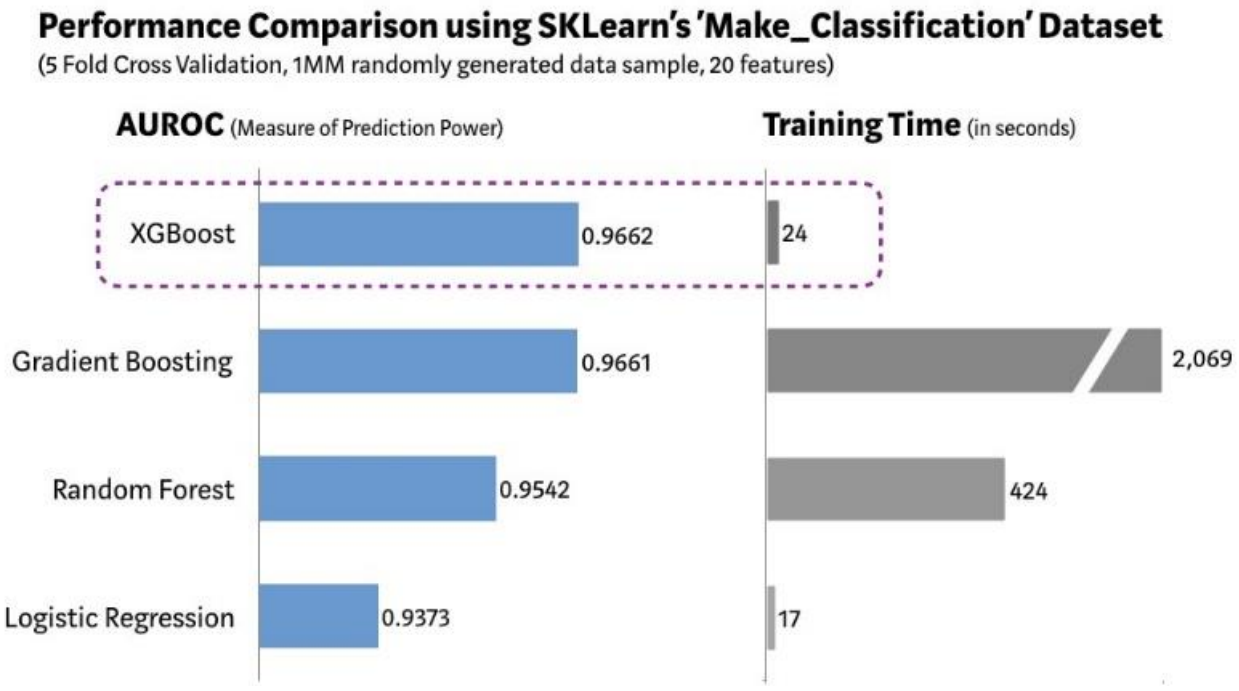
Figuur 12: Scores van de verschillende algoritmes

De algoritmes met een boomstructuur hadden de hoogste scores. Met deze algoritmes was ik van plan verder te gaan.

8.2.5 ML model Iteratie #4

Na het afronden van het vorige model, was ik boom algoritmes verder gaan onderzoeken. Tijdens dit onderzoek was ik de XGBoost (eXtreme Gradient Boosting) algoritme tegen gekomen. Een gradient boosting algoritme combineert zwakke voorspellingen om een sterker model te creëren (Totta data lab, 2019).

XGBoost is een gradient boosting algoritme dat gebruik maakt van een aantal nieuwe technieken waardoor het meer betrouwbare voorspellingen maakt en minder resources in beslag neemt (Morde, 2019). In de onderstaande grafiek zijn de resultaten van XGBoost te zien in vergelijking met een aantal andere algoritmes.



Figuur 13: Prestaties van XGBoost in vergelijking met een aantal andere algoritmes (Morde, 2019)

In de grafiek is XGBoost een duidelijke winnaar. Deze algoritme was ik van plan om voor deze iteratie te gebruiken.

Het doel van de 4e iteratie was om het model klaar te maken voor deployment. Ik had al de stappen gevolgd, zoals in de voorgaande modellen, alleen dit keer had ik gebruik gemaakt van de XGBoost algoritme en had ik getest met nieuwe data.

Ik had met de XGBoost algoritme net als de andere boomstructuur algoritmes, op de bekende data een score van 100% behaald, maar op de nieuwe data was dit een score van 98% wat nog steeds enorm hoog was. Aan de score was dus met XGBoost niet veel veranderd, maar de performance was velen malen beter. Met de Random Forest duurde het trainen van het model 2 minuten. Met XGBoost heeft het maar 5 seconden geduurd.

8.2.6 Deployment

Voor het deployen van het model moest ik eerst een automatische pipeline opzetten. Deze pipeline voert de verschillende stappen die ik eerst handmatig had uitgevoerd, zoals data opschonen, data transformeren en features ontwerpen, automatisch uit. Ik had het model gedeployed op een AWS SageMaker endpoint, zoals gepland. Het deployen van het model was probleemloos verlopen. Na het deployen kon ik met omliggende software een call (met de ruwe data in de body) maken naar de endpoint, en kreeg ik de gemaakte voorspelling terug. Echter, moest ik nog een component ontwikkelen om de voorspellingen voor het originele doeleinde te kunnen gebruiken. Deze component was de Reward Generator.

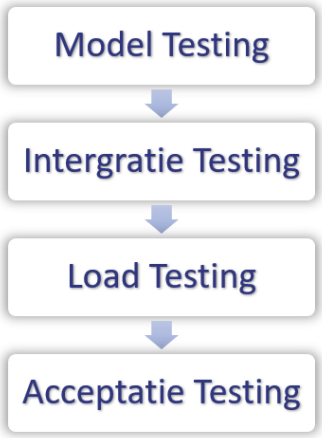
De eerste versie van de Reward Generator was zo ingesteld dat de merge tag in het SVG plaatje aangepast werd aan de parameterwaarde in de URL. Dus als de source van de afbeelding rewardgenerator.com?amount=20 was, dan veranderde de merge tag naar 20, zoals te zien is in de onderstaande afbeelding.

Figuur 14: De tekst van de merge tag is veranderd naar 20

De Reward Generator was zo genoeg uitgewerkt om te beginnen met de verificatiefase van het project. In hoofdstuk 10 Afronding behandel ik de afronding en integratie van de Reward Generator met de gehele software.

9 Verificatie

In dit hoofdstuk geef ik aan hoe ik aan de verschillende requirements heb voldaan en hoe ik het systeem getest heb. Er zijn een aantal verschillende testtechnieken die ik had uitgevoerd. Deze zijn gebaseerd op de testtechnieken die in blok 11 IT Operations zijn behandeld. Ik had een aantal aanpassingen gemaakt in de fasering van de testen, aangezien niet alle technieken relevant waren voor dit project. De testen had ik voornamelijk handmatig uitgevoerd. Het project is een proof-of-concept, dus het automatiseren van de testen heeft geen extra waarde. De resultaten van dit hoofdstuk bepalen mijn werkzaamheden voor de afrondingsfase.



Figuur 15: De verschillende testtechnieken

Acceptatie Testing is een testtechniek die wordt uitgevoerd om te bepalen of het softwaresysteem aan de requirements heeft voldaan. Dit kan dus pas aan het einde van het project worden gedaan, vandaar dat ik deze testtechniek pas behandel in hoofdstuk 11 Evaluatie.

9.1 Unit Testing

Ik had geen gebruik gemaakt van unit testing. Unit testing is een testtechniek waar een functie een andere functie (unit) aanroept en controleert of de geretourneerde waardes overeenkomen met de verwachte output. Unit tests hebben de meeste toegevoegde waarde als ze vroeg in de timeline worden uitgevoerd.

