SOGETI

Voorspellingen van data met Azure.

Sogeti B.V.

Gegevens student
Naam: R.F. Donner
Studentnummer: 2180827
Afstudeerrichting: Voltijd HBO-ICT Software Engineering
Afstudeerperiode: Van 09-11-2015 tot

Gegevens bedrijf
Naam: Sogeti BV.
Afdeling: Microsoft
Plaats: Noord Brabantlaan 265, Eindhoven
Naam begeleider: R. Verhaaf

Gegevens Docentbegeleider
Naam: T. Cats

Gegevens verslag
Titel: Voorspellingen van data met Azure
Datum uitgifte:

Getekend voor gezien door bedrijfsbegeleider:

X

Richard Verhaaf
Bedrijfsbegeleider

Samenvatting

Azure Machine Learning (AML) omvat een serie tools (uit de Microsoft-suite “Azure” omgeving) die “machine learning” ondersteunen. Er was onderzocht welke mogelijkheden het biedt voor Sogeti. Op de Microsoft-afdeling van het bedrijf is nog weinig ervaring met machine learning opgedaan. Er is een Proof of Concept ontwikkeld om de kracht van AML aan te tonen en in een onderzoeksverslag (Bijlage B) staat de functionaliteit van AML beschreven.

Het onderzoek is ontstaan tijdens de ontwikkeling van het VEL project. Daar heeft één van de projectleden een test uitgevoerd met AML en de conclusie getrokken dat dit een goede mogelijkheid is om voorspellingen te maken op data, wat voor dat project van groot belang is.

Bij Machine Learning krijgt een algoritme van een (groot) aantal bestaande gevallen de invoergegevens en de uitkomst aangeboden. Op grond hiervan voorspelt AML de uitkomst voor nieuwe situaties. AML biedt daarvoor vier verschillende algoritmen aan, afhankelijk van wat de gebruiker wil bereiken: Anomaly Detection, Classification, Clustering en Regression

Classification en Regression zijn voor het VEL project van toepassing en hier is dan ook gebruik van gemaakt in de voorspellingsmodellen. De twee types zijn vergeleken, waarna er een besluit is genomen welk algoritme er gebruikt zal worden in het uiteindelijke project.

De implementatie van een Proof of Concept (PoC) is gemaakt in twee vormen, een Console en een WPF applicatie. Dit is stapsgewijs gedaan, in (omdat er maar één ontwikkelaar was) een variant van Scrum, met korte sprints.

Summary

Azure Machine Learning (AML) contains a series of tools (from the Microsoft-suite “Azure”) that support “machine learning”. A research was started to investigate the possibilities that AML poses for Sogeti. The Microsoft-department has little to no experience with machine learning. A Proof of Concept (PoC) was developed to demonstrate the strengths of AML These functionalities were documented in a report (Bijlage B)

The research was started in the development phase of the VEL project. One of its project members made an experiment in AML and found that it was a good possibility to predict data. This is of great importance in the VEL project.

Data is used as input for an algorithm in machine learning, and a possible outcome is predicted based on this input. AML offers four kinds of algorithms depending on what the user needs to accomplish: Anomaly Detection, Classification, Clustering and Regression.

Classification and Regression are applicable in the VEL project and have been used to make prediction models in AML. These two types were compared, after which one of them was chosen to be used in the final project.

The PoC was developed in two forms: a Console application and a WPF application. The WPF application was developed with small increments, in (because there was only one developer) a variation of Scrum.

Voorwoord

Beste lezer,

Bij Sogeti Nederland, een IT-detacheringbedrijf met diverse vestigingen in Nederland, heb ik de afgelopen vijf maanden een afstudeerproject uitgevoerd. Het document dat voor u ligt is de eindscriptie daarvan.

Na een gesprek met mijn bedrijfsbegeleider, Richard Verhaaf, was ik tot de conclusie gekomen dat Azure Machine Learning een interessant afstudeeronderwerp is. Hier ben ik dan ook heel enthousiast mee aan de slag gegaan.

Het doel was om te onderzoeken wat de mogelijkheden zijn van Azure Machine Learning en welke toepassingen ervan voor het 'Voorlopig energie label' (VEL) project binnen Sogeti van belang zijn.

Dit afstudeertraject is voor mij heel leerzaam geweest. Ik heb meer ervaring opgedaan met .Net en 'Design Patterns'. Verder heb ik mijzelf bekend gemaakt met Azure Machine Learning.

Voordat u begint met het lezen van het verslag, wil ik nog enkele mensen bedanken voor de steun tijdens mijn afstudeerproject. Als eerste Richard Verhaaf, voor de technische en morele ondersteuning tijdens het project. Ook wil ik Youri Huig, Hasan Köse, Stefan van der Pas en Xiang Hu bedanken voor de gezellige sfeer die jullie gecreëerd hebben tijdens mijn stage. Ook wil ik Theo Cats bedanken voor het geven van feedback op de nodige documentatie.

En bedankt, u als lezer. Ik wens u veel leesplezier met dit verslag, en hoop u veel nieuwe dingen te leren over Azure Machine Learning.

Robert Donner
Bedrijvencentrum Luminos Eindhoven, 18 februari 2016

Woordenlijst

Woord	Betekenis
BIEB	Onderzoeksmethode waarbij bestaande bronnen, zoals boeken, websites en experts geraadpleegd worden om een conclusie te trekken.
Cloud	De mogelijkheid om via een netwerk (bijv. internet) software en hardware voor de gebruiker beschikbaar te stellen, waardoor de gebruiker lokaal deze software of hardware niet nodig heeft.
Dataset	Een matrix van een gegeven aantal kolommen, die data bevat over een onderwerp.
Design Patterns	Een uniforme regel die gebruikt wordt in het schrijven van code om helder, overzichtelijk en efficiënter te programmeren.
Feature	De waardes uit een enkele kolom van een dataset. Bijvoorbeeld: een auto heeft vier wielen. Aantal wielen is een feature in de dataset.
LAB	Onderzoeksmethode waarbij experimenten uitgevoerd worden om een conclusie te trekken.
Label	De kolom in een dataset die voorspeld wordt in machine learning.
Machine Learning	Een vorm van kunstmatige intelligentie die voorspellingen doet gebaseerd op ingevoerde data.
Mean Absolute Error	Een getal dat aangeeft hoever de waarde afwijkt van de daadwerkelijke waarde. Een waarde van 1.2 zou aangeven dat er op een voorspelde waarde een gemiddelde afwijking van 1.2 is.
Overfitting	Overfitting is in het algemeen wanneer een model te gecompliceerd is, zoals te veel parameters in verhouding tot het aantal waarnemingen . Een model dat lijdt aan overfitting, geeft algemeen slechte voorspellende prestaties.
Predictive Analytics	Het analyseren van data om een voorspelling te maken op nieuwe input.
Trainingset	Een dataset die gebruikt wordt om een algoritme te trainen en te testen met de verwachting dat gelijksoortige data gebruikt worden als input voor het getrainde algoritme, dat dan soortgelijke uitkomsten voorspelt.
NaN	Not a number. Term die gebruikt wordt om aan te geven dat een variabele (onverwacht) geen getal bevat.
User story	Een beschrijving van een functie die een gebruiker terug wilt zien in een applicatie.
Wireframe	Een bouwtekening van een website, waarin de verschillende onderdelen van een applicatie getoond worden.

Inhoudsopgave

1 Inleiding

Sogeti verdiept zich steeds meer in de *Cloud* en heeft als doel om een voorsprong te krijgen in het toepassen van nieuwe Cloud-technologieën. Sogeti vindt het belangrijk om deze voorsprong te behouden op haar concurrenten en wil daarom constant de nieuwe mogelijkheden van de Cloud onderzoeken en kunnen toepassen.

Azure Machine Learning (AML) is één van deze mogelijkheden, waarbij het mogelijk is om *Predictive Analytics* uit te voeren in de Cloud-omgeving. AML is een krachtige Machine Learning tool, waar Sogeti graag meer kennis over wil vergaren.

Sogeti wil weten wat de mogelijkheden zijn van AML en hoe dit toepasbaar is in het bedrijf en voor haar klanten. Deze toepasbaarheid gaat onderzocht worden. Ook wordt er onderzocht wat de algoritmes van AML inhouden en wanneer deze het beste van toepassing zijn op een dataset.

AML biedt veel kansen in het automatiseren van voorspellingen gebaseerd op eerder ingevoerde data. Hoe AML dit doet en wanneer dit van toepassing is wordt ook onderzocht gedurende dit onderzoek.

In dit document zijn woorden die terug te vinden zijn in de woordenlijst onderstreept en cursief (voorbeeld: *Cloud*)

De bijlagen in dit document zijn ingevoegd als PDF bestand. Door te dubbelklikken op het document opent deze in de standaard uitgekozen PDF-lezer. Ze zijn dan ook enkel digitaal beschikbaar en dienen handmatig uitgeprint te worden als dit nodig is.

2 Aanleiding

2.1 Inleiding

Wat is de reden van dit project? Wie is de opdrachtgever en wie heeft het uitgevoerd? Deze vragen worden beantwoord in dit hoofdstuk.

2.2 Sogeti

Sogeti is een dochter/zuster bedrijf van Capgemini. Zij specialiseert zich in verscheidene takken van de IT-wereld, met nadruk op Cloud en Security oplossingen voor klanten in grotere bedrijven zoals ABN-AMRO en Rijkswaterstaat.

Sogeti is, met meer dan 2400 werknemers alleen al in Nederland, één van de grootste detacheringbedrijven ter wereld. Sogeti zet haar grote hoeveelheid specialisten in voor (tijdelijke) werkzaamheden bij bedrijven, waarna zij weer beschikbaar zijn voor een andere klant.

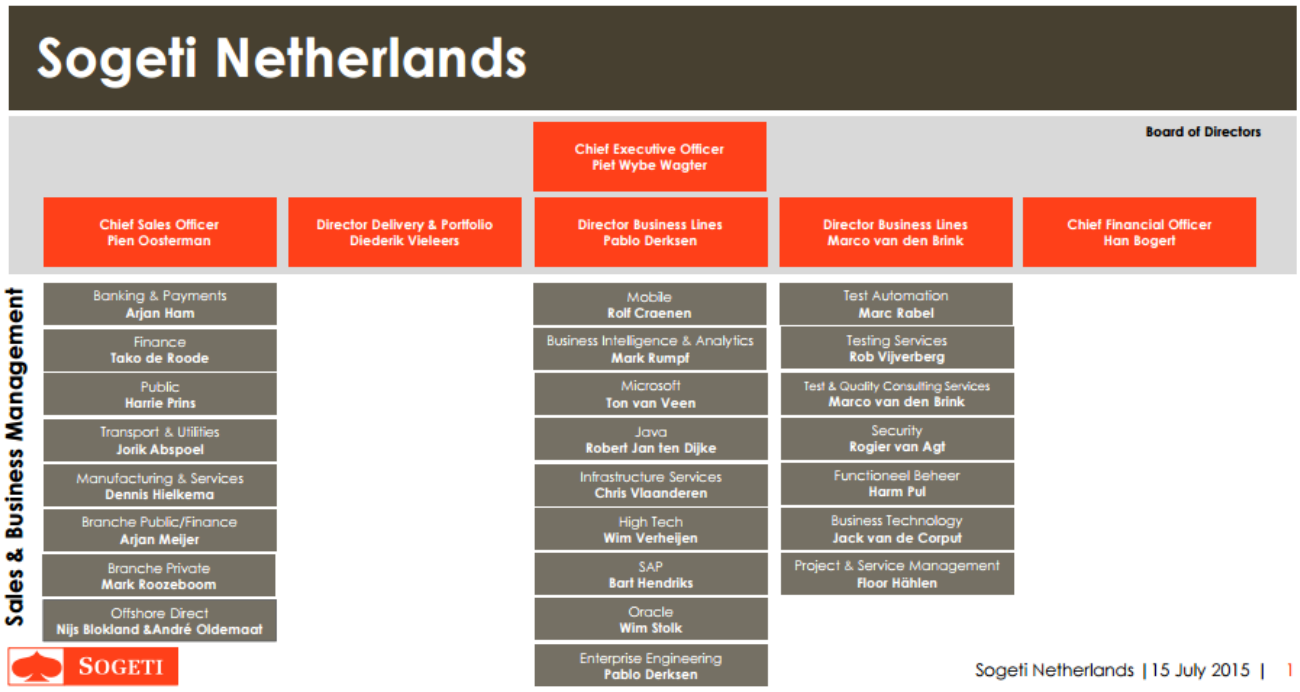
2.2.1 De oprichting en geschiedenis van Sogeti

Sogeti is opgericht op 1 oktober 1967 door Serge Kampf. Hij startte het bedrijf als antwoord op de grote vraag naar ICT-oplossingen. Het bedrijf maakte diverse overnames, fusies, splitsingen en herstructureringen mee totdat het in 1975 geen deel meer uitmaakte van de bedrijfsketen die Serge opgericht had. (Over Sogeti | Sogeti)

In 2002 werd er nieuw leven in Sogeti geblazen, met dezelfde doelstelling als dat het in 1967 had: het leveren van ICT-diensten waar Sogeti invloed op uit kan oefenen, met een sterke lokale binding.