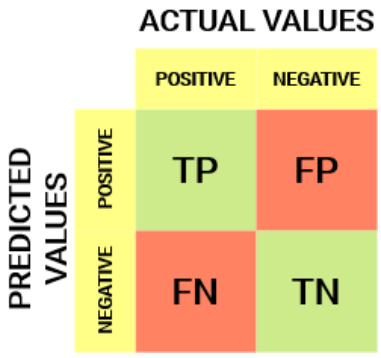
Voor mijn project vond ik dat Unit Testing niet nodig was. Ten eerste, wordt er in een Jupyter Notebook niet veel gebruik gemaakt van verschillende functies (units). Dit betekent dat er speciaal functies moeten worden gecreëerd om Unit Testing mogelijk te maken wat ten koste gaat van de beschikbare tijd. Ten tweede, was het voor mij in het begin van de timeline moeilijk de verwachte uitkomst van bepaalde functies te achterhalen, aangezien ik werkte in een nieuwe en

onbekende development omgeving. De verwachte uitkomsten kon ik pas later in de timeline achterhalen, nadat ik genoeg kennis had. Hierdoor was het dus niet mogelijk om Unit Testing vroeg in de timeline te implementeren, waardoor ze nog maar weinig toegevoegde waarde konden hebben.

9.2 Model Testing

Bij het testen van het model gaat het vooral om de nauwkeurigheid en betrouwbaarheid van de voorspellingen. De score die ik in de ML model iteraties van hoofdstuk 8 had behandeld was de accuracy score. Dit is het percentage goede voorspellingen. Echter, is dit niet altijd de beste manier om een model te testen. Bijvoorbeeld als er in een dataset van 100 dieren, 10 vogels zijn en 90 vissen, en het model voorspelt alles als vissen, heeft het alsnog een score van 90% terwijl er helemaal geen goede vogels zijn voorspelt.

Om dit op te lossen, had ik een confusion matrix gebruikt. De matrix vergelijkt de werkelijke waardes met de voorspelde waarde van het ML model. Dit geeft een beeld van hoe goed het model presteert en welke soorten fouten het maakt.



Figuur 16: Confusion matrix (Bhandar, 2020)

9.3 Integratie Testing

Bij Integratie Testing worden individuele componenten gecombineerd en getest als een groep. Het doel van deze techniek is om fouten in de interactie tussen geïntegreerde eenheden te vinden. In mijn project zijn er 2 componenten die met elkaar moeten integreren. Het ML model en de Reward Generator. Hiervoor had ik een manuele integratie test verricht.

9.4 Load Test

Een van de requirements die ik had gekregen was dat de responstijd van de voorspelling onder 2 seconden moet zijn. Hiervoor had ik een load test uitgevoerd. Ik was begonnen met handmatig 5 keer een voorspelling op te vragen aan het model. De responstijd heb ik in de onderstaande tabel verwerkt. De tabel laat zien dat de eerste request veel langzamer is dan de rest. Dit komt, omdat de endpoint nog moet

opwarmen. Dit was geen probleem, aangezien het nog steeds minder dan 2 seconden was.

Request Nr.	Responstijd (ms)
1	1198
2	656
3	656
4	846
5	619

Figuur 17: Resultaten handmatige test

Ik had daarna een real-life-scenario nagebootst met een tool genaamd Artillery.io. Ik had hiervoor gekozen, omdat het door Amazon werd aanbevolen voor het testen van AWS endpoints en het model was ook op een AWS endpoint gedeployed.

Gebaseerd op de cijfers van de e-mails waar het model op getraind was, was te zien dat het peak van het aantal e-mail openingen in het eerste uur lag. 25% van de ~1000 ontvangers openden dan de e-mail. Het aantal daalde erna drastisch. Het gemiddelde aantal requests in deze eerste uur was dus $\frac{250}{60} = 4\ requests\ per\ minuut$. Het maximum was niet te achterhalen.

Echter, was het voor een test niet nodig om een uur lang te wachten. Ik had een test gedaan die 1 minuut duurde en 4 request naar de endpoint maakte. De gemiddelde responstijd hiervan was 707 ms en de max 821 ms. Met 4 requests per minuut deed het systeem dus aan de requirement. Echter, is waarschijnlijk het maximaal aantal requests per minuut hoger dan 4. Dit wordt in het volgende scenario getest.

Een andere requirement was dat het systeem schaalbaar moest zijn. Om dit te realiseren werd al gebruik gemaakt van AWS Lambda, wat volgens Amazon schaalbaar is. Of dit echt zo was, moest wel nog getest worden. Hiervoor had ik een test gedaan waar er een minuut lang 4 request per seconde werden gemaakt naar het endpoint. Dit waren 60x meer requests dan de verwachte peak van de vorige test. De gemiddelde responstijd hiervan was 737 ms en de max 1004 ms. Deze tijden zijn niet veel hoger dan de vorige test en nog steeds onder de 2 seconden dus er werd voldaan aan beide requirements.

Als de responstijd niet zal voldoen aan de requirement, was een van de oplossingen om het aantal features te verwijderen. Dit zal wel ten koste gaan van de betrouwbaarheid van de voorspellingen. Maar dat is dan een afweging die zal moet worden gemaakt door de stakeholders.

10 Afronding

In dit hoofdstuk wordt de afronding van het project behandeld. Er wordt uitgelegd hoe de verschillende onderdelen zijn afgerond, het verloop van het experiment en de resultaten ervan. Ten slotte wordt er een conclusie getrokken en aangegeven of er mogelijkheden zijn voor een vervolgonderzoek.

10.1 Zoektocht naar bedrijven

In een van de laatste weken had ik uiteindelijk een overeenkomst kunnen sluiten met 2 bedrijven die bereid waren om mee te doen aan mijn opdracht. Het eerste bedrijf maakte geen gebruik van e-mail marketing, maar had wel een e-mail lijst van 4000 klanten. Het tweede bedrijf had een e-mail lijst van 500 klanten, maar gebruikte geen Mailchimp. Ik had besloten om eerst aan de slag te gaan met het bedrijf met 4000 klanten, aangezien hiermee meer data verzameld kon worden. Deze bedrijf kon ik ook Mailchimp aanbevelen, aangezien zij nog geen gebruik maakte van e-mail marketing. Daarnaast had ik al een model gemaakt voor Mailchimp dus wist ik bijna zeker dat het hiermee zal werken en wat ik kon verwachten.

Echter, liepen wij tegen een probleem aan bij het gebruiken van Mailchimp. Bij het versturen van de e-mails kregen we een melding van Mailchimp dat we hun services niet meer mochten gebruiken. Ik had onderzocht waarom dit was gebeurd en had hier een mogelijk antwoord op gevonden. We hadden een opwarmfase overgeslagen (Alex, 2020). Dit houdt in dat er langzaam naar steeds meer mensen een e-mail wordt verstuurd. Dus in week 1 wordt er maar naar 100 mensen een e-mail verstuurd, in week 2 naar 200, in week 3 naar 500 en zo verder. Wij hadden meteen een e-mail naar 4000 klanten geprobeerd te sturen, en hierdoor waren we verbannen van Mailchimp.

Verder gaan met Mailchimp was geen optie meer en we moesten op zoek naar een andere e-mail marketing provider. Echter, was het wegens de tijdnood niet mogelijk om met een andere provider door de opwarmfase te lopen. Dit was een probleem, aangezien we hier ook verbannen zullen worden als we deze fase zullen overslaan. Als gevolg had ik mijn samenwerking met deze bedrijf moeten beëindigen.

De tweede bedrijf maakte wel al gebruik van e-mail marketing dus hier hoefden we geen opwarmfase te houden. Deze bedrijf gebruikte wel ActiveCampaign in plaats van Mailchimp, waardoor ik alle stappen opnieuw moest doorlopen met een nieuwe API en dataset. Dit ging veel sneller dan de eerste keer, aangezien ik het allemaal al en keer had gedaan.

10.2 Dataverzameling

Voor het verzamelen van gelabelde data, had ik met het samenwerkende bedrijf 2 e-mails verstuurd naar haar klanten. Het bedrijf wilde geen

gebruik maken van ReviewByMe, maar van een eigen systeem. Dit was geen probleem, aangezien dit systeem de data ook bijhield. In de 2 e-mails, is er een taak dat moet worden uitgevoerd in ruil voor punten. Deze taak en het aantal punten verschilt voor de 2 e-mails.

Er was geen e-mail gestuurd met een taak zonder beloning wat wel het plan was, aangezien het bedrijf dit niet wilde. Dit was ook geen probleem, alleen kon het effect van de aanwezigheid van een beloning niet worden vergeleken met wanneer deze afwezig was. Bij het opstellen van de dataset waren er geen verhoudingsproblemen, aangezien een groot percentage van de klanten, de taken hadden uitgevoerd.

10.3 Uiteindelijke ML model

Voor het uiteindelijke ML model voor het nieuwe scenario had ik de volgende heuristische oplossing opgesteld: de top 25% klanten die het langst lid zijn, gaan de actie wel uitvoeren. De score van deze oplossing was 52%. Deze moest ik dus proberen te overtreffen om het model een succes te verklaren. Ik had de XGBoost algoritme gebruikt, omdat ik hier het meeste succes mee had in de voorgaande modellen. Uiteindelijk had ik een score van 98% behaald. Het model was een succes en het was dus niet nodig om te proberen deze score nog hoger te krijgen.

10.4 Uiteindelijke Reward Generator

Voor het genereren van een beloning was er een basis aantal punten dat bepaald was door een van de marketeers van het bedrijf waar ik samen mee ging werken. Dit aantal was 100 punten. Wat er daarna gebeurde was dat aan deze 100, extra punten werden toegekend afhankelijk van de voorspelling die het ML model had gemaakt over de kans dat iemand de taak zal uitvoeren voor 100 punten.

In het algemeen, levert een grotere beloning en grotere kans op conversie (Dube, 2019). Vandaar dat ik ook punten had toegevoegd aan de basisbeloning in plaats van punten eraf halen. De grotere beloningen moesten theoretisch het totale respons verhogen, waardoor ik meer data zal hebben om een conclusie te trekken. In het algemeen geldt hoe hoger de respons, hoe meer betrouwbaar de resultaten zijn (SurveyMonkey, 2020; Bullen). Als de kans dat iemand de taak uitvoert laag was, dan kreeg iemand een groter beloning in hoop dat deze zal leiden tot het uitvoeren van de taak. Als de kans hoog was, dan kreeg iemand een lager beloning, aangezien deze waarschijnlijk toch al de taak zal uitvoeren.

Het maximaal punten dat iemand kon krijgen was 2x het basisaantal. Deze limiet was opgesteld door het bedrijf waar ik samen mee ging werken.

Nadat de extra punten werden toegevoegd aan de basisbeloning, werd het getal afgerond op de meest dichtbij zijnde product van 5, zodat het een minder onbekend getal was. Uit mijn deskresearch was gebleken dat

psychologie een belangrijke rol speelt op conversie (M., 2020). Mensen hebben een voorkeur voor het bekende (Vries, Erasmus, & Gerber, 2017). Vandaar dat ik geprobeerd had de getallen bekender te maken, zonder teveel in te grijpen op de personalisatie.

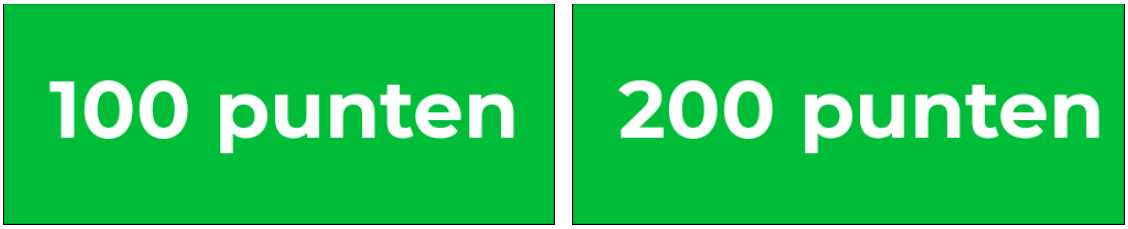
Op https://www.joinhoney.com/ (Honey, 2020) zijn populaire kortingen te zien. Ik had hier gezien dat de meeste kortingen een product van 5 zijn, dus 5%, 10%, 15% of 20%. Vandaar dat ik deze afronding had gebruikt. Zo hield het genereren van de beloning, rekening met psychologie.

De formule voor het berekenen van de beloning is in een versimpelde Excel formaat hieronder opgenomen.

$$Round(BaseReward + ((1 - Probability) * BaseReward); 5)$$

- BaseReward is in dit geval 100
- Probability is de kans dat iemand toch al de taak zal uitvoeren voor het basisaantal punten
- Het aantal wordt afgerond op een product van 5

Ik had de Reward Generator gedeployed op een AWS Lambda functie. De endpoint van deze functie kon nu worden gebruik in een ActiveCampaign mail met de volgende structuur: rewardgenerator.com?amount=%USERBELONING%. Als deze link in een e-mail werd geplaatst, werd %USERBELONING% vervangen met de beloning die was opgeslagen in het profiel van de klant en kreeg de klant hier een afbeelding van te zien. Hieronder zijn 2 afbeeldingen te zien van deze functie in actie. De eerste afbeelding is van een klant die als beloning 100 punten kreeg, de tweede van een klant die 200 punten als beloning kreeg.



Figuur 18: De gegenereerde afbeelding van de beloning voor 2 verschillende klanten

10.5 Experiment

Voor het uiteindelijke experiment zijn er 2 e-mails verstuurd met een A/B-test. De eerste groep was een controlegroep waar de beloning niet was aangepast door het ML model. Iedereen in deze groep kreeg dus een beloning van 100 punten. In de tweede groep, was de beloning wel voor elke klant aangepast door het ML model. De splitsing van de groep heeft een 50/50 verhouding. Afwijken van een 50/50 splitsing verlaagt de statistische kracht (Mark H. White II, 2018).

10.5.1 Resultaten conclusie en discussie

Groep	Grootte	Conversies	Conversiepercentage	Kosten per conversie
Controle	232	57	25%	100
ML	233	47	20%	134

Figuur 19: Resultaten van het experiment

Dit is niet wat ik had verwacht. Machine Learning heeft slechter gepresteerd dan de controlegroep. In paragraaf 8.1.4 Probleem en oplossing heb ik gedefinieerd wanneer het model een succes is: Het model is een succes wanneer de gemiddelde kosten per conversie relatief minder stijgen dan het aantal conversies of andersom.

Echter, is het aantal conversies lager en de kosten per conversie hoger dan de controle groep. Het ML model is dus geen succes. Bij het analyseren van de resultaten heb ik een aantal punten ontdekt waardoor de ML groep mogelijk slechter heeft gepresteerd dan de controlegroep.

Als eerste heb ik gekeken naar de steekproefomvang. Een uitspraak over de gehele populatie (alle mensen die een e-mail beloning ontvangen) is heel zwak, aangezien de steekproefomvang veel te klein is. De populatie heeft een grootte van in de miljoenen en misschien zelfs miljarden. Daarnaast, is de steekproef, ook niet representatief voor de gehele populatie, aangezien het niet alle bevolkingsgroepen bevat. In dit geval, zijn de 465 klanten vooral mannen met een leeftijd van tussen de 25-55. Er kan wel een uitspraak worden gedaan, over de klanten van het bedrijf waar de resultaten op gebaseerd zijn.