2.2.2 De organisatie van Sogeti

Sogeti heeft een Board of Directors, bestaande uit een CEO en ondersteunende managers. Verder heeft Sogeti 24 Business Lines, wat inhoudt dat voor iedere specialiteit er een groep specialisten klaarstaat. De organisatiestructuur is te vinden in het onderstaande figuur



Figuur 1. Het organogram van Sogeti NL (Over Sogeti | Sogeti)

2.2.3 Het doel van Sogeti

Sogeti ziet zichzelf als een klant- en medewerkervriendelijk bedrijf waar beide partijen een positieve ervaring beleven van het bedrijf. Zo werkt iedere medewerker enthousiast en vol motivatie aan opdrachten die hij/zij zelf heeft uitgekozen, waarmee de klant sneller tevreden zal zijn door de hoge kwaliteit van het product. Door samenwerking met zogenaamde 'best-in-class partners', in combinatie met een veilige omgeving voor medewerkers en de inzet voor een duurzame omgeving, houdt het bedrijf de kwaliteit van de ICT-oplossingen hoog.

2.3 De opdracht

De opdracht is om de mogelijkheden te onderzoeken van AML. Dit wordt gedaan door een overzicht te maken van de toepassingen van de modules die hierin aanwezig zijn en een beschrijving te geven van de algoritmes die AML gebruikt om voorspellingen te doen op ingevoerde data.
Zodra deze mogelijkheden onderzocht zijn wordt een Proof of Concept, vanaf nu PoC genoemd, ontwikkeld als showcase van de mogelijkheden van AML.

Meer informatie over de opdracht kan gevonden worden in Bijlage A: 'PID Hoofdstuk 1.1: Projectdoelstellingen'.

2.3.1 De aanleiding

Sogeti verdiept zich steeds meer in de *Cloud*, en is een 'Azure graduate partner' van Microsoft, met als doel om een voorsprong te nemen in het toepassen van nieuwe Cloud-technologieën. Sogeti vindt het belangrijk om deze voorsprong te houden op haar concurrenten. Sogeti wil daarom constant de nieuwe mogelijkheden van de Cloud, waaronder ook Azure, onderzoeken en kunnen toepassen.

Azure Machine Learning is één van deze mogelijkheden, waarbij het mogelijk is om *Predictive Analytics* uit te voeren in de Cloud-omgeving. AML bevat enorme potentie en Sogeti is er sterk in geïnteresseerd om er meer van te weten te komen.

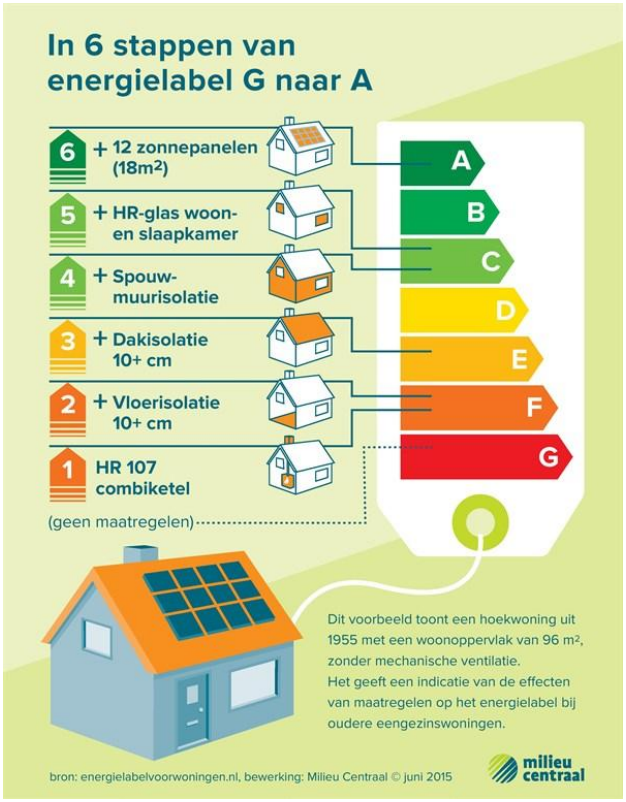
2.3.2 De doelstelling

Het doel van de opdracht is om Sogeti de kennis te verlenen die nodig is om AML toe te passen in de bedrijfsvoering, intern of extern bij een klant. Het PoC zal zelf niet geïntegreerd worden in het bedrijfsproces, maar kan wel als referentiemateriaal gebruikt worden voor toekomstige projecten.

2.3.3 VEL

Het Vereenvoudigd Energie Label, of VEL in het kort, is een opdracht die uitgevoerd wordt door Sogeti. 'Binnenlandse zaken' heeft aan de 'Rijksdienst voor Ondernemend Nederland' de opdracht gegeven om een oplossing te creëren voor het energielabel. Dit is uitbesteed aan de 'Dienst ICT uitvoering' van de overheid die de opdracht aan Sogeti heeft aangeboden.

De opdracht is om een systeem te ontwikkelen dat het energieverbruik van woningen kan berekenen. Afhankelijk van de kenmerken van het huis wordt hier een energielabel aan toegekend. Denk hierbij aan bijvoorbeeld de mate van isolatie of de geïnstalleerde ketel. Nadat de kenmerken van een woning zijn ingevoerd wordt er een energielabel toegekend aan deze woning.



Figuur 2. De 7 verschillende energielabels, en de mogelijkheden waarmee het energielabel verbeterd kan worden. (Centraal, 2015)

Dit wordt gedaan door een zogenaamde reken tool. Deze tool wordt momenteel uitgebreid zodat deze gebruikt kan worden door de eigenaar van de woning zelf. Aangezien deze tool door alle Nederlandse woningeigenaren gebruikt moet gaan worden, is een gestroomlijnde en simpele applicatie nodig.

2.4 De opdrachtnemer

De opdracht wordt uitgevoerd door Robert Donner, een Bachelor-student aan de Fontys Hogescholen te Eindhoven. Hij studeert af aan de opleiding ICT & Software Engineering.

3 De probleemstelling

3.1 Inleiding

Om mee te gaan in de ontwikkelingen van Cloud-technologieën moet Sogeti constant blijven innoveren in het toepassen van nieuwe technologieën. Na de introductie van AML en artikelen over deze toepassingen was de indruk ontstaan dat hier een onderzoek naar gestart moest worden.

3.2 Onderzoeksvraag

De hoofdvraag van dit onderzoek is: **Op welke manier is AML toe te passen in een software project?**

Deze vraag wordt geanalyseerd en de resultaten worden hieronder samengevat. Bijlage B bevat een uitgebreid verslag van het onderzoek.

3.2.1 Deelvragen

De hoofdvraag is in de volgende vier deelvragen geconcretiseerd:

Wat zijn de modules van AML en wat is de functie van deze modules?
Deze vraag wordt gesteld om technische kennis van AML te verkrijgen, zodat de volgende deelvragen met meer kennis van AML onderzocht en beantwoord kunnen worden.

Welke algoritmes gebruikt AML voor Machine Learning?
Hiermee bestuderen we wat de algoritmes voor Machine Learning doen en hoe ze dit doen.

Hoe is AML toepasbaar in applicaties?
Met deze vraag gaat er onderzocht worden hoe AML gekoppeld wordt aan een applicatie, en hoe deze reageert op data die uit de applicatie komen.

Wanneer is AML een verstandige keuze voor een applicatie?
Deze vraag is cruciaal, omdat Sogeti een advies wil voor het gebruik in de eigen applicaties. Hiervoor zal onderzoek gedaan worden wanneer AML *niet* toepasbaar is in een applicatie, bijvoorbeeld doordat de uitkomsten onvoldoende nauwkeurig zijn.

3.3 Doelstelling van het onderzoek

Het onderzoek zal een beschrijvend onderzoek zijn. Dit houdt in dat de conclusie een rapport is van de visie die de opdrachtnemer heeft van AML. Gebaseerd op dit rapport wordt er een advies opgesteld voor Sogeti. Dit rapport zal de onderbouwing zijn voor een ontwikkeling van een PoC.

De uiteindelijke doelstelling is om Sogeti en haar medewerkers een duidelijk beeld te geven van AML, en de toepassingen hiervan, maar natuurlijk ook wanneer het verstandig is om deze module van Azure te gebruiken.

4 Machine Learning, wat is dat?

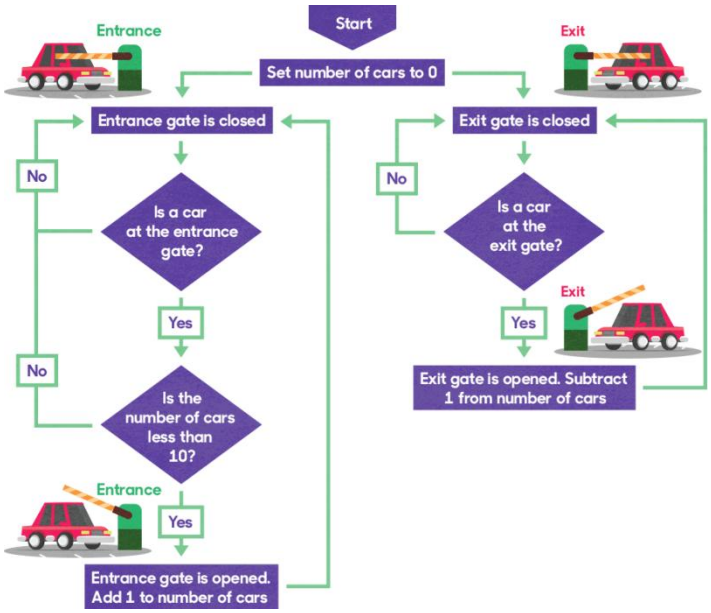
4.1 Inleiding

In dit hoofdstuk wordt u geïntroduceerd in de wereld van *Machine Learning*. Er wordt verteld wat Machine Learning is, en wat het kan betekenen voor een applicatie. Ook wordt de implementatie van deze theorie uitgelegd, voornamelijk AML, en wat de functies van AML zijn.

4.2 Machine Learning

"In general, machine learning involves adaptive mechanisms that enable computers to learn from experience, learn by example and learn by analogy." (Negnevitsky, 2014)

Een computer is gemaakt om het voor de mens makkelijker te maken berekeningen uit te voeren. Een computer maakt hiervoor gebruik van een zogenaamd '*algoritme'* om tot een antwoord van een berekening te komen. Normaliter is het de taak van een programmeur om een algoritme te schrijven, waarbij de computer dat uitvoert en het antwoord teruggeeft.



"An algorithm is a sequence of instructions or a set of rules that are followed to complete a task. This task can be anything, so long as you can give clear instructions for it." (BBC Bitesize)

Hoe zou het zijn als een computer zelf zijn algoritmes kan schrijven en bijwerken, en berekeningen kan maken op complexe stukken data waar de programmeur zelf niet aan kan werken door de complexiteit? Dit is waar kunstmatige intelligentie te pas komt. Een stuk code dat niet alleen zichzelf nieuwe dingen aanleert, maar ook zichzelf verbetert.

Figuur 3. Een simpel voorbeeld van een algoritme. De diamantjes zijn een keuzemoment, met als keuze ja of nee. Als de keuze is gemaakt volg je de pijl met het betreffende antwoord. (BBC Bitesize)

"What [Artificial Intelligence] does is appropriate for its circumstances and its goal, it is flexible to changing environments and changing goals, it learns from experience, and it makes appropriate choices given perceptual limitations and finite computation." (Poole, 1998)

Negnevitsky legt uit dat Machine Learning een adaptief stuk software is dat leert van ervaring. Poole verklaart dat kunstmatige intelligentie gebruik maakt van dezelfde soort technieken. Je kan hieruit afleiden dat Machine Learning een vorm van kunstmatige intelligente is.

Machine Learning maakt gebruik van een algoritme om gegevens die het ontvangt, de zogenaamde input, om te zetten naar een verwacht resultaat, de zogenaamde output. Door het gebruik van trainingsets is de computer in staat om het algoritme specifiek te trainen voor de gegeven data.

De rol van machine learning is meestal om de gebruiker hiervan een advies te geven gebaseerd op input. Dit advies is de voorspelling en kan geaccepteerd of geweigerd worden door de gebruiker.

4.3 Azure Machine Learning

AML is een Machine Learning tool aangeboden door Microsoft, en is zeer krachtig en flexibel. AML wordt voornamelijk gebruikt om voorspellingen te maken, maar kan ook gebruikt worden om data te manipuleren voor andere applicaties. Het maakt gebruik van bestaande data uit een *dataset*, een collectie van informatie, om een algoritme te trainen. Dit algoritme kan gebruikt worden om een voorspelling te maken op een *feature*.

Een feature is een enkele waarde uit een dataset. Dit is bijvoorbeeld de temperatuur in graden Celsius, in een dataset waar alle informatie van het weer opgeslagen wordt.

In dit hoofdstuk wordt beschreven wat de functies zijn die AML biedt en hoe deze functies van toepassing zijn op het trainen van een algoritme. Het beschrijft de zogenaamde modules, en een korte samenvatting van de functie van deze modules. De categorieën van de algoritmes worden besproken aan het eind van dit hoofdstuk.