De error marge van de individuele groepen is hoger dan wat ik definieer als acceptabel. Voor de controlegroep is het 11% en voor de ML groep 13% (Conroy; Survey Monkey, 2020). Dit betekent dat het daadwerkelijke conversiepercentage voor de controlegroep tussen de 14% en 36% ligt en voor de ML groep tussen de 7% en 33%. Dit vind ik niet acceptabel, aangezien er een factor van meer dan 2 tussen de ondergrens en bovengrens zit, waardoor er geen betrouwbare conclusie kan worden getrokken.

Overigens, moet het minimale conversies per groep 100 zijn om enig zinvol resultaat te krijgen (Bullen). Met het huidige conversiepercentage betekent dit dat de groepsomvang ongeveer twee keer zo groot moet zijn.

Naast de steekproefomvang, kunnen er nog wat aanpassingen worden gemaakt aan het ML model. De maximale beloning waarmee het model getraind was, is 50. Hierdoor ziet het model bij het voorspellen van de kans dat iemand de taak gaat uitvoeren, voor een beloning van 100, geen verschil tussen 100 en 50 punten. De 100 punten zijn voor het model evenveel als 50 punten. Hier kwam ik pas achter tijdens het analyseren van de resultaten.

Maar zelfs als het ML model wel het verschil zag tussen de 50 en 100, is niet alles opgelost. Uit de resultaten is te zien dat de ML groep een grotere kosten per conversie heeft en minder aantal conversies. Een van de hypotheses die ik had opgesteld was dat het verhogen van de beloning het aantal conversies en het conversiepercentage verhoogt. De beloning voor de ML groep is gemiddeld hoger maar er zijn minder conversies. Dit zijn twee minpunten en heeft nog niks met Machine Learning te maken, er is hier iets anders aan de hand dan toeval wegens kleine groepsomvang.

Ik heb het eerder gehad over dat psychologie een belangrijke rol speelt op conversie (M., 2020; Clarke, 2019; Gordon-Hecker, Pittarello, Shalvi, & Roskes, 2019). Dit kan hier een belangrijker rol hebben gespeeld, dan dat ik had verwacht. Iedereen in de controlegroep kreeg een beloning van 100 punten. Bij de ML groep lag dit tussen de 100 en 200. De beloningen van de ML groep, zoals 110, 175 en 200 zijn waarschijnlijk minder vaak voorkomend dan 100. Zoals eerder besproken, hebben mensen een voorkeur voor het bekende (Vries, Erasmus, & Gerber, 2017). Het aantal conversies van de ML groep is dus waarschijnlijk lager, wegens de minder bekende getallen.

De tweede hypothese die ik had opgesteld was dat het gebruik van ML voorspelde beloningen het aantal conversies en conversiepercentage verhoogt. De resultaten laten zien dat dit niet waar is. Het conversiepercentage van de ML groep is zelfs lager. Dit zijn weer twee minpunten waardoor ik toch denk dat dit niet te maken heeft met toeval wegens de kleine groepsomvang. Dit komt waarschijnlijk ook omdat er niet genoeg rekening is gehouden met de psychologische factor.

De laatste hypothese die ik had geformuleerd is dat gebruik van ML voorspelde beloningen een lagere kosten per conversie oplevert. Uit de tabellen is te zien dat de kosten per conversie van de ML groep hoger zijn, maar dit komt wegens de aanpak die ik had gekozen: het toevoegen aan de basis beloning. Als ik juist punten ging weghalen van de beloning, of als iedereen in de controlegroep, de maximum beloning kreeg, waren de kosten per conversie van de ML groep lager. De controlegroep heeft dan een kosten per conversie van 200 en de ML groep 134, wat lager is dan de controle groep. De grotere beloningen moesten theoretisch het totale respons verhogen. Echter, hebben de grotere beloningen, niet een positief effect gehad op de respons, zoals te lezen is in de conclusie van de eerste hypothese.

Met het experiment heb ik dus niet aan kunnen tonen dat Machine Learning de effectiviteit van e-mail marketing promoties kan verhogen. Dit heeft te maken, omdat er in de gekozen aanpak niet genoeg rekening is gehouden met psychologie. Zie de “Vervolgonderzoek” bijlage voor een beknopte aanpak voor een vervolgonderzoek die meer rekening houdt met psychologie.

11 Evaluatie

In dit hoofdstuk geef ik een oordeel over het product, proces en de resultaten hiervan. Ik trek conclusies voor het vervolg van mijn carrière en geef aan hoe ik de verschillende beroepstaken heb behaald en op welke niveau.

11.1 Acceptatie Testing (Requirements)

Acceptatie Testing is een testtechniek die wordt uitgevoerd om te bepalen of het softwaresysteem aan de requirements heeft voldaan.

ID	Requirement	Voldaan en uitleg
R1	Het systeem maakt gebruik van Machine Learning	**Ja.** Het systeem maakt gebruik van Machine Learning om een voorspelling te doen over de kans dat een klant voor een bepaalde beloning gaat converteren.
R2	Het systeem maakt gebruik van AWS	**Ja.** Het systeem maakt gebruik van een aantal AWS-diensten waarvan AWS SageMaker en AWS Lambda de twee meest gebruikte zijn.
R3	Het systeem maakt gebruik van de e-mail marketing kanaal	**Ja.** Het ML model is gebouwd voor e-mail marketing promoties en het hele systeem werkt op dit moment alleen met de e-mail marketing provider ActiveCampaign.
R4	Het systeem is schaalbaar	**Ja.** Het systeem is inderdaad schaalbaar en dit wordt behandeld in 9.4 Load Test.
R5	Het systeem genereert beloningen real-time	**Nee.** De beloningen worden niet volledig real-time genereert. Dit was wegens een beperking van ActiveCampaign. Het genereren van de afbeelding is nog wel real-time, maar de beloning is al van te voren bepaald Dit behandel ik in 11.2.3 Ontwerpen en 11.2.4 Reward Generator.
R6	Het systeem geeft binnen 2 seconden een gegenereerde beloning weer	**Ja.** Het systeem geeft inderdaad binnen 2 seconden een gegenereerde beloning weer en dit wordt behandeld in 9.4 Load Test.
R7	De Reward Generator is ontwikkeld in Node.js	**Ja.** De Reward Generator is een AWS Lambda functie die is geschreven in Node.js.
R8	De Reward Generator genereert SVG plaatjes	**Ja.** De Reward Generator genereert inderdaad SVG plaatjes. Zie 8.2.6 Deployment voor meer informatie.
R9	Het systeem heeft een user interface. (Won't Have)	**Ja.** Het systeem heeft inderdaad geen user interface. In een Jupyter Notebook worden de beloningen gegenereerd via code. De e-mails met de beloningen moeten handmatig via ActiveCampaign worden verstuurd. ActiveCampaign heeft wel een interface maar dat is niet mijn product.
R10	Het Machine Learning model kan integreren met andere software	**Ja.** Het ML model kan integreren met andere software, aangezien het een endpoint heeft. Aan deze endpoint kunnen