4.4 Aanpak van het onderzoek

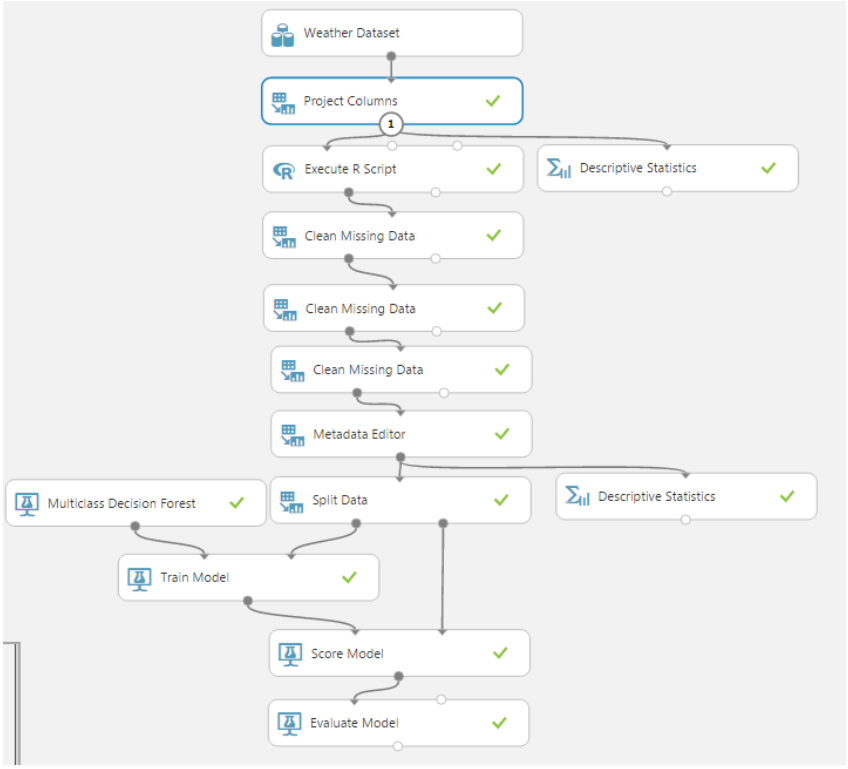
Het onderzoek naar de mogelijkheden van AML is een combinatie van "*LAB*" en "*BIEB*" onderzoek. LAB betekent dat er experimenten uitgevoerd zijn in de praktijk, waarbij de resultaten meegenomen worden in het onderzoek. BIEB betekent dat er onderzoek gedaan wordt door gebruik te maken van bestaande kennis, zoals boeken en rapporten. Als eerste is er zonder voorkennis van de mogelijkheden een experiment uitgevoerd om de nauwkeurigheid van de modules te testen. Hierna zijn de mogelijkheden onderzocht, waarbij specifiek is gekeken naar de functie en gebruik van ieder van deze mogelijkheden en de toepasbaarheid binnen AML. Als laatste is er een nieuw experiment uitgevoerd. Dit experiment is uitgevoerd om aan te tonen wat het verschil is tussen een gebruiker met kennis van AML en een gebruiker zonder kennis van AML.

4.5 Modules van AML

AML maakt gebruik van aparte stukken code om een transformatie uit te voeren op een gegeven set data. Iedere aparte stuk code is een module. Deze modules kunnen gebruikt worden om een dataset te transformeren naar de wens van de gebruiker.

Iedere module valt onder een categorie van transformatie. Een uitgebreide uitleg is terug te vinden in Bijlage B. Een korte samenvatting van iedere categorie:

- **Filter**, Deze modules worden gebruikt om een signaal te filteren. Zo is het bijvoorbeeld mogelijk om kleuren te herkennen in een afbeelding, of een stem te herkennen.
- **Learning with Counts**, voor het tellen en vertalen van features uit een dataset naar een nieuwe dataset.
- **Machine Learning**, de algoritmes van AML en de modules die vereist zijn om deze te trainen en te evalueren.
- **Manipulation**, manipulatie van data, ofwel het verwijderen, toevoegen of aanpassen van data uit een dataset.
- **Sample and Split**, het splitsen van een dataset in twee of meer kleinere datasets. Bijvoorbeeld voor het trainen of testen van een getraind algoritme, vanaf nu model genoemd.
- **Scale and Reduce**, modules gebruikt om een dataset te vergroten of juist te verkleinen door het toepassen van kennis over de dataset.
- **Feature Selection**, om een onderzoek uit te voeren over de dataset en te bepalen welke features het beste gebruikt kunnen worden om een model te trainen.
- **Code modules**, modules waarin 'OpenCV', 'Python' of 'R' gebruikt kunnen worden als programmeertaal.
- **Statistics**, voor het genereren van een rapport gebaseerd op de data uit een dataset.
- **Text Analytics**, accepteert tekst als input, en kan hierop een dataset creëren, of bepaalde patronen herkennen in de input.



Figuur 4. Een experiment waarin een dataset ingeladen wordt en door modules uit diverse categorieën wordt getransformeerd, waarna er voorspeld wordt met een Machine Learning algoritme.

4.6 Algoritmes van AML

Zoals eerder aangegeven maakt AML gebruik van algoritmes om voorspellingen uit te voeren. Deze algoritmes zijn beschreven in Bijlage B. Ieder algoritme valt onder een bepaalde categorie. Deze categorieën zijn:

- **Anomaly Detection:** Hierin kunnen afwijkingen in data herkend worden. Deze algoritmes kunnen gebruikt worden als er in je dataset een binaire feature zit, waarvan je veel instanties hebt met 1 van deze 2 uitkomsten, bijvoorbeeld het detecteren van fraude door een bank.
- **Classification:** classificeren van data. Het is mogelijk om zowel binaire als 'multiclass' (een feature met meer dan 2 waardes) te voorspellen met deze algoritmes. Hiermee kan bijvoorbeeld het type auto voorspeld worden gebaseerd op motor, wielbasis, etc.
- **Clustering**: Clusteren in verschillende groepen. Dit is bijvoorbeeld nuttig als je wilt weten in wat voor gevarenzone een klant zit bij de bank.
- **Regression:** op basis van numerieke data een voorspelling doen op wat de mogelijke (meestal) numerieke uitkomst is voor een feature. Dit kan bijvoorbeeld gebruikt worden om het weer te voorspellen.

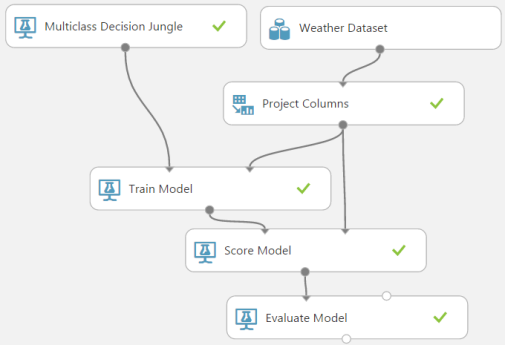
4.7 AML in de praktijk

Er is gebruik gemaakt van AML om voorspellingen te doen op verscheidene zaken. Het gros van de experimenten is gebaseerd op twee datasets: de weerstatistiek op luchthavens, en het Energielabel van de Nederlandse overheid (dataset van het VEL project).

4.7.1 Weerstatistiek

Het weerstatistiek experiment is uitgevoerd zonder voorkennis van AML. Het bleek een lastige opgave doordat er niet duidelijk was wat er met data gebeurde en hoe dit aangepast kan worden. Het eerste experiment was een classificatie algoritme trainen om de temperatuur te voorspellen op een bepaalde

dag. De nauwkeurigheid was lager dan voorheen verwacht. Met een voorspellingskracht van 23.3% was dit gewoon niet accuraat genoeg voor een bevredigend resultaat.



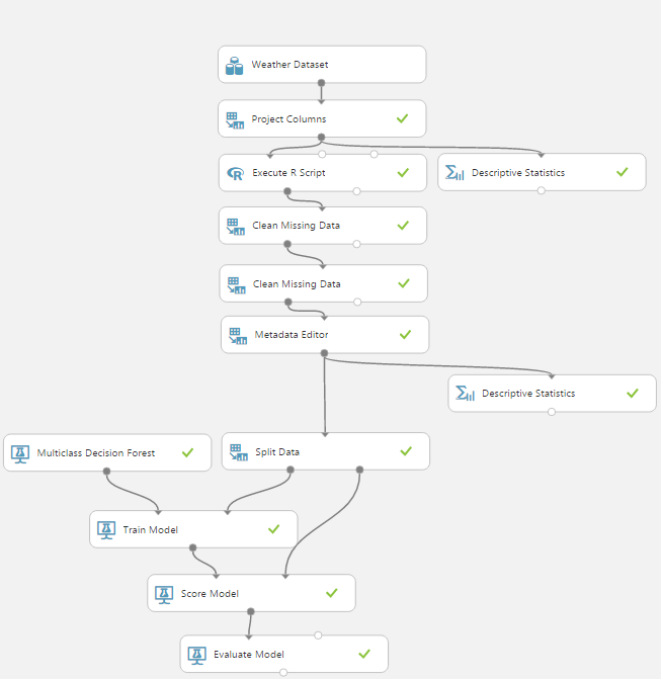
Overall accuracy	0.048475
Average accuracy	0.994251
Micro-averaged precision	0.048475
Macro-averaged precision	NaN
Micro-averaged recall	0.048475
Macro-averaged recall	0.007418

Figuur 5. Het eerste weerexperiment met AML. Dit experiment heeft een nauwkeurigheid van 4.85%.

Hierna is er gekeken of er mogelijkheden waren om deze nauwkeurigheid te verhogen. Één van deze mogelijkheden was om de dataset aan te passen met R. R is een programmeertaal die gebruikt wordt om analyses en statistieken te maken gebaseerd op data.

Vervolgens is er besloten om een cursus in R te volgen. Met een R-script zijn alle tabellen in de dataset naar een numerieke waarde omgezet. Dit was nuttig omdat alle gegevens als strings waren weergegeven, zelfs als ze een numerieke waarde bevatten.

Daarna zijn er kolommen verwijderd om een snellere trainingstijd te verkrijgen en *overfitting* te voorkomen. Overfitting betekent dat de data in overvloed is waardoor de algoritmes niet accuraat kunnen trainen. Het eerste uitgebreide experiment resulteerde in een betrouwbaarheid van 23.3%.

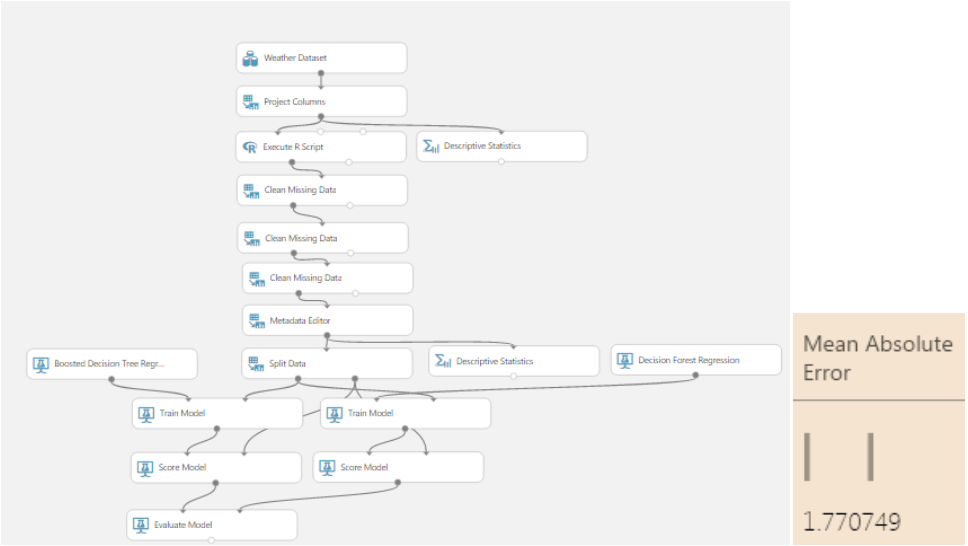


Overall accuracy	0.233257
Average accuracy	0.986185
Micro-averaged precision	0.233257
Macro-averaged precision	NaN
Micro-averaged recall	0.233257
Macro-averaged recall	NaN

Figuur 6. Het eerste uitgebreide weerexperiment met AML. Dit experiment heeft een nauwkeurigheid van 23.3%.

Hierna zijn er nog wat kleine aanpassingen gemaakt aan de instellingen van het experiment. Daardoor is het gelukt om een maximale nauwkeurigheid van 35.5% te verkrijgen: een hogere precisie dan 5%, maar niet voldoende om dit een nauwkeurige voorspelling te noemen.

Er waren een paar verbeteringen in het experiment om het nauwkeuriger te maken dan dat het op dat moment was. Er kon onder meer veel beter regressie gebruikt worden om het weer te voorspellen, doordat deze numerieke voorspellingen doet, en de kans groter is dat deze voorspellingen dichter bij elkaar liggen.



Figuur 7. Het weerexperiment, nadat het is aangepast voor regressie. De Mean Absolute Error van 1.77 geeft aan dat de waarde gemiddeld 1.77 afwijkt van de daadwerkelijke data.

Er is nu een gemiddelde marge van 1.77 graden Celsius per voorspelling. Dit is acceptabel, zeker als er verondersteld wordt dat een machine deze voorspelling maakt zonder enige kennis van meteorologie.

4.7.2 Energielabel

Na het experimenteren met de weerstatistiek is er een relevante dataset voor Sogeti gebruikt, namelijk woninginformatie. Het VEL-project van Sogeti maakt gebruik van data van woningen om energielabels te verstrekken. Dit wordt momenteel gedaan door een applicatie die specifiek hiervoor geïmplementeerd is. Maar wat is de mogelijkheid van AML hierin en is het eventueel mogelijk voor AML om deze taak over te nemen?

Er zijn veel verschillende mogelijkheden om een dataset te testen. De vraag die voor dit experiment gesteld wordt is of er voor snelheid of nauwkeurigheid gekozen moet worden. Verder is er ook de vraag of het algoritme snel getraind moet worden, of snel moet zijn in voorspellen.