		voorspellingen worden gevraagd via GET API calls.
R11	Het systeem kan beloningen in e-mails personaliseren	**Ja**. Het systeem kan beloningen in e-mails personaliseren. Een voorbeeld hiervan is te zien in 10.4.
R12	Het systeem haalt de data op van ReviewByMe	**Nee**. Het systeem haalt de data NIET op van ReviewByMe. Dit was niet nodig en hier is meer over te lezen in 8.2.2 ML model iteratie #1.
R13	Het systeem haalt de data op van Mailchimp	**Nee**. Het uiteindelijke systeem maakt gebruik van ActiveCampaign. De eerdere modellen haalde wel de data op van Mailchimp. Waarom dit veranderd is, is te lezen in 8.2.2 ML model iteratie #1
R14	Het Machine Learning model is gebouwd in AWS SageMaker	**Ja**. Het ML model is inderdaad gebouwd in AWS SageMaker. Dit wordt behandeld in 8.1.8 Model deployment en gebruik en 8.2.6 Deployment.
R15	Het Machine Learning model traint met een gebalanceerd dataset	**Ja**. Het uiteindelijke ML model is inderdaad getraind met een gebalanceerd dataset. Dit wordt behandeld in 10.2 Dataverzameling.
R16	Het Machine Learning model maakt gebruik van demografische gegevens	**Nee**. Het uiteindelijke model maakt geen gebruik van demografische data, aangezien deze niet beschikbaar was. De eerdere modellen maakte hier wel gebruik van. Het uiteindelijke systeem maakt gebruik van ActiveCampaign. ActiveCampaign houdt in tegenstelling tot Mailchimp geen demografische gegevens bij en deze zelf voorspellen was geen optie, aangezien er geen gelabelde data beschikbaar was.
R17	Het Machine Learning model verandert non-numerieke waardes naar numeriek	**Ja**. Dit wordt inderdaad gedaan. Zonder deze stap is het onmogelijk voor een ML algoritme een voorspelling te doen. Er wordt een error weergeven als het ML model non-numeriek data binnen krijgt.
R18	Het Machine Learning model normaliseert numerieke waardes in de dataset	**Ja**. De numerieke waardes in de dataset zijn inderdaad genormaliseerd.
R19	Het Machine Learning model plaatst numerieke waardes in de dataset in bins	**Nee**. Voor het uiteindelijke model is dit niet gedaan, aangezien het niet nodig was. Bij de eerdere modellen was dit wel gedaan voor de coördinaten feature.
R20	Het Machine Learning model heeft een hogere accuracy score dan een heuristische oplossing	**Ja**. Het uiteindelijke model heeft een score van 98% en de heuristische oplossing een score van 52%. Het model heeft een hogere score dus aan deze requirement is voldaan.

Figuur 20: Acceptatie Testing tabel

11.2 (Tussen)producten

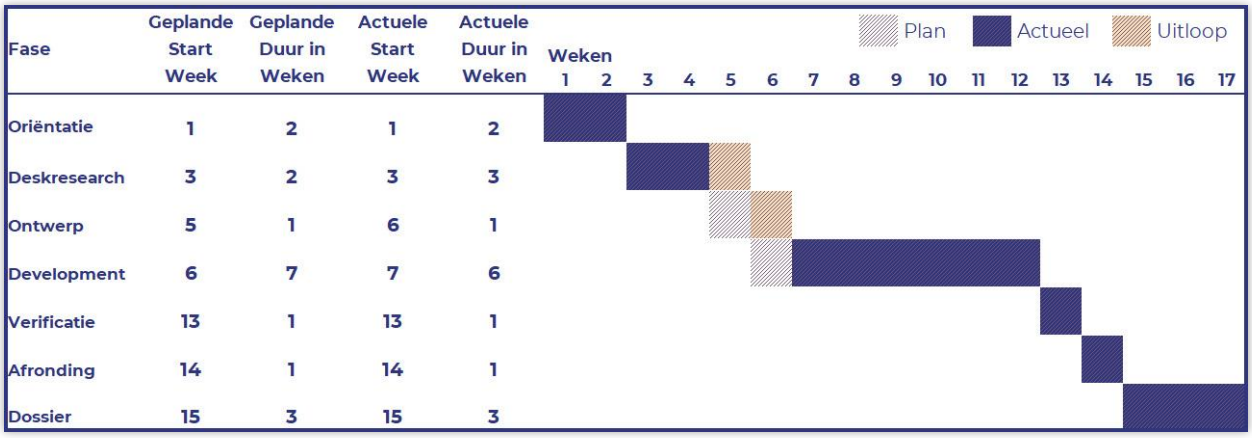
In dit hoofdstuk beoordeel ik de tussentijds gemaakte en opgeleverde producten. Ik beschrijf in hoeverre de producten bijdragen aan het doel waarvoor ze opgesteld zijn.

11.2.1 Plan van aanpak

Het plan van aanpak was het eerste product dat ik had opgesteld. Het doel ervan was om structuur en visie aan het project te geven. Het product is hierin geslaagd. Bij het opstellen van het plan van aanpak had ik wel de nadruk gelegd op deskresearch. Hierdoor bevat het kenmerken van een Research Proposal. Een verbeterpunt is om voor de Research Proposal een apart document op te stellen.

11.2.2 Deskresearch rapport

Een rapport van mijn bevindingen tijdens de deskresearch fase was het tweede product dat ik had opgesteld. Het doel hiervan was om mij voor te bereiden op de development-fase. Het product is hier gedeeltelijk in geslaagd. Tijdens de development-fase, moest ik toch nog wat deskresearch verrichten, aangezien ik niet alles had onderzocht in de deskresearch fase. Dit was zelfs nadat ik een week extra besteed had aan deskresearch, omdat ik nog veel moest onderzoeken. Ik had de tijd die ik nodig had voor deskresearch, onderschat. Doordat ik een week meer heb besteed aan deskresearch, heb ik een week minder besteed aan de development-fase, omdat ik hier de meeste tijd voor had ingepland. Hieronder is de Gantt chart te zien met de geplande en actuele duur van de fases.



Figuur 21: Gantt chart met de geplande en actuele duur van de fases

Echter, denk ik dat het moeilijk is om 100% voorbereid te zijn. Een verbeterpunt is misschien om een kortere deskresearch fase te houden, aangezien ik toch weer onderzoek ga moeten verrichten.

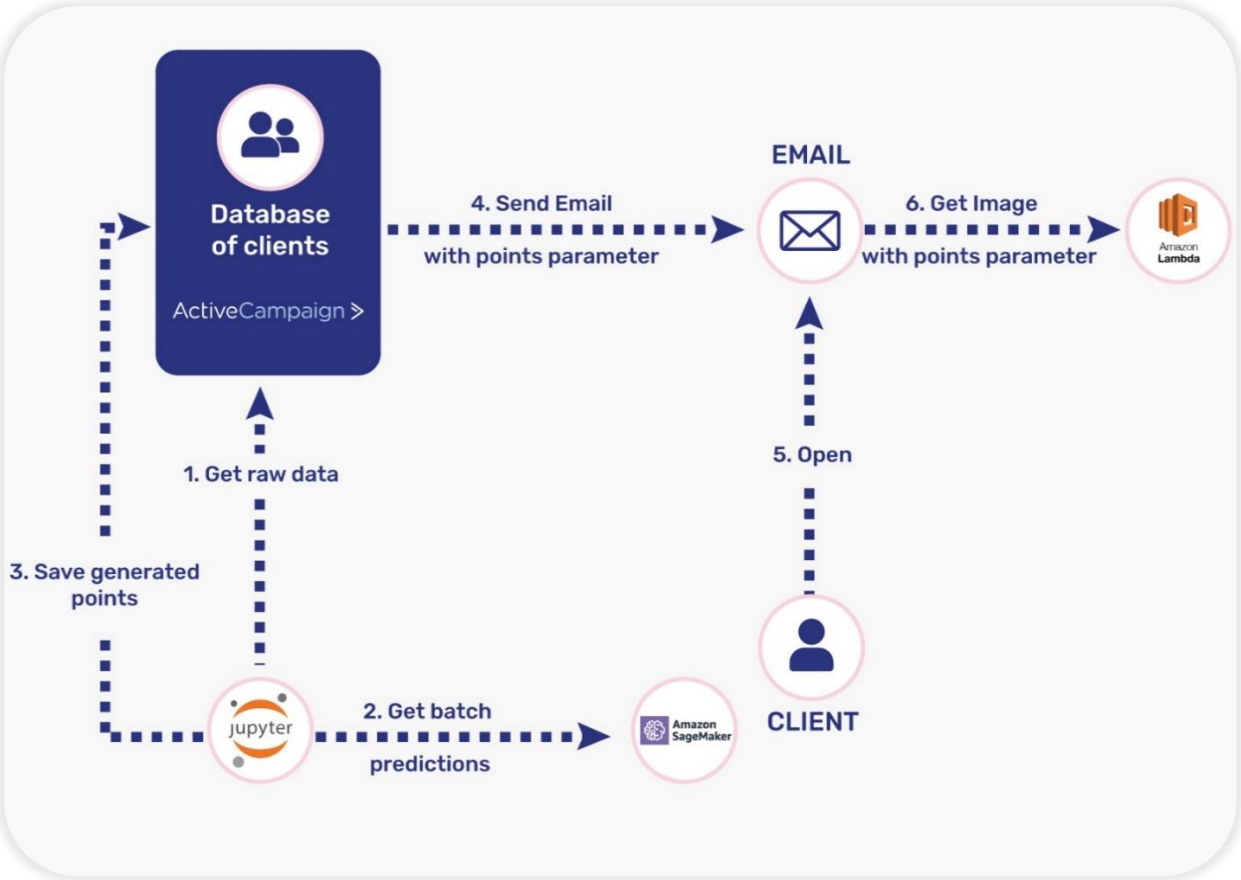
11.2.3 Ontwerpen

De ontwerpen waren opgesteld met als doel mij een plan te geven voor de development-fase. Ze zijn hierin geslaagd. Echter, is het uiteindelijke ontwerp anders dan het oorspronkelijke (zie figuur 21).

Ik heb bij de integratie met ActiveCampaign een aantal ontwerpkeuzes moeten maken die anders zijn dan oorspronkelijk gepland. ActiveCampaign heeft een limiet aan hoeveel calls er naar de API gemaakt kunnen worden. Dit is 5 per seconden. Gebaseerd op de resultaten van de load test, gaan er niet zo snel 5 mensen tegelijk een e-mail openen waardoor dit limiet niet snel bereikt zal worden.

Echter, wordt er niet 1 call per klant naar de ActiveCampaign gemaakt. Voor het ophalen van de ruwe gegevens van een contact, moeten meerdere calls worden gemaakt en ook voor het opslaan van de gegenereerde beloning moet een call worden gemaakt. Hierdoor is er een groot kans dat de limiet van 5 requests per minuut wordt bereikt. Dit kan al gebeuren als 2 mensen de e-mail in dezelfde seconde openen. Dit wordt een probleem, aangezien dan alleen de eerste klant op tijd de beloning ziet. Als dit gebeurt, wordt er niet meer voldaan aan de requirement.

Ik heb dit opgelost door vlak voor het versturen van de e-mail een batch voorspelling te maken voor alle klanten. De beloningen worden dan met een API call toegevoegd aan het ActiveCampaign profiel van de klanten. De beloning wordt dus al bepaald voor het versturen van de e-mail in plaats van op het moment van openen. Het genereren van de afbeelding is nog wel real-time. Het nadeel van deze methode is, dat de data en hierdoor de voorspelde score en beloning tussen de batch voorspellingen en het openen van de mail kan veranderen. Echter, is de kans dat dit gebeurt klein, aangezien de meeste features die het model gebruikt, niet snel veranderen, zoals het adres of leeftijd van een klant. Daarnaast is de tijd tussen de twee momenten niet veel.



Figuur 22: Versimpelde architectuur van het uiteindelijke systeem

11.2.4 Reward Generator

Het doel van de Reward Generator was om een persoonlijke aanbieding in de vorm van een plaatje in e-mails aan klanten te weergeven. Deze is voldaan. Echter, is het uiteindelijke ontwerp, anders dan de oorspronkelijke planning. Het oorspronkelijke idee was om de Reward Generator een ID van een klant mee te geven. De Reward Generator zal dan met deze ID een call maken naar de API van de e-mail marketing provider om de data op te halen. Vervolgens zal de Reward Generator deze data versturen naar de endpoint van de ML model om een voorspelling te maken.

Echter, was deze aanpak niet mogelijk door beperkingen van de API van de uiteindelijke e-mail marketing provider, ActiveCampaign, die ik moest gebruiken. Met de oplossing die ik heb geïmplementeerd, krijgt de Reward Generator direct het aantal punten en maakt geen calls naar de endpoint van de ML model of de ActiveCampaign API. Het genereren van de afbeelding is nog wel real-time, maar de beloning is al van te voren bepaald. Deze aanpak heeft geen merkbare nadelen. Hier is meer over te lezen in de vorige paragraaf (11.2.3).

11.2.5 ML model

Het ML model maakt de voorspellingen. Het originele idee was om een score te genereren tussen de 0 en 1 en dit wordt ook gedaan. Echter, zijn de scores van het uiteindelijke model heel extreem. Dus ze zijn of heel dicht bij de 0 of heel dichtbij de 1. Hierdoor zijn er totaal minder verschillende scores, dus ook minder verschillende beloningen.

De meeste tijd was ik verloren aan het opstellen van mijn dataset. Het trainen van een model met verschillende algoritmes was zo gedaan. Daarnaast was het nog puzzelen om het ML model beschikbaar te stellen voor externe software en voorspellingen met deze externe software op te vragen.

11.2.6 Testrapport

Dit rapport weergeeft of het systeem voldoet aan de vereiste en of het inderdaad functioneert. Het is geen volledig apart document, maar hoofdstuk 9 van dit verslag. Mijn opdrachtgevers, vonden een apart document niet nodig.

11.2.7 Experiment

Voor het experiment zijn er, zoals gepland, drie e-mails verstuurd. Echter, heeft er maar 1 split test plaatsgevonden en is er geen gebruik gemaakt van ReviewByMe.

Het idee voor de eerste e-mail was dat het een RVBM enquête bevatte zonder beloning. De uiteindelijke e-mail bevatte een taak in ruil voor punten. Het bedrijf waar ik samen mee ging werken, wilde geen gebruik maken van ReviewByMe en wilde perse een beloning geven in ruil voor het verrichten van een taak.

Het idee voor de tweede e-mail was om een A/B-test te houden, waarbij de helft van de klanten allemaal een enquête zal ontvangen met dezelfde beloning en de andere helft een e-mail zonder de beloning. Dit was ook niet mogelijk, aangezien het bedrijf waar ik samen mee ging werken, geen taken wilden versturen zonder een beloning. De tweede e-mail bevatte uiteindelijk ook een taak (wel een andere) in ruil voor punten (niet hetzelfde aantal als de vorige taak).

Het idee voor de derde e-mail was om een A/B/C-test te houden, maar dit was niet mogelijk wegens de kleine omvang van de groep. Voor het uiteindelijke experiment was er een A/B-test gehouden. De eerste groep was een controlegroep en ontving een e-mail waar de beloning niet was aangepast door het ML model. Iedereen in deze groep kreeg dus dezelfde beloning. In de tweede groep, was de beloning wel voor elke klant aangepast door het ML model.