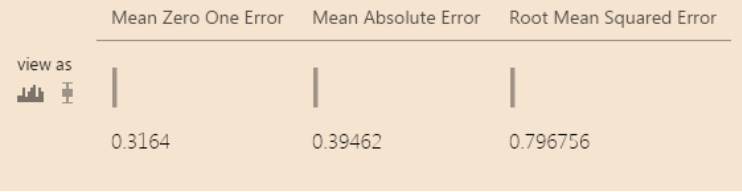
Aangezien het VEL-project accurate voorspellingen vereiste, was al snel duidelijk dat de voorspellingen zo nauwkeurig mogelijk moesten zijn. Snel te trainen experimenten waren wel welkom, maar als dit zijn weerslag zou hebben op de nauwkeurigheid, zou dat geen optie zijn.

Uiteindelijk is de keuze gemaakt om gebruik te maken van regressie, omdat de energielabels in principe vervangen kunnen worden met numerieke waardes. Hiervoor is een R-script geschreven, waarna een regressiealgoritme een voorspelling maakt voor het energielabel. Dit offert echter wel snelheid op voor nauwkeurigheid, doordat regressie langer doet over het trainen en voorspellen.

Na enig onderzoek en experimenteren met de regressiealgoritmes waren er twee die goed toepasbaar bleken voor de voorspelling van de data:

- **Lineaire regressie:** een accurate optie voor het voorspellen van de labels. Nadelen zijn dat de data nog met behulp van R-scripts bewerkt moesten worden om de labels correct weer te geven, en dat het laagste energielabel (G) niet werd weergegeven. Dit komt doordat de training van het algoritme een lineair verband van 1 t/m 6 heeft gemaakt. Hierdoor kan dit algoritme niet energielabel ‘G’ (met waarde 7) voorspellen, ook al is dit gewenst.
- **Ordinale regressie:** Dit algoritme heeft geen modificaties nodig om de voorspelling weer te geven en geeft alle mogelijke labels weer, maar heeft als nadeel dat het minder accuraat is dan lineaire regressie.

Uiteindelijk is ervoor gekozen om ordinale regressie te gebruiken omdat alle labels gerepresenteerd dienen te worden en er geen workarounds nodig zijn om de voorspelling goed te laten werken.



Figuur 8. Het VEL experiment met een getraind model (Energielabel Regression 2.0).

Het getrainde model in Figuur 8 maakt gebruik van ordinale regressie om een voorspelling te doen gebaseerd op de ingevoerde data. De drie variabelen betekenen het volgende voor de nauwkeurigheid:

- Mean Zero One Error (MZOE), de gemiddelde afwijking van het algoritme.
- Mean Absolute error (MAE), het gemiddelde verschil tussen het voorspelde en het werkelijke label.
- Root Mean Squared Error (RMSE) toont de gemiddelde afwijking. Door het kwadrateren wordt er geen verschil gemaakt tussen negatieve en positieve afwijkingen.

De MAE waarde van dit getrainde algoritme is 0.395. Dit betekent dat de voorspelling van een label gemiddeld 0.395 afwijkt van de werkelijke waarde. De MZOE toont aan dat 31.6% van de voorspellingen niet het juiste label teruggeeft. De afwijking van 0.395 wordt over de 31.6% verdeeld doordat dit de enige waardes zijn die afwijken.

De formule om de gemiddelde afwijking van de incorrecte voorspellingen te geven is $X = Y/Z$, waarin X de afwijking is, Y de MAE en Z de MZOE. Voor dit algoritme is de gemiddelde afwijking 1.25. De afwijking toont aan dat een mislukte voorspelling geen grote afwijking zal hebben. De kans is klein dat een afwijking groter dan 1 categorie is.

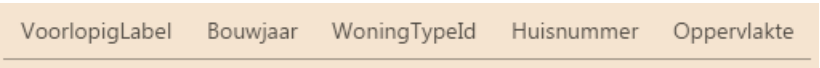
De nauwkeurigheid van 68.4% is niet hoog genoeg voor gebruik in het VEL-project. Er moest een betere optie komen op de huidige dataset. Er waren twee alternatieven om het label beter te voorspellen:

- Het gebruik van een dataset met meer kolommen.
- Het gebruik van een dataset met meer rijen.

Aangezien de eerste dataset waarmee getest werd een kleine variant was van de daadwerkelijke dataset (zowel rijen als kolommen) waren beide opties te testen.

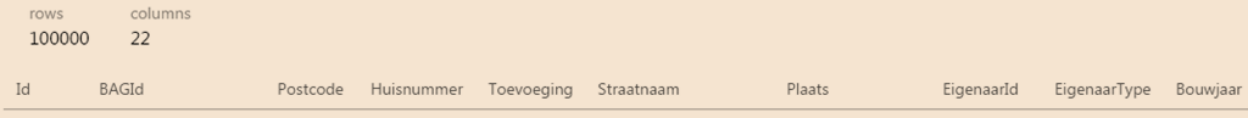
Het gebruik van een dataset met meer kolommen

De eerste dataset die gebruikt werd was een kleinere versie van de daadwerkelijke dataset. Er waren kolommen verwijderd uit de dataset om trainingstijd in te korten, maar er was geen experiment gedaan met de originele dataset.



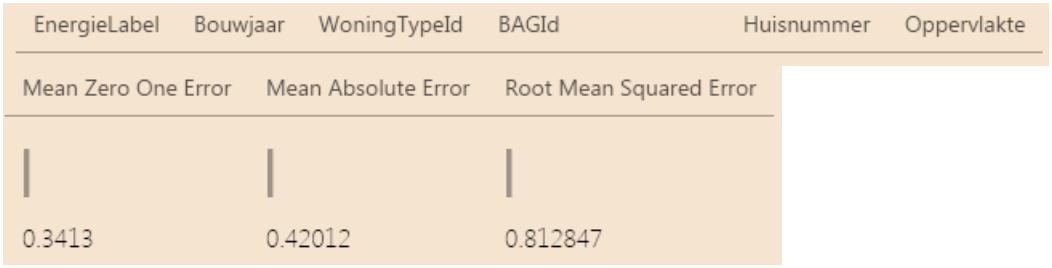
Figuur 9. De kolommen gebruikt uit de originele dataset.

De originele dataset bevat veel meer kolommen dan Figuur 9 laat zien. Deze kolommen waren geselecteerd omdat deze de grootste voorspelkracht hadden en om overfitting door de grote variatie die meenemen van meer kolommen met zich meebrengt, te voorkomen.



Figuur 10. Een klein deel van de originele dataset, die 22 kolommen aan data bevat.

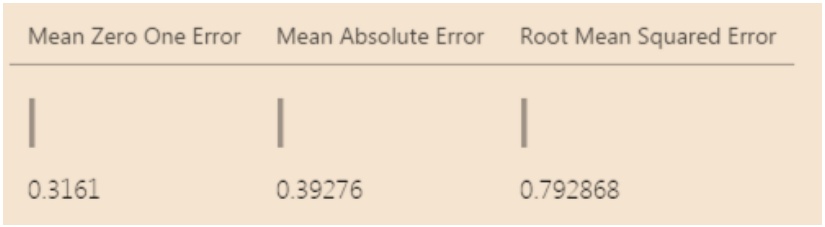
Gebruik maken van de originele dataset zorgde voor een fout in AML, waardoor de dataset eerst aangepast moest worden. Hierna werd iedere kolom geprepareerd voor AML. Dit onderzoek is echter snel gestopt nadat de nauwkeurigheid van het algoritme omlaag ging na het toevoegen van de eerste kolom aan de dataset uit Figuur 9.



Figuur 11. Het resultaat van het experiment bij gebruik van de gehele dataset.

Figuur 11 toont aan dat het toepassen van de volledige dataset de nauwkeurigheid verlaagt als deze vergeleken wordt met het origineel (Figuur 8).

Het gebruik van een dataset met meer rijen



Figuur 12. De evaluatie van het algoritme bij gebruik van vier keer dezelfde trainingsdata.

Voordat er meer data van het VEL-project gebruikt werden, is er eerst getest of het dupliceren van de al bekende data enig effect had op de nauwkeurigheid van het algoritme. Figuur 12 laat zien dat het algoritme accurater is geworden. Dit is mogelijk doordat de trainingdata en de testdata vergelijkbare features hebben. Hierop is AML getraind, en stelt resultaten met grotere zekerheid vast door de grotere hoeveelheid vergelijkbare data. Dit is zeer opmerkelijk doordat het algoritme vierdubbele rijen heeft en zichzelf toch sterker kan trainen. Daarna kwam de vraag of nog meer data de nauwkeurigheid nog groter zouden kunnen maken.

De resultaten van dit onderzoek hebben ervoor gezorgd dat er een onderzoek gestart is naar de verschillende algoritmes en of er een hogere precisie verwacht kan worden bij dit soort voorspellingen. Meer hierover staat beschreven in Bijlage C: Algoritme Training.

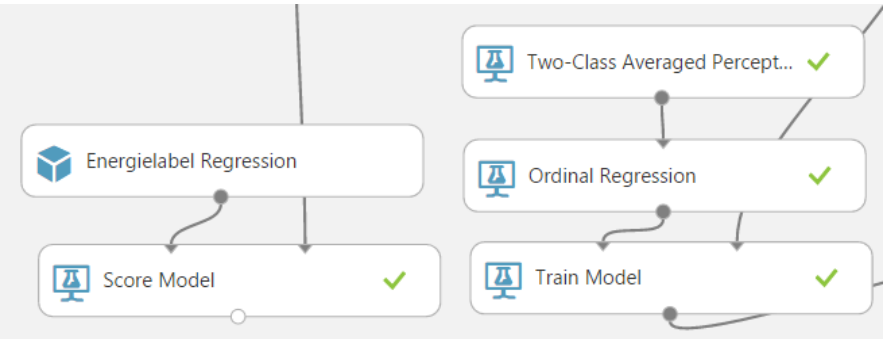
5 Toepassing AML in applicaties

5.1 Inleiding

Na de vergelijkingen van de algoritmes in AML was de volgende stap om de gekozen algoritmes te gebruiken in een applicatie. Hiervoor is het wel nodig om een model te maken en het algoritme voor te bereiden voor de toepassing in een applicatie. Dit hoofdstuk beschrijft de ondernomen stappen voor het voorbereiden van een model voor gebruik in applicaties. Een gedetailleerde uitleg van het voorbereiden van een model is te vinden in Bijlage B: Onderzoeksdocument.

5.2 Opslaan van een getraind algoritme

Een algoritme dat getraind wordt in AML slaat zichzelf niet automatisch op. Het is dus nodig om deze handmatig op te slaan, en het model aan te passen om dit 'getrainde algoritme te gebruiken.



Figuur 13 Het verschil tussen een getraind algoritme (links) en een ongetraind algoritme (rechts)

Een getraind algoritme zorgt uiteraard voor een kortere draaitijd van het model. Natuurlijk zorgt dit er ook voor dat het algoritme geen data meer nodig heeft om te vergelijken, omdat het algoritme al getraind is. Een opgeslagen algoritme maakt gebruik van alle kolommen die meegegeven zijn tijdens het trainen van het algoritme. Als een kolom mist in de input tijdens het voorspellen werkt het algoritme niet. Een kolom extra is niet problematisch. Deze extra kolom wordt genegeerd tijdens de voorspelling.

5.3 Verwijderen van label in experiment

Het *label* is meestal onbekend bij de gebruiker als deze een voorspelling wilt doen op data. Een getraind algoritme heeft dit label ook niet meer nodig in het model, wat betekent dat dit label uit de dataset verwijderd kan worden. Dit is nodig voor een stap die later in dit hoofdstuk uitgelegd zal worden.

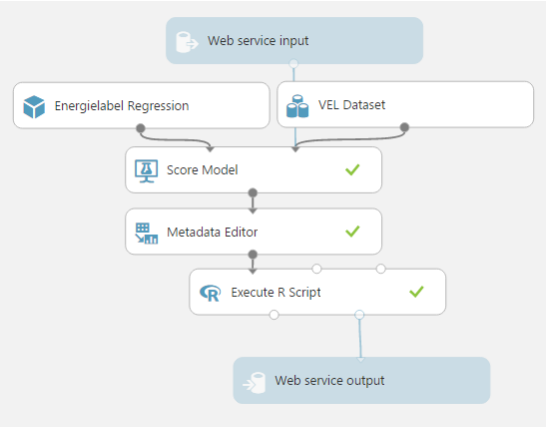
Na het verwijderen van het label is het belangrijk om de dataset op te slaan. Dit is nodig voor een latere stap in dit hoofdstuk.

5.4 Deployen van de webservice

AML biedt de gebruiker aan om een model om te zetten naar een webservice. Op het moment van schrijven is er ook geen andere mogelijkheid om AML toe te passen in applicaties. De webservice maakt gebruik van input en output.

De input is één of meer rijen met data met dezelfde kolommen als de dataset die gebruikt wordt in het model.

De output is de resulterende dataset na alle modificaties van AML.



Figuur 14. Input en output voor de webservice (blauwe balken).

Na het deployen van de webservice is het mogelijk om het getrainde algoritme te gebruiken in applicaties. Hoe de webservice aangeroepen wordt is te vinden in Bijlage B: Onderzoeksdocument.

5.5 Bruikbaarheid

Als de gebruiker AML oproept, worden er data verstuurd over het internet. Dit kan voor sommige applicaties een belemmering zijn. Denk aan applicaties die in een gesloten netwerk draaien of die gebruik maken van persoonlijke data. In het geval van persoonlijke data is het in sommige gevallen niet mogelijk om AML te gebruiken door de regelgeving van privacy en persoonlijke data in software.