11.2.8 Eindresultaat

Het eindresultaat bestaat uit een systeem, dat in staat is om beloningen in e-mails die verstuurd worden via ActiveCampaign, te

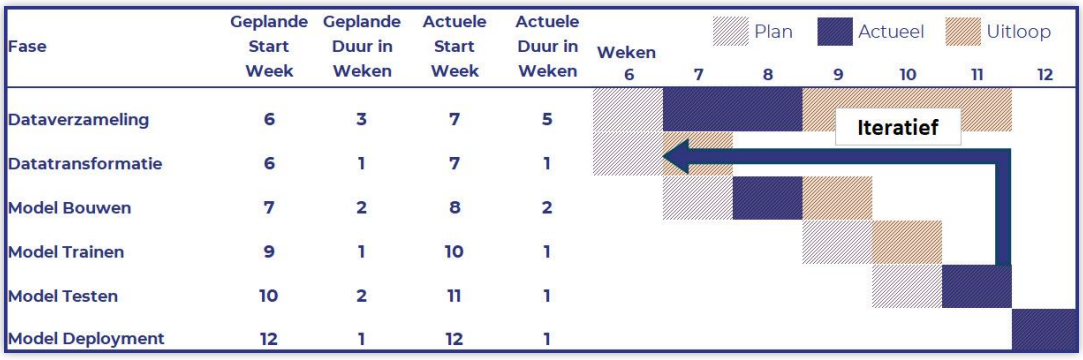
personaliseren voor de klanten. Het systeem maakt gebruik van Machine Learning om deze beloning te personaliseren.

Naast het ontwikkelde systeem, is er antwoord gegeven op de onderzoeksvraag en is dit verslag onderdeel van het eindresultaat.

11.3 Proces reflectie

In deze paragraaf reflecteer ik op onderdelen van het proces die niet paste bij de voorgaande paragraaf 11.2 (Tussen)producten.

Het vinden van bedrijven die bereid waren om deel te nemen aan het experiment, was veel moeilijker dan dat ik verwacht had. De volgende keer ga ik dit veel eerder doen. Echter, heeft dit niet veel invloed gehad op de planning van mijn development-fase. Ik had alle sub-fases op tijd kunnen uitvoeren. Hieronder is de Gantt chart te zien met de geplande en actuele duur van de sub-fases. De development-fase begint een week later, omdat ik een week langer heb besteed aan de deskresearch fase (dit wordt behandeld in 11.2.2). Hierdoor had ik ook de development-fase ingekort om alsnog op schema te lopen. De sub-fase die ik ingekort had was "Model Testen", aangezien deze ook in één week kon worden afgerond.



Figuur 23: Gantt chart met de geplande en actuele duur van de development-fase.

11.4 Beroepstaken

In dit hoofdstuk laat ik per beroepsproduct die ik in het afstudeerplan had opgenomen, zien dat ik deze adequate beheers. Dit wordt gedaan door de taken op te noemen die ik heb uitgevoerd om de beroepstaken te realiseren. Ik ga waar het mogelijk is verwijzen naar voorgaande hoofdstukken.

11.4.1 Verplichte beroepstaken

- A-1 Analyseren probleemdomein & opstellen probleemstelling – Niveau = 3

Ik heb helder in kaart gebracht waar, wanneer en waarom welke problemen opdoen en wat de gevolgen hiervan zijn. Ik heb de mogelijke risico's en de factoren die daarbij een rol spelen ook meegenomen. Dit heb ik zelfstandig gedaan. Dit is terug te lezen in het hoofdstuk 3

Making Offers Intelligent.

- G-c Kritisch, onderzoekend & methodisch werken – Niveau = 3

Ik heb voor het systeem en onderzoek en voor mijzelf meetbare kwaliteitsnormen gehanteerd. Ik heb vragen opgesteld en tijdens het op zoek naar antwoorden ben ik gestandaardiseerde methoden gaan gebruiken. Dit is terug te lezen in de hoofdstukken 5 Deskresearch en 7 Ontwerp.

- G-f Leren leren: voorbereiden op volgende studiefase en beroep – Niveau = 3

Ik heb tijdens het project mijn eigen leerproces een vorm gegeven. Ik heb veel geleerd over een technologie dat steeds populairder wordt. Ik heb gemerkt, dat de technologie zo snel veranderd, dat er onderdelen waren die in het begin van mijn afstuderen nog nieuw en voor het einde al verouderd waren. Dit heeft mij voorbereid op life-long-learning. Dit is terug te lezen in hoofdstuk 11 Evaluatie.

11.4.2 Aanvullende beroepstaken

- A-2 Informatie vergaren, analyseren, beoordelen & verwerken – Niveau = 3

Ik heb onderzoek verricht en informatie methodisch verzameld waarbij ik gemotiveerde keuzes heb gemaakt. De informatie heb ik vervolgens geanalyseerd, gerapporteerd en verwerkt in het eindproduct. Ook heeft deze informatie mij richting gegeven voor de vervolgstappen van het project. Dit is terug te lezen in de hoofdstukken 5 Deskresearch, 7 Ontwerp, 8 Development en 10 Afronding.

- D-15 Testen– Niveau = 3

Ik heb getoetst of het systeem voldoet aan de opgestelde requirements. Hierbij ben ik planmatig en volgens best practices te werk gegaan. Waar het mogelijk was, heb ik gebruik gemaakt van automatisering. Ik heb een gemotiveerde afweging gemaakt van de analyses die ik tijdens het testen heb uitgevoerd. Het resultaat heeft mijn vervolgstappen bepaald. Dit is terug te lezen in hoofdstuk 9 Verificatie.

- G-e Innovatief en creatief werken – Niveau = 4

Bij dit project was creativiteit van zeer hoge belang. Ik heb buiten het eigen referentiekader een systeem ontwikkeld dat betrekking heeft tot een bloeiende technologie. Het is een toekomstgericht product dat nu ingezet kan worden, maar beter wordt in de toekomst. Het systeem is innovatief en is op de gekozen manier niet eerder ontworpen. De voorspellingen van het model worden op een creatieve manier gebruikt. Ik heb ook op een creatieve en innovatieve manier 406 bedrijven benaderd. Ik heb problemen, zoals de beperkingen van ActiveCampaign, op een creatieve en innovatieve manier opgelost. Mijn innovatieve en creatieve manier van werken is terug te lezen in mijn aanpak in hoofdstuk 8 Development en 10 Afronding.

12 Bronnenlijst

Alex. (2020). *Deliverability in your first 30 days*. Opgehaald van ActiveCampaign: https://help.activecampaign.com/hc/en-us/articles/115000045650-Deliverability-in-your-first-30-days

AWS Online Tech Talks. (2020, February 27). *Amazon SageMaker Studio Deep Dive - AWS Online Tech Talks*. Opgehaald van YouTube: https://www.youtube.com/watch?v=pGhn8Ax8QmQ

Bernazzani, S. (2020). *Customer Loyalty: The Ultimate Guide*. Opgehaald van HubSpot: https://blog.hubspot.com/service/customer-loyalty

Bhandar, A. (2020). *Everything you Should Know about Confusion Matrix for Machine Learning*. Opgehaald van Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model.

Brownlee, J. (2017, July 24). *How Much Training Data is Required for Machine Learning?* Opgehaald van Machine Learning Mastery: https://machinelearningmastery.com/much-training-data-required-machine-learning/

Brownlee, J. (2019, December 5). *A Tour of Machine Learning Algorithms*. Opgehaald van Machine Learning Mastery: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Brownlee, J. (2019, December 20). *What is Deep Learning?* Opgehaald van Machine Learning Mastery: https://machinelearningmastery.com/what-is-deep-learning/

Brownlee, J. (2020). Opgehaald van https://machinelearningmastery.com/start-here/

Brownlee, J. (2020, March 15). *8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset*. Opgehaald van Machine Learning Mastery: https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Brownlee, J. (2020). *Why you should be Spot-Checking Algorithms on your Machine Learning Problems*. Opgehaald van Machine Learning Mastery: https://machinelearningmastery.com/why-you-should-be-spot-checking-algorithms-on-your-machine-learning-problems/

Brunson, R. (2018, April 13). *The Hook, The Story, And The Offer*. Opgehaald van Marketing Secrets: https://marketingsecrets.com/the-hook-the-story-and-the-offer/

Bullen, P. B. (sd). *How to choose a sample size (for the statistically challenged)*. Opgehaald van tools4dev: http://www.tools4dev.org/resources/how-to-choose-a-sample-size/#:~:text=A%20good%20maximum%20sample%20size,%2C%2010%25%20would%20be%2020%2C000.