Het VEL-project maakt ook gebruik van persoonlijke data, maar voor de toepassing in AML zijn deze data geanonimiseerd. Dit houdt in dat de gebruikte data niet gerelateerd kunnen worden aan een persoon. Hierdoor is de dataset wel bruikbaar, aangezien er geen persoonlijke data verzonden zullen worden over het internet.

6 De implementatie van het PoC

6.1 Inleiding

Het is mogelijk om het algoritme te gebruiken in applicaties nu dat AML de webservice gedeployed is. Dit hoofdstuk beschrijft de implementatie van het PoC en de oplevering hiervan.

6.2 Proces van implementatie

Het plan van de implementatie is om een Proof of Concept te ontwikkelen waarin informatie van een woning ingevuld kan worden, waarna deze verzonden wordt naar AML. Dat voorspelt vervolgens op basis van de ingevoerde data wat het energielabel zou zijn en geeft dat terug aan het PoC.

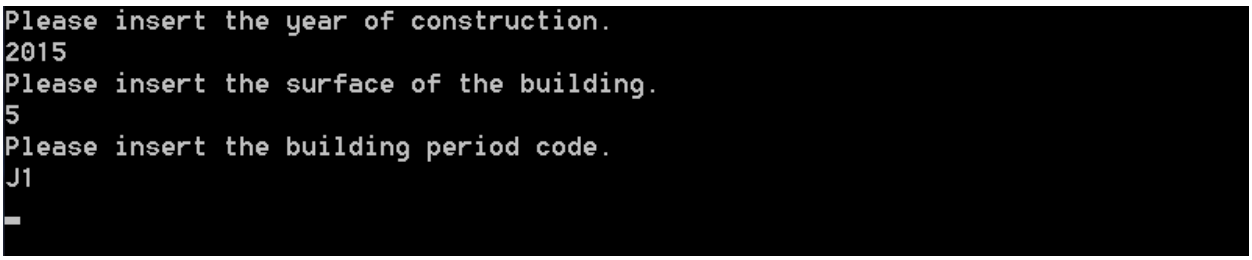
Er worden twee varianten van dit PoC ontwikkeld: een consoleapplicatie om aan te tonen dat het mogelijk is om AML aan te roepen vanuit C# en een demonstratie hoe een eventuele applicatie eruit kan zien die gebruik maakt van AML om een label te voorspellen.

Voor de implementatie van de eerste PoC is geen planning gemaakt. De implementatie is bedoeld als prototype en vereiste geen strakke planning, aangezien dit prototype in een dag of twee ontwikkeld ging worden.

Voor het tweede PoC wordt er een planning gemaakt en zal de implementatie stapsgewijs plaatsvinden.

6.2.1 De console-PoC

De console PoC is ontwikkeld nadat de webservice van AML klaar was om in gebruik genomen te worden. Het minimale dat nodig is om de webservice werkend te krijgen is geïmplementeerd, en gedemonstreerd aan de opdrachtgever.



Figuur 15. De consoleapplicatie waarin het bouwjaar, oppervlakte en een code voor de bouwperiode worden gebruikt voor de voorspelling.

Deze applicatie maakt gebruik van een AML webservice die ontwikkeld is voor het VEL project. Deze webservice maakt gebruik van drie eigenschappen om het energielabel te bepalen, namelijk:

- Bouwjaar
- De oppervlakte van het gebouw
- De periode waarin het gebouwd is.

Deze drie waardes waren op het moment van implementeren de meest bruikbare features voor het voorspellen van het energielabel, maar zorgden voor een lage nauwkeurigheid van het algoritme. Er moest dus een manier verzonnen worden om dit algoritme beter te laten presteren.

Er is opnieuw gekeken naar de kolommen van de VEL-dataset. Door gebruik te maken van 'Filter Based Feature Selection' (Bijlage B : De modules van AML) is er een andere stel kolommen geselecteerd om de voorspelling op te baseren. Deze kolommen zijn:

- Bouwjaar
- Oppervlakte
- Huisnummer

- Gebouwtype

AML kiest deze features uit gebaseerd op zijn verwachting van de voorspelkracht door het verband te leggen tussen iedere feature en de te voorspellen label. Het huisnummer is een zeer vreemde waarde die een hoge voorspellingswaarde heeft (hoger dan anderen), maar is interessant genoeg om te gebruiken. Hierom is deze vreemde waarde toch meegenomen in de laatste versie.

Met dit PoC is er aangetoond dat het mogelijk is om deze service aan te maken en te gebruiken in de .NET omgeving. Om een beter beeld te scheppen van de toepassing in een applicatie, is er een tweede PoC ontwikkeld die gebruik maakt van deze service. Het tweede PoC zal ook gebruik maken van de nieuwe kolommen om het label te voorspellen.

6.2.2 Planning

De tweede PoC is in een paar stappen geïmplementeerd. Als eerste is er gekeken naar de requirements vanuit de opdrachtgever. Deze zijn genoteerd in een document en daarna omgezet naar user-story's. Deze user-story's zijn later gebruikt tijdens de implementatie als maatstaaf.

Werkmethode

Er wordt gewerkt met Agile Scrum, wat inhoudt dat de samenwerking tussen klant en ontwikkelaars nauw ligt, doordat de klant betrokken wordt in het proces.

Voordat er ontwikkeld ging worden is er eerst een lijst met requirements opgesteld. Deze requirements zijn te vinden in Bijlage E: Requirements document.

Agile Scrum
In Agile Scrum wordt er ontwikkeld in zogenaamde sprints, korte implementatieperioden van 2 tot 4 weken waarin telkens een deel van het product wordt opgeleverd.

Iedere dag wordt er een stand-up-meeting gehouden tussen de ontwikkelaars waarin de volgende drie vragen worden gesteld:
- Wat heb ik gedaan?
- Wat ga ik vandaag doen?
- Is er iets wat mijn werk momenteel blokkeert?

Doordat dit project met maar 1 ontwikkelaar wordt ontwikkeld wordt de stand-up overgeslagen. De ontwikkelaar reflecteert wel wekelijks met de opdrachtgever om de voortgang van de sprint te peilen.

Aan het einde van iedere sprint wordt er een korte demo gegeven van de gerealiseerde functionaliteit.

Figuur 16: Een voorbeeld van Agile Scrum. (Rasmusson, 2015)

6.2.3 Tools

De volgende tools zijn gebruikt om de implementatie van het product mogelijk te maken:

- Visual Studio 2015 Community Edition, een software-ontwikkelplatform dat gebruik maakt van .NET.
- Resharper 8.1, een tool die codekwaliteit controleert en zorgt voor consistentie.
- Azure, waar de webservice van AML zich bevindt.
- Visual Studio Online, een teamportaal waarin de code opgeslagen wordt en de vooruitgang van de sprint wordt bijgehouden.
- Microsoft Visio, voor het maken van de klassendiagrammen.

6.2.4 Requirements

De requirements van het PoC zijn flexibel opgesteld zodat er met Scrum gewerkt kan worden. De implementatie was over 3 sprints van 2 weken verdeeld, waarin het project opgeleverd werd. De sprints (zonder *user-story's*) waren als volgt:

1. Request-Response implementatie, deze sprint is gefocust op het verzenden en ontvangen (inclusief label)van de gegevens van een woning.
2. Toepassen *Design Patterns*, deze sprint is gewijd aan het verhelderen van de code en het uniformeren van de projectstructuur zodat men de code beter begrijpt.
3. Frontend ontwikkeling, deze sprint is gewijd aan het ontwikkelen van de frontend van de applicatie.

Request-Response

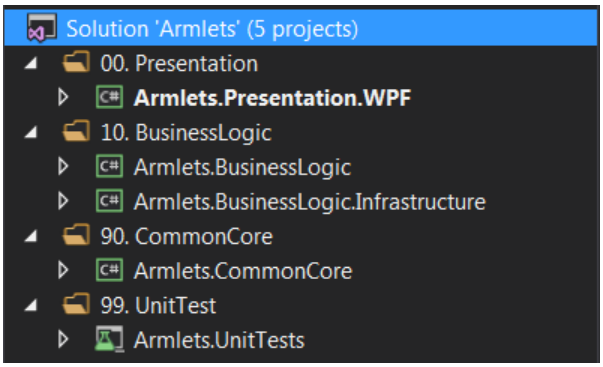
Deze sprint is volledig gewijd aan het implementeren van het verzenden van een verzoek naar de webservice van AML, om vervolgens een respons te ontvangen en te verwerken in de applicatie. Tijdens deze sprint is er niet echt gedacht aan 'code maintainability', ofwel de lees- en onderhoudbaarheid van de code.

Toepassen Design-Patterns

De tweede sprint werd gewijd aan het aanpassen van de code aan de standaarden van de begeleider. De projectstructuur is aangepast, zodat deze uniform is volgens de .NET standaard.

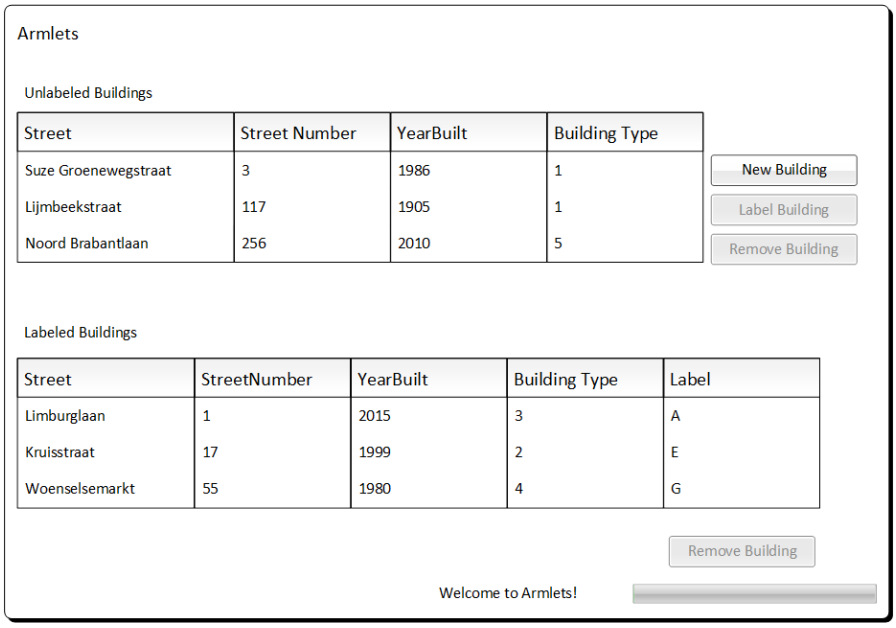
Verder is er gebruik gemaakt van design patterns om de code leesbaarder te maken. Die voldoet nu aan uniformiteitregels. Dit kan vergeleken worden met grammatica uit de taal. Het geheel is nu leesbaarder en het is duidelijker waar wat gevonden kan worden.

De code die gebruikt gaat worden voor de frontend is universeel. Door het gebruik van interfaces is de applicatie onafhankelijk van andere klassen en zijn componenten makkelijk te vervangen. Dit heeft de complexiteit van de applicatie wel verhoogd, maar zorgt er wel voor dat er voldaan wordt aan de codestandaard die de opdrachtgever stelt.



Figuur 17. De folders binnen het project, met de onderliggende projecten.

Frontend ontwikkeling

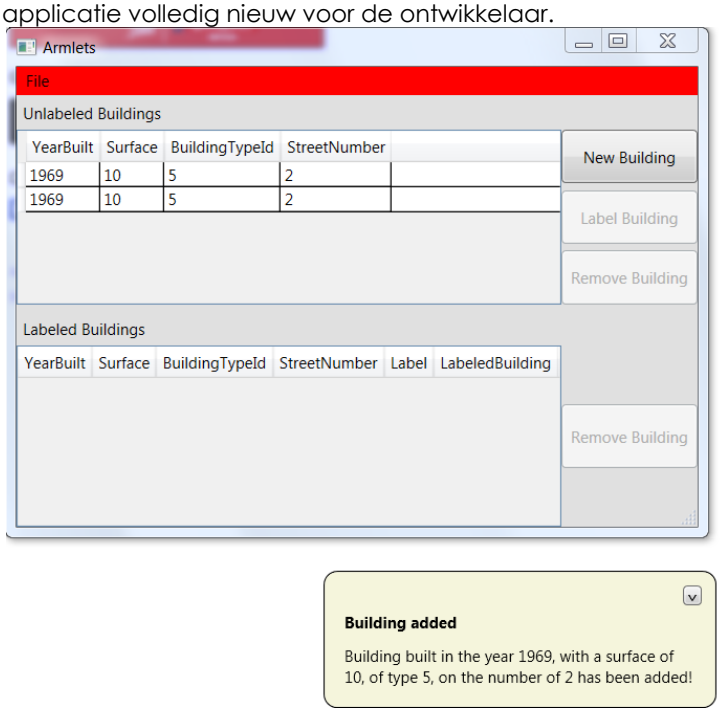


Figuur 18. Een *wireframe* van de te ontwikkelen frontend in WPF ontwikkeld in Visio.

Deze sprint is gefocust op de implementatie van een frontend voor de applicatie. Zoals al eerder beschreven werd is deze frontend ontwikkeld in WPF. Er waren geen eisen gesteld aan de applicatievorm en de opdrachtnemer wilde graag meer kennis opdoen van WPF. Het is echter mogelijk om de applicatie een andere frontend te geven door de losgekoppelde back end die ontwikkeld is voor deze applicatie.

Er is een wireframe ontwikkeld om voor de ontwikkelaar een helder beeld te krijgen van wat er precies geïmplementeerd gaat worden in de frontend. Deze wireframes zijn te vinden in Bijlage D: Wireframes.

De implementatie van de frontend is gestart na het ontwikkelen van de wireframes. Tijdens deze ontwikkeling werden er veel nieuwe dingen geleerd op basis van implementatie. Zo was het stijlen van een applicatie volledig nieuw voor de ontwikkelaar.



Figuur 19 De frontend ontwikkeld in WPF. Notificaties tonen veranderingen in de applicatie.

Figuur 19 laat de applicatie zien die ontworpen was volgens Figuur 18. Het heeft de mogelijkheid om een gebouw aan te maken en deze te labelen door middel van AML.

7 Conclusies en aanbevelingen

Azure Machine Learning is een krachtige tool die gebruikt kan worden voor het manipuleren en voorspellen van data. Er dient rekening gehouden te worden met de nauwkeurigheid van het gebruikte algoritme en de voor- en nadelen die een algoritme met zich mee brengt.

Voor Sogeti is AML een optie om analyses uit te voeren op data en toe te passen in applicaties. De grote hoeveelheid aan opties maakt het divers en aanpasbaar voor verschillende scenario's waarin het gebruikt kan worden.

Het is mogelijk om AML toe te passen in het VEL-project. Het voert goede voorspellingen uit op de gegeven data(tot op 100%). Het wordt aangeraden om een 'Classification'-algoritme te gebruiken omdat deze minimale aanpassingen nodig heeft op de dataset.

De aanbeveling is om AML te gebruiken als kalibratie-middel voor de applicaties die ontwikkeld zijn voor het VEL project. De data die gebruikt wordt om de labels te voorspellen is niet altijd volledig leidend, doordat er ook gebouwen zijn die een uitzondering zijn op de regel. Aangezien het VEL project geen fouten hierin tolereert kan AML niet als definitieve oplossing gebruikt worden.

Er zijn veel vragen die gesteld kunnen worden over AML waarop dit onderzoek geen antwoord geeft. Wat zijn bijvoorbeeld de mogelijkheden van zogenaamde 'filters' en wat betekent dat voor Sogeti? Verder onderzoek naar AML is dan ook een zeer aannemelijke stap, aangezien de potentie duidelijk aanwezig is.

Evaluatie

Het traject dat ik doorlopen heb bij Sogeti was zeer aangenaam. Ik heb veel geleerd van mijn begeleider en heb ook geleerd om zelfstandiger te werken. Ik voel me ook comfortabel met het werk dat ik gedaan heb, en inlever.

Rond augustus had ik mijn eerste sollicitatiegesprek bij Sogeti. Ik vond het destijds erg spannend omdat ik niet wist of het bedrijf bij me zou passen. Toen ik in november begon, was deze twijfel er nog een beetje, maar ik besloot om er toch alles voor te geven en mijn stage tot een goed einde te laten brengen.

Sogeti introduceerde zichzelf met een intro van twee dagen waarin het bedrijf de nieuwe medewerkers welkom heette en liet zien waar Sogeti voor staat en wat het doet. Dit wekte mijn interesse meer om hier mijn afstudeerproject te doen en ik begon dan ook wat enthousiaster na deze introductie.

AML is een zeer interessante tool om mee te werken. Ik heb ook veel tijd gestoken in het leren kennen van de mogelijkheden van AML, wat uiteraard een onderdeel was van de opdracht. Er is zoveel te leren van AML. Ik heb het grootste gedeelte uitgelegd, maar onderdelen waar ik niet mee gewerkt heb zijn niet in detail beschreven.

Het onderzoek was soms heel vermoeiend, omdat ik alle modules wilde bespreken in het verslag. Dit maakte het soms zeer eentonig om eraan te werken doordat veel informatie herhaald werd. Ook dit zorgde ervoor dat de eerste versie van dat document veel informatie miste of haastige conclusies trok.

Ik heb veel geleerd, voornamelijk het gebruiken van AML en .NET met WPF. AML en WPF waren nieuw voor me. Voordat ik begon met de afstudeerperiode, had ik er geen kennis van. Ik ben tevreden over het feit dat ik deze kans gegrepen heb om hierover te leren en ik zou graag nog meer hiervan willen leren.

Het is helaas niet gelukt om een applicatie te ontwikkelen die meerdere algoritmes van AML kan toepassen om een label te voorspellen. Dit is iets wat ik wel in de toekomst zou willen doen.

Ik had voor mijzelf het doel gesteld om meer voor mijzelf op te komen in de opdracht en om vragen te stellen wanneer ik vastloop op een onderdeel. Deze twee factoren waren een belemmering in mijn vorige stageperiode en dit waren dan ook de punten die ik graag wilde verbeteren. Dit is goed gelukt. Ik heb de communicatie tussen mijzelf en mijn begeleider, die een grote rol heeft gespeeld in mijn groei in de afstudeerperiode, goed open gehouden. Ik moet hier echter wel aan blijven werken, omdat dit twee enorme knelpunten zijn in mijn groei.

Als laatste wil ik een nieuw doel voor mijzelf stellen, namelijk om mijzelf en anderen te blijven motiveren met het werk dat ik doe, zodat de resultaten van mijn projecten nog beter worden.

Bibliografie

BBC Bitesize. (sd). Opgeroepen op January 25, 2016, van BBC Bitesize: http://www.bbc.co.uk/guides/zqrq7ty

Centraal, M. (2015, Juni). EnergielabelvanGnaarA. Nederland: Milieu Centraal.

Cultuur en Structuur | Sogeti. (sd). Opgeroepen op Januari 7, 2016, van Sogeti: https://www.sogeti.nl/over-sogeti/cultuur-en-structuur

Facts en Figures | Sogeti. (sd). Opgeroepen op Januari 7, 2016, van Sogeti: https://www.sogeti.nl/over-sogeti/facts-en-figures

Jose, C. (sd). *Microsoft research*. Opgeroepen op January 4, 2016, van Local Deep Kernel Learning for Efficient Non-linear SVM Prediction: http://research.microsoft.com/en-us/um/people/manik/pubs/Jose13.pdf

Negnevitsky, M. (2014). *Artificial Intelligence: A guide to intelligent systems*. Opgeroepen op Januari 25, 2016, van Whatis?: https://books.google.nl/books?id=NP5bBAAAQBAJ

Over Sogeti | Sogeti. (sd). Opgeroepen op Januari 7, 2016, van Sogeti: https://www.sogeti.nl/over-sogeti

Poole, M. &. (1998). *David Poole*. Opgehaald van http://people.cs.ubc.ca: http://people.cs.ubc.ca/~poole/ci/ch1.pdf

Rasmusson, J. (2015). *Scrum*. Opgehaald van agilenutshell.com/scrum: http://www.agilenutshell.com/scrum

Zaykov, Y. (2016, Maart 7). *Mean zero one error?* Opgehaald van social.msdn.microsoft.com: https://social.msdn.microsoft.com/Forums/en-US/078b27d5-3abf-4de5-948d-26afd340f0eb/mean-zero-one-error?forum=MachineLearning

Bijlage A: PID

project: Sogeti azure machine learning

(Sample)

Project initiatie document

Projectcode:	RD_002
Datum voltooid:	4-12-2015
Auteur/groep:	Robert Donner
Versie:	1.0
Status:	Finaal
Document ID:	RD_002.PID_01
Bestandsnaam:	PID_2_0

Documenthistorie

Revisies

Versie	Status	Datum	Wijzigingen
0.1	Concept	13-11-2015	Eerste versie
0.2	Concept	16-11-2015	Product-decompositie-structuur en Project-organisatie-structuur van extra informatie voorzien.
0.3	Concept	24-11-2015	Aanpassen Managementsamenvatting, H1, H2 en H3 met feedback van Theo Cats. H4 Planning toegevoegd.
1.0	Minuut	27-11-2015	Feedback van Richard Verhaaf verwerkt. Het document leesbaarder maken.
2.0	Finaal	04-12-2015	Verbeteren van stijl/spellingsfouten, extra informatie bij H4: Planning

Goedkeuring

Dit document behoeft de volgende goedkeuringen:

Versie	Datum goedkeuring	Naam	Functie
2.0		Richard Verhaaf	Bedrijfsbegeleider
2.0		Theo Cats	Stagebegeleider
2.0		René van der Heijden	Afstudeer coördinator

Distributie

Dit document is verstuurd aan:

Versie	Datum verzending	Naam	Functie
0.1	13-11-2015	Theo Cats	Afstudeerbegeleider
0.2	16-11-2015	Theo Cats	
0.3	24-11-2015	Theo Cats	
	26-11-2015	Richard Verhaaf	Bedrijfsbegeleider
1.0	27-11-2015	Theo Cats	Afstudeerbegeleider
		Richard Verhaaf	Bedrijfsbegeleider
2.0	04-12-2015	Theo Cats	Stagebegeleider
		Richard Verhaaf	Bedrijfsbegeleider
		René van der Heijden	Afstudeer coördinator

Woordenlijst

Dit hoofdstuk bevat alle begrippen en woorden die niet in de context uitgelegd worden. Deze woorden zijn onderstreept-cursief in de tekst en zullen in volgorde hier verschijnen.

Woord	Context
Azure	Een collectie van geïntegreerde Cloud services, aangeboden door Microsoft.
Machine Learning	Een subcategorie van de IT waarbij data gebruikt wordt om de machine nieuwe technieken te leren.
Cloud	De mogelijkheid om via een netwerk (bijv. internet) software en hardware voor de gebruiker beschikbaar te stellen, waardoor de gebruiker lokaal deze software of hardware niet nodig heeft.
Predictive Analytics	Het analyseren van data om een voorspelling te maken op nieuwe input.
Microsoft Developer Network	Een informatiedienst van Microsoft die ondersteuning en ontwikkelkracht biedt aan ontwikkelaars.

Managementsamenvatting

Het doel van dit document is om de kaders te stellen aan de afstudeeropdracht bij Sogeti waarin ik de mogelijkheden van *Azure Machine Learning* (AML) van Microsoft ga onderzoeken. Ik zal mezelf verdiepen in de volgende punten:

1. De toepasbaarheid van AML binnen Sogeti.
2. Het analyseren van de mogelijkheden en functies van Machine Learning in het algemeen.
3. De mogelijkheden van het automatiseren van input met AML.
4. Het realiseren van een Proof of Concept, waarin data ingevoerd wordt, en met de ingevoerde data een voorspelling gemaakt kan worden over nieuwe situaties.

Ik zal mezelf verdiepen in de hiervoor genoemde punten en deze nader beschrijven in dit document.

Aanleiding

Sogeti verdiept zich steeds meer in de *Cloud*, en is een 'Azure graduate partner' van Microsoft, met als doel om een voorsprong te maken in het toepassen van nieuwe Cloud-technologieën. Sogeti vindt het belangrijk om deze voorsprong te houden op zijn concurrenten, en wil daarom constant de nieuwe mogelijkheden van de Cloud, en dus ook Azure, onderzoeken en kunnen toepassen.

Azure Machine Learning is één van deze mogelijkheden, waarbij het mogelijk is om *Predictive-Analytics* uit te voeren in de Cloud-omgeving. AML bevat enorme potentie en Sogeti is hier ernstig in geïnteresseerd om meer van te weten te komen

Globale aanpak

De prioriteit is om de mogelijkheden van AML te onderzoeken, en dus wat de mogelijkheden zijn om dit toe te kunnen passen in nieuwe applicaties van Sogeti.

Daarna zal er onderzoek gedaan worden naar de aangeboden oplossing en hoe deze te ontwikkelen in de *Microsoft Developer Network* (MSDN) omgeving. Als deze oplossing helder is en AML grondig onderzocht is zal er binnen MSDN een Proof of Concept gerealiseerd worden waarin kennis die eerder vergaard is door de ontwikkelaar toegepast zal wordsen.

Doorlooptijd

De doorlooptijd van het project zal 100 werkdagen omslaan, waarbij de start van dit project op 09-11-2015 van start zal gaan.

Globale kosten

Het traject beslaat 100 dagen en het budget bedraagt €11.000-.

Risico's

De risico's voor Sogeti zijn niet significant groot, en zal het bedrijf niet schaden als deze risico's tot stand komen.

Inhoudsopgave

1 Projectdefinitie

1.1 Projectdoelstellingen

Dit project is van start gegaan om een onderzoek te doen naar de mogelijkheden van Azure Machine Learning en met behulp van de resultaten van dat onderzoek een Proof of Concept te ontwikkelen. Dit Proof of Concept zal een gebruiker kunnen simuleren door het analyseren van eerder ingevoerde input door de gebruiker.