Chapman, S. (2019). *JavaScript and Emails*. Opgehaald van ThoughtCo: https://www.thoughtco.com/javascript-and-emails-2037682

Clarke, G. (2019). *Should You Offer 50% Off or 'Buy One, Get One Free'?* Opgehaald van Medium: https://medium.com/better-marketing/should-you-offer-50-off-or-buy-one-get-one-free-e7fd8d17df37

Comarch. (2019). *Can We Identify Loyalty Fraud with the Use of Machine Learning?* Opgehaald van Comarch: https://www.comarch.com/trade-and-services/loyalty-marketing/blog/can-we-identify-loyalty-fraud-with-the-use-of-machine-learning/

Conroy, R. (sd). *Sample size A rough guide*. Opgehaald van http://www.beaumontethics.ie/docs/application/samplesizecalculation.pdf

D., C. (2020, March 25). *Best Python Libraries for Machine Learning and Deep Learning*. Opgehaald van Towards Data Science: https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c

Dube, S. (2019). *11 E-Commerce Discount Pricing Strategies That Will Increase Conversion Rate of Your e-Commerce Website*. Opgehaald van Invesp: https://www.invespcro.com/blog/ecommerce-discount-pricing-strategies/

Emarsys. (2020). *The Bridge Between Data and Personalization*. Opgehaald van Emarsys: https://emarsys.com/white-paper/the-bridge-between-data-and-personalization-2/

Ferlitsch, A. (2019). *Making the machine: the machine learning lifecycle*. Opgehaald van Google Cloud: https://cloud.google.com/blog/products/ai-machine-learning/making-the-machine-the-machine-learning-lifecycle

Google. (2019, July 11). *Data Preparation and Feature Engineering for Machine Learning*. Opgehaald van Google: https://developers.google.com/machine-learning/data-prep/transform/bucketing

Google. (2019, July 12). *Introduction to Machine Learning Problem Framing*. Opgehaald van Google: https://developers.google.com/machine-learning/problem-framing

Google. (2019). *Introduction to Machine Learning Problem Framing*. Opgehaald van Google Developers: https://developers.google.com/machine-learning/problem-framing

Google. (2020). *Machine learning training for marketers*. Opgehaald van Think with Google: https://www.thinkwithgoogle.com/feature/machine-learning-101-training/

Google. (2020). *Normalization*. Opgehaald van Googe Developers: https://developers.google.com/machine-learning/data-prep/transform/normalization

Gordon-Hecker, T., Pittarello, A., Shalvi, S., & Roskes, M. (2019). *Buy-One-Get-One-Free Deals Attract More Attention than Percentage Deals*. Opgehaald van ResearchGate: https://www.researchgate.net/publication/331585386_Buy-One-Get-One-Free_Deals_Attract_More_Attention_than_Percentage_Deals

Honey. (2020). Opgehaald van https://www.joinhoney.com/

Kharkovyna, O. (2019). *Top 10 Machine Learning Algorithms for Data Science*. Opgehaald van Towards Data Science: https://towardsdatascience.com/top-10-machine-learning-algorithms-for-data-science-cdb0400a25f9

Krensky, P., Hamer, P. d., Brethenoux, E., Hare, J., Idoine, C., Linden, A., . . . Choudhary, F. (2020). *Gartner 2020 Magic Quadrant for Data Science and Machine Learning Platforms*. Gartner.

Kukhnavets, P. (2019). *Agile vs Waterfall: the Difference Between Methodologies*. Opgehaald van Hygger: https://hygger.io/blog/the-difference-between-agile-and-waterfall/

Lotza, M. (2018). *Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?* Opgehaald van Segue Technologies: https://www.seguetech.com/waterfall-vs-agile-methodology/

M., N. (2020, April 30). *Email Marketing 101: Optimizing For Higher Conversions*. Opgehaald van vwo: https://vwo.com/blog/optimize-email-conversion-rate/

Mailchimp. (2020). *Use Open Tracking in Emails*. Opgehaald van Mailchimp: https://mailchimp.com/help/about-open-tracking/

Mark H. White II, P. (2018, August 20). *HOW BIG SHOULD THE CONTROL GROUP BE IN A RANDOMIZED FIELD EXPERIMENT?* Opgehaald van https://www.markhw.com/blog/control-size

Mialki, S. (2020, January 6). *Email Conversion Rate 101: What You Should Know & 5 Ways to Boost Yours*. Opgehaald van Instapage: https://instapage.com/blog/email-conversion-rate

Morde, V. (2019). *XGBoost Algorithm: Long May She Reign!* Opgehaald van Towards Data Science: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Patel, S. (2017). *10 Essential Characteristics of Highly Successful Entrepreneurs*. Opgehaald van Inc.: https://www.inc.com/sujan-patel/10-essential-characteristics-of-highly-successful-.html

Prakash, A. (2020, January 12). *9 Effective Marketing Practices to Boost Email Conversion Rate*. Opgehaald van Aritic : https://aritic.com/blog/aritic-pinpoint/boost-email-conversion-rate/

Research, G. (2017). *Find out how you stack up to new industry benchmarks for mobile page speed.* Opgehaald van Think With Google: https://www.thinkwithgoogle.com/marketing-resources/data-measurement/mobile-page-speed-new-industry-benchmarks/

Schleckser, J. (2018). *Does Your E-Mail Address Say Hipster, Professional, or Behind the Times?* Opgehaald van Inc.: https://www.inc.com/jim-schleckser/what-your-business-email-address-says-about-you.html

Simon, J. (2019, December 3). *Amazon SageMaker Studio: The First Fully Integrated Development Environment For Machine Learning.* Opgehaald van AWS: https://aws.amazon.com/blogs/aws/amazon-sagemaker-studio-the-first-fully-integrated-development-environment-for-machine-learning/

Smart Insights. (2019, February 21). *How Artificial Intelligence is reshaping email marketing.* Opgehaald van Smart Insights: https://www.smartinsights.com/email-marketing/how-artificial-intelligence-is-reshaping-email-marketing/

Song, L., & Yang, W.-S. (2018). *A Novel Digital Coupon Use Prediction Model Based on XGBoost.* Opgehaald van Atlantis Press.

Survey Monkey. (2020). *Margin of error calculator.* Opgehaald van Survey Monkey: https://www.surveymonkey.com/mp/margin-of-error-calculator/

SurveyMonkey. (2020). *Sample size calculator.* Opgehaald van SurveyMonkey: https://www.surveymonkey.com/mp/sample-size-calculator/?ut_source=help_center

Svetlik, C. (2014, May 21). 7 *WAYS TO INCREASE YOUR EMAIL CONVERSION RATES.* Opgehaald van Newbreedmarketing: https://www.newbreedmarketing.com/blog/increase-your-email-conversion-rates

Totta data lab. (2019). *Gradient Boosting Algoritmes: efficiënt zwakke voorspellers combineren.* Opgehaald van Totta data lab: https://www.tottadatalab.nl/2019/01/07/gradient-boosting-algoritme/

Unchained Carrot. (2019). *How to make an Unchained Carrot intelligent.*

Vries, A. d., Erasmus, P. D., & Gerber, C. (2017). *The familiar versus the unfamiliar: Familiarity bias amongst individual investors.* Opgehaald van ResearchGate: https://www.researchgate.net/publication/313253657_The_familiar_versus_the_unfamiliar_Familiarity_bias_amongst_individual_investors