Om tot een conclusie te komen stel ik de volgende punten als doelstellingen voor dit project:

1. Onderzoek de mogelijkheden van AML.
    a. De mogelijkheden van AML zijn aan het begin van het traject nog niet bekend bij mij. Dit zal ik als eerste onderzoeken voordat andere stappen ondernomen zullen worden.
2. Maak een overzicht van de mogelijke toepassingen voor eventuele projecten van Sogeti.
    a. Sogeti wil dat AML een onderdeel kan gaan spelen in toekomstige projecten. Als het bedrijf weet wat de mogelijkheden zijn van AML kan het dit bij toekomstige applicaties toepassen.
3. Maak een overzicht van de algoritmes van AML en wat deze betekenen voor Predictive Analytics.
    a. Om een diepere blik te krijgen op de mogelijkheden van Machine Learning worden de gebruikte algoritmes en technieken nader bekeken en onderzocht, zodat er aangegeven kan worden welk algoritme in welke situatie toepasbaar is.
4. Onderzoek of het mogelijk is om data in te voeren, waar AML van kan leren, en kan toepassen op applicaties zonder input van de gebruiker.
    a. AML kan vanuit een database data ophalen en gebruiken om voorspellingen te maken, maar er dient onderzocht te worden of het ook mogelijk is om dit live te doen in een .Net applicatie, en hoe dit gedaan moet worden.
    b. Er zal onderzocht worden in hoeverre de gebruiker de data moet verifiëren, en waar AML zo accuraat is dat verificatie niet nodig zal zijn.
5. Ontwikkel een Proof of Concept waarin Azure ML leert van de gebruiker door input te lezen en vergelijken met eerdere input. Azure ML moet hier dusdanig van leren dat hij met een accuratie van 85% een voorspelling kan maken.
    a. De combinatie van de hierboven gespecificeerde activiteiten zal uiteindelijk een oplossing bieden voor het implementeren van het Proof of Concept.

1.2 Gekozen oplossing of aanpak

Er zal eerst onderzoek gedaan worden naar de specificaties van AML voordat er een Proof of Concept ontwikkeld kan worden.

De conclusies die getrokken worden in het onderzoek zullen leiden naar een aanbieding van een mogelijke implementatie van het Proof of Concept

Als eerste zal er een onderzoek gedaan moeten worden voordat er een oplossing aangeboden kan worden aan Sogeti. Het onderzoek zal stapsgewijs conclusies naar voren brengen die gebruikt kunnen worden om de oplossing te ontwerpen.

Zodra de oplossing volledig ontworpen is wordt er een plan gemaakt om haar te realiseren. Deze oplossing wordt gerealiseerd volgens de Agile methode Scrum, waar iedere week een iteratieslag geleverd wordt. In die iteraties groeit een would-be deployable uit tot het eindproduct.

De tools die in gebruik genomen gaan worden voor het ontwikkelen van het Proof of Concept zijn als volgt:

- Azure Machine Learning van Microsoft
- Microsoft Visual Studio 2015

1.3 Producten c.q. eindresultaat

De volgende producten zullen opgeleverd zijn aan het einde van de doorlooptijd van het project. Al deze producten zullen zowel intern (Sogeti) als extern (Fontys Hogescholen) beoordeeld worden.

- Een project initiatie document.
    - Document waarin de algemene afspraken en doelstellingen van het project aangegeven staan. Ieder die met dit document akkoord gaat accepteert de voorwaarden die in dit document gegeven zijn.
- Een dagenverantwoording

    - Een document waarin aangegeven wordt op welke dagen welke taken verricht zijn, en wat er gebeurd is tijdens die dagen. Dit is om de bedrijfsbegeleider en stagebegeleider op de hoogte te houden van de voortgang van het project.

- Procesverslag
    - Een document waarin mijn bevindingen van het traject terug te vinden zullen zijn. Hiermee kan het proces bekeken worden, en zal er een duidelijk beeld zijn van mijn aanpak van het project.
- Een onderzoeksverslag
    - Dit document zal de conclusies bevatten die gebruikt worden om een oplossing voor dit project te kunnen realiseren.
- Het Proof of Concept
    - De applicatie zal de geïmplementeerde oplossing zijn die geboden wordt met de conclusies uit het onderzoeksverslag.
- Eindpresentatie
    - Presentatie waarin het onderzoeksverslag samengevat wordt en het Proof of Concept gedemonstreerd zal worden.

1.4 Uitsluitingen

De data die geanalyseerd wordt zorgt voor een voorspelling voor toekomstige input. Doordat deze voorspelling nooit 100% accuraat kan zijn, zal er altijd nog een handmatige verificatie nodig zijn om te controleren of de voorspelling van AML de gewenste uitkomst heeft. Er mag vanuit gegaan worden dat AML een accuratie heeft die, afhankelijk van de hoeveelheid data, tussen 50-90% zal variëren.

Er moet rekening mee gehouden worden dat de applicatie die opgeleverd wordt een Proof of Concept is en dus niet bedoeld is om te integreren binnen het systeem van Sogeti. De conclusies die uit het Proof of Concept getrokken kunnen worden, kunnen echter wel gebruikt worden in latere projecten van Sogeti.

1.5 Budget

- Salaris: Er zijn twee medewerkers van Sogeti actief bezig met het project, namelijk Robert Donner(40 uur per week) en Richard Verhaaf (5 uur per week). De totale kosten van het salaris van deze medewerkers is ongeveer €1.000,- per maand.
- Workspace (eenmalig): De ontwikkelaar heeft een laptop nodig en licenties om aan het werk te kunnen. Dit bedrag omslaat ongeveer €2.500.
- Workspace (maandelijks): De ontwikkelaar werkt op locatie en heeft toegang tot een bureau en alle andere faciliteiten (reiskosten declaratie, seminars, etc.)die Sogeti aan zijn werknemers biedt. De kosten hiervan zullen maandelijks ongeveer €400,- bedragen.

Globaal zal er dus €1.400 per maand aan salaris en werkplaats besteed worden en een eenmalige besteding van €2.500 voor het aanschaffen van de benodigde materialen voor het uitvoeren van de opdracht.

Het traject zal 100 werkdagen omslaan (~6 maanden) waarbij de uiteindelijke kosten dus zullen vallen op ongeveer €10.900 voor het gehele traject. Hierbij zijn afschrijvingen en dergelijke zaken niet meegerekend.

1.6 Risicomanagement

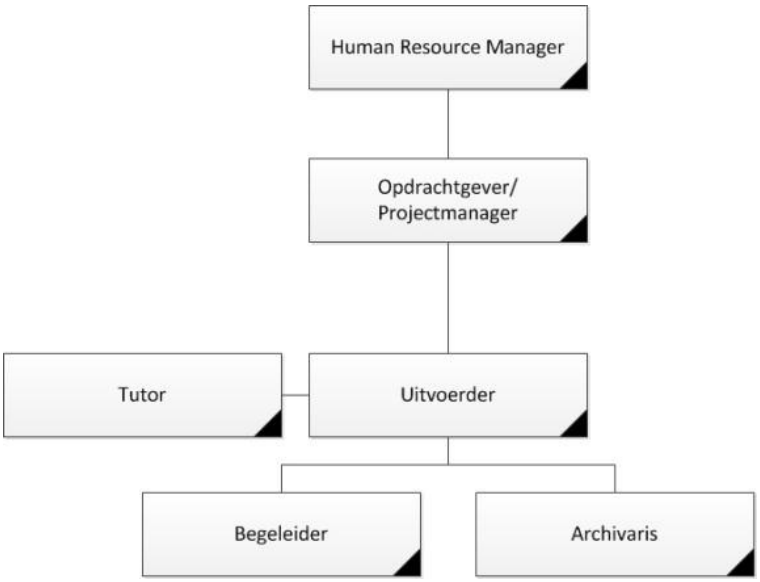
Voor Sogeti is er geen groot risico gebonden aan dit project. De reden hiervoor is dat de lage kosten van de stagiair in combinatie met het feit dat dit een interne opdracht is waar een klant geen last van zal hebben bij een onvolledige uitvoering.

Er is een risico dat niet de volledige potentie van AML onderzocht wordt, wat zou betekenen dat er of onvoldoende informatie is om een conclusie te trekken, of dat er een nieuw traject gestart moet worden om meer informatie te vergaren.

Mocht er een probleem optreden binnen het traject waarbij het project risico loopt, zal Robert Donner samen met Richard Verhaaf zoeken naar een oplossing, en deze oplossing uitvoeren.

Mocht de opdracht voor het afstuderen van Robert Donner voldoende zijn, maar het project niet afgerond zijn, zal dit geen invloed hebben op het afstudeerproces.

2 Projectorganisatiestructuur



2.1 Opdrachtgever/Projectmanager : Richard Verhaaf

Rol: Het begeleiden en leiden van het project.

Verantwoordelijkheid:

- Het succesvol laten verlopen van het project.

Taken

- Begeleiding van de uitvoerder
- Stellen van eisen aan uitvoerder
- Goedkeuren van ingeleverde producten.
- Ondersteuning bij technische en niet technische problemen.
- (Mede)beoordelen van de werkwijze en de resultaten van het project.

2.2 Tutor : Theo Cats

Rol: Het begeleiden van de uitvoerder in de uitvoering en processen omtrent het stage-traject.

Verantwoordelijkheid

Taken:

- Geven van feedback op producten.
- Ondersteuning bij niet technische problemen.
- (Mede)beoordelen van de werkwijze en de resultaten van het project.

2.3 Human Resource Manager: Farida Hoelas

Rol: Het begeleiden van de uitvoerder in de processen rondom het project.

Verantwoordelijkheid:

- Ondersteuning bieden bij problemen die buiten de project scope vallen.

Taken:

- Ondersteuning bij niet technische problemen.

2.4 Uitvoerder : Robert Donner

Rol: Het uitvoeren van het onderzoek en implementeren van de opdracht.

Verantwoordelijkheid:

- Tijdig verzenden van documentatie naar Tutor en Opdrachtgever.
- Implementeren en onderhouden van de producten gedurende de doorlooptijd, als in aangegeven in hoofdstuk 3.

Taken:

- Uitvoeren van het onderzoek.
- Implementeren van het Proof of Concept.

2.5 Archivaris : Robert Donner

Rol: Het maken van documentatie en verslaan van voortgang project.

Verantwoordelijkheid:

- Het correct en duidelijk documenteren van de benodigde producten.

Taken:

- Het schrijven van documentatie.

2.6 Begeleider : Richard Verhaaf

Rol: Het verwijzen naar kennisbronnen en contactpersonen indien begeleiding benodigd is.

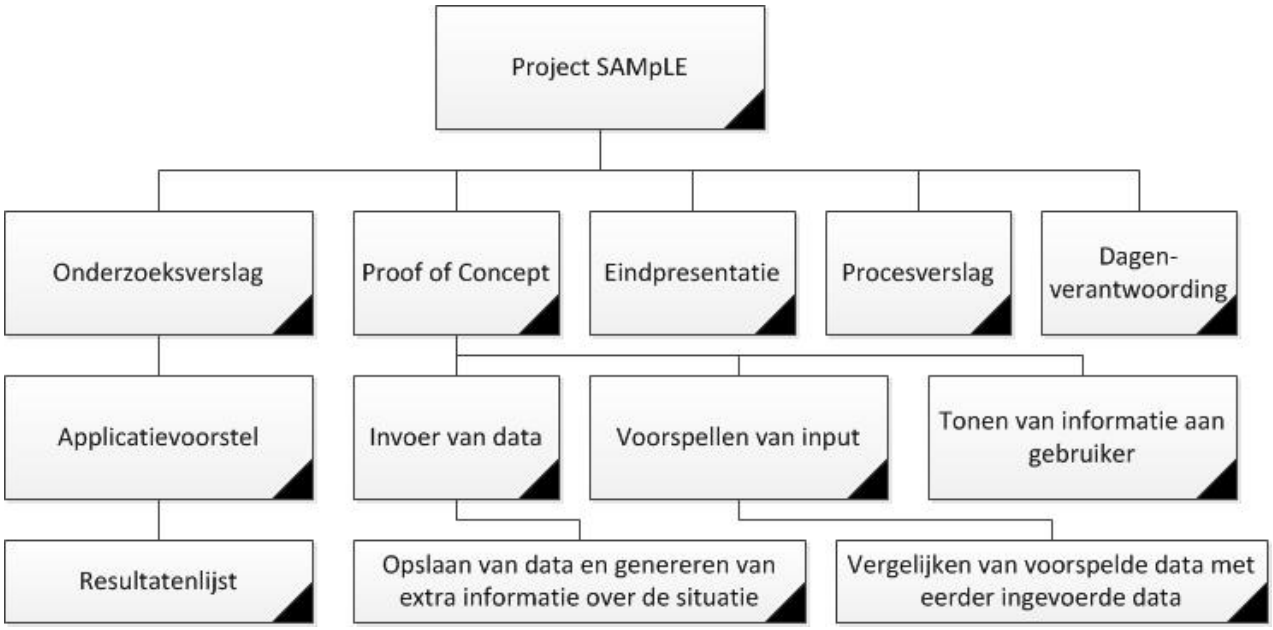
Verantwoordelijkheid:

- Het begeleiden van de uitvoerder in het succesvol volbrengen van het project.

Taken:

- Geven van feedback op documentatie en implementatie.
- Informatie proberen te bieden op vragen van de uitvoerder of archivaris.

3 Product-decompositie-structuur



4 Planning

De planning is geschreven in een traject van 85 werkdagen. Hierbij zijn er dus 15 dagen uitloop mogelijk.

Voor onderdeel 15 (Ontwikkelen applicatie) zal een detailplanning gemaakt worden na de uitvoering van de voorgaande onderdelen.

ID	Task Name	Duration	Start	Finish	Predecessors
1	Opstellen PID	0 days	11/9/15	11/9/15	
2	Management samenvatting	5 days	11/9/15	11/13/15	1
3	Project-organisatie-structuur	5 days	11/9/15	11/13/15	1
4	Project definitie	10 days	11/16/15	11/27/15	2
5	Product-decompositie-structuur	10 days	11/16/15	11/27/15	3
6	Verwerken Feedback PID	4 days	11/30/15	12/3/15	1,2,3,4,5
7	Start project SAMpLE	0 days	12/4/15	12/4/15	
8	Opstart Onderzoek	0 days	12/4/15	12/4/15	
9	Uitzoeken mogelijkheden AML	5 days	12/7/15	12/11/15	8
10	Mogelijke toepassingen AML uitwerken	5 days	12/7/15	12/11/15	8
11	Onderzoek naar de verschillende AML Algoritmes	10 days	12/14/15	12/25/15	10
12	Invoertoepassingen onderzoeken	5 days	12/28/15	1/1/16	11
13	Onderzoeksverslag uitschrijven	2 days	1/4/16	1/5/16	
14	Opleveren onderzoeksverslag	1 day	1/6/16	1/6/16	13
15	Ontwikkelen applicatie	30 days	1/4/16	2/12/16	
16	Testen	11 days	2/15/16	2/29/16	15
17	Bugfixing	11 days	2/15/16	2/29/16	15
18	Voorbereiden eindpresentatie	2 days	2/15/16	2/16/16	15

Bijlage B: Onderzoeksverslag

Het trainen van Machine Learning algoritmes met Azure

Robert Donner
Sogeti Nederland B.V.

Inhoudsopgave

1 Woordenlijst

Woord	Context
Azure	Een Cloud-oplossing aangeboden door Microsoft.
Binomiale verdeling	De verdeling van het aantal successen in een reeks van twee alternatieven met een vaste succeskans.
BLOB (Binary Large Object)	Een verzameling van binaire data opgeslagen als een enkele entiteit. Wordt vaak gebruikt voor multimedia objecten, zoals geluid, of plaatjes.
Categorical	Een eigenschap van een feature die aangeeft dat de waardes in de dataset de enige mogelijke waardes zijn voor deze feature.
Cloud	De mogelijkheid om via een netwerk (bijv. internet) software en hardware voor de gebruiker beschikbaar te stellen, waardoor de gebruiker lokaal deze software of hardware niet nodig heeft.
Dataset	Een matrix van een gegeven aantal kolommen, die data bevat over een bepaald onderwerp.
Eigenwaarde	De waarde waarmee een eigenvector kan strekken, afhankelijk van de matrix.
Eigenvector	Een eigenvector is een vector die niet van richting veranderd tijdens een lineaire transformatie van data, die door een matrix kan worden voorgesteld.
Feature	Een groep waardes uit een dataset, bijvoorbeeld de temperatuur in graden Celsius uit een meteorologische dataset.
Label	De feature waar voorspellingen op gedaan zullen worden.
Loss Function	Een getal dat aangeeft hoeveel de voorspelde waarden afwijken van de daadwerkelijke waarden.
Log Likelihood	Een getal dat de aannemelijkheid van een bepaalde waarde in de voorspelling representeert.
Machine Learning	Een subcategorie van de IT waarbij data gebruikt wordt om de machine nieuwe technieken “Aan te leren”.

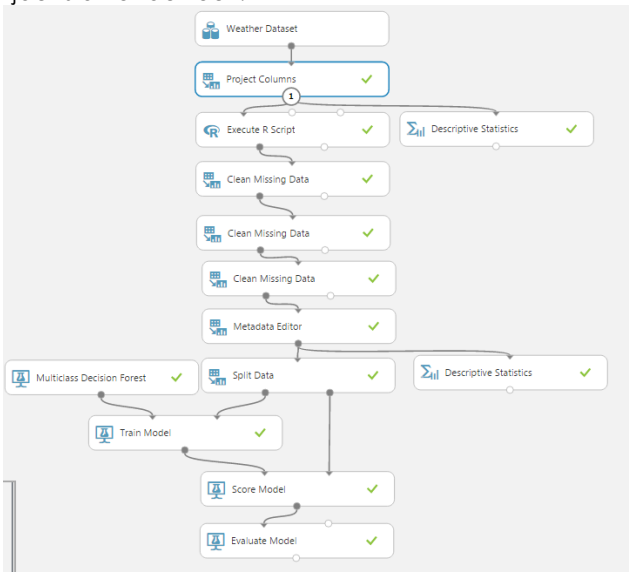
Magnitude spectrum	Het spectrum waar het maximum verschil tot het nulpunt is aangegeven van een signaal.
Non-categorical	Een eigenschap van een feature die aangeeft dat de waardes in de dataset niet de enige mogelijke waardes zijn voor deze feature.
Overfitting	Het gebruiken van teveel data in het trainen van een algoritme, waardoor de training langer duurt of het getrainde model onnauwkeuriger is dan verwacht.
Pearson Correlation	De lineaire correlatie tussen twee gegeven variabelen.
Predictive Analytics	Het analyseren van data om een voorspelling te maken op nieuwe input.
Request-Response Service	Een communicatiewijze in programmatuur waarin de cliënt een verzoek maakt met parameters, en de server een antwoord terug geeft.
Tweezijdig gemiddelde	Een gemiddelde dat zowel de data gebruikt van de training, als enige input van na de training van het algoritme.
Voortschrijdend gemiddelde	Een gemiddelde van een bepaald aantal opeenvolgende elementen.
Weak Learner	Een Machine Learning algoritme dat als enige definitie heeft dat het accurater moet zijn dan willekeurig keuzes maken.

2 Inleiding

Sogeti verdiept zich steeds meer in de *Cloud*, en is een 'Azure graduate partner' van Microsoft, met als doel om een voorsprong te maken in het toepassen van nieuwe Cloud-technologieën. Sogeti vindt het belangrijk om deze voorsprong te houden op zijn concurrenten en wil daarom constant de nieuwe mogelijkheden van de Cloud (en dus ook Azure) onderzoeken en kunnen toepassen.

Azure Machine Learning (AML) is één van deze mogelijkheden, waarbij het mogelijk is om *Predictive-Analytics* uit te voeren in de Cloud-omgeving. AML bevat enorme potentie en Sogeti is er sterk in geïnteresseerd om er meer van te weten te komen.

AML biedt veel kansen in het analyseren van input door voorspellingen te doen die zijn gebaseerd op eerder ingevoerde data. Hoe AML dit doet en wanneer dit van toepassing is zal ook onderzocht worden tijdens dit onderzoek.



Figuur 20: Een voorbeeld van een experiment rond weersvoorspelling. De modules worden in dit verslag beschreven.

De onderzoeksvraag die behandeld wordt in dit onderzoek, is gebaseerd op de context die hierboven is gegeven: Wat zijn de mogelijkheden van AML, en hoe is dit toe te passen in de applicaties van Sogeti? Het VEL project wordt als scope gebruikt voor het onderzoek. Waar mogelijk zullen voorbeelden gedemonstreerd worden aan dit project en zullen hier conclusies uit worden getrokken gebaseerd op dit project.

Hierbij komen de volgende deelvragen te pas:

Wat zijn de modules van AML, en wat is de functie van deze modules? Deze vraag wordt gesteld om technische kennis van AML te ontwikkelen, zodat de volgende deelvragen met meer kennis van AML onderzocht en beantwoord kunnen worden.

Welke algoritmes gebruikt AML voor Machine Learning? Tijdens deze deelvraag wordt er antwoord gegeven op de vraag wat de algoritmes voor Machine Learning doen, en hoe ze dit doen.

Hoe is AML toepasbaar in applicaties? Met deze vraag gaat er onderzocht worden hoe AML gekoppeld wordt aan een applicatie, en hoe deze reageert op data die uit de applicatie komt.

Wanneer is AML een verstandige keuze voor een applicatie? Deze vraag is cruciaal, omdat Sogeti een advies wil voor het gebruik in de eigen applicaties. Hiervoor zal een onderzoek gedaan worden wanneer AML niet toepasbaar is op een applicatie, door bijvoorbeeld een gebrek aan betrouwbaarheid.

3 De modules van AML

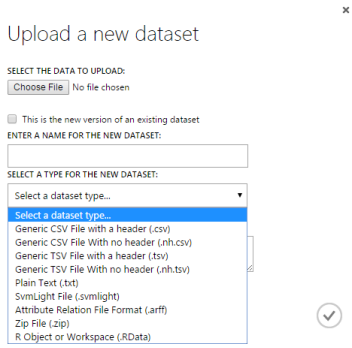
AML is een zeer krachtige en flexibele tool die gebruikt kan worden om voorspellingen te maken. Het maakt gebruik van bestaande data uit een *dataset* om een algoritme te trainen, dat gebruikt kan worden om een voorspelling te doen op een *feature*. De stappen die de gebruiker kan ondernemen staan hieronder aangegeven.

Al de modules die hier beschreven staan worden gebruikt ter voorbereiding op het trainen van één of meer algoritmes van AML. Deze algoritmes en de functies hiervan staan beschreven in het volgende hoofdstuk.

3.1 Importeren en converteren van een dataset

AML maakt gebruik van datasets om voorspellingen te kunnen maken. Zonder data is er immers geen referentiemateriaal voor die voorspellingen. AML accepteert negen verschillende typen datafiles als input, en wel de volgende:

- Generieke CSV met header (.CSV)
- Generieke CSV zonder header (.nh. CSV)
- Generieke TSV met header (.TSV)
- Generieke TSV zonder header (.nh. TSV)
- Tekst bestand (.TXT)
- SVM Light bestand (.svmlight)
- Attribute Relation File Format (.ARFF)
- Zip-bestand (.ZIP)
- R object of Workspace (.RData)



Figuur 21: Het uploaden van een dataset.

AML kan deze 'formats' ook naar elkaar omzetten, bijvoorbeeld.TSV naar .CSV. Dit is bijvoorbeeld handig als een applicatie gebruik maakt van .TSV, maar een andere applicatie de informatie in .CSV wilt ontvangen.

Het voordeel van .CSV en .TSV is dat de data een header bevat. Dit maakt de data leesbaarder voor de gebruiker. Een header houdt in dat de datakolommen een naam meekrijgen.

Col1	Col2	Col3
Robert	Eindhoven	06-12345678
Pieter	Maastricht	06-87654321

Tabel 1 Een voorbeeld van een dataset zonder header.

Naam	Woonplaats	Telefoonnummer
Robert	Eindhoven	06-12345678
Pieter	Maastricht	06-87654321

Tabel 2 Een voorbeeld van een dataset met header.

AML vertaalt zelf zijn informatie naar een standaard formaat (DataSet) en converteert eventuele output indien gewenst ook terug naar het oorspronkelijke formaat.

Het is ook mogelijk om zelf data in te typen voor AML, voor het creëren en bewerken van kleine datasets. Dit is een input voor AML net als al het andere, behalve dat deze in AML zelf gecreëerd wordt.

Als laatste is het ook mogelijk om data te lezen en te schrijven naar een web-url. De Reader en de Writer accepteren verscheidene opties, namelijk de volgende:

- Web URL via http, als de URL direct naar één van de bovengenoemde input formats verwijst, met uitzondering van .Zip en .RData. (enkel voor de Reader)
- Hive Query, als de data van een *Hadoop* opslag komt.
- Azure SQL Database, als de link naar een SQL database van Azure verwijst.
- Azure Table, om tabel data op te halen van Azure.
- Azure Blob Storage, voor het ophalen van data die als BLOB in Azure opgeslagen zijn.
- Data Feed Provider, voor het ophalen van data van een ondersteunde 'Feed provider'. (enkel voor de Reader)

3.2 Aanpassen van data

AML heeft de mogelijkheid om ingevoerde data aan te passen aan de wens van de gebruiker. Zo kan het bijvoorbeeld een kolom data verwijderen omdat deze niet praktisch is voor een berekening, of het automatisch aanvullen van lege rijen data. Dit is meestal de eerste of tweede stap nadat de data is ingevoerd in AML.

3.2.1 Filters

Filters kunnen gebruikt worden om geluid en afbeeldingen te filteren. (Smith, 2007) Dit kan gebruikt worden om ruis te verminderen of een patroon te herkennen in een gegeven signaal van een geluid of afbeelding. De volgende filters zijn beschikbaar om toe te passen:

- FIR Filter ('Finite Impulse Response Filter'), kan gebruikt worden om het *magnitude spectrum* van een signaal te veranderen terwijl de golfvorm behouden blijft.
- IIR Filter ('Infinite Impulse Response Filter'), voor non-lineaire filtering van een signaal.
- Median Filter, voor het filteren van een gemiddeld verwacht signaal, zodat een signaal beter herkend kan worden.
- Moving Average Filter, voor het berekenen van een serie van *een- of tweezijdige gemiddelden* over een dataset. Hieruit ontstaat een *voortschrijdend gemiddelde* dat gebruikt kan worden voor een filtering.
- Threshold Filter, voor het beperken van numerieke waardes tot een gebruikerbepaald bereik. Dit kan zowel een ondergrens zijn als een bovengrens.
- User-Defined Filter, voor het toepassen van een eigen bewerking op een signaal.

3.2.2 Learning with Counts

Counts worden gebruikt om een compactere set van features te maken, gebaseerd op het tellen van waardes van een bepaalde set features. Dit wordt gebruikt als een feature veel verschillende unieke waardes heeft. Counts zoekt naar bepaalde patronen en vertaalt deze naar een andere dataset, die dan gemengd kan worden met een al bestaande dataset. (Microsoft M. L., 2015)

3.2.3 Manipulation

Het grootste gedeelte van het aanpassen van de data zal onder het kopje Manipulation vallen. Toevoegen, aanpassen en verwijderen van data vallen allemaal onder deze categorie. De volgende mogelijkheden zijn er voor manipulatie van de dataset:

Add Columns
Voor het toevoegen van één of meerdere kolommen aan een dataset. Dit wordt gedaan door een zogenaamde 'merge' toe te passen op twee datasets. De datasets moeten een sleutelkolom hebben waarmee ze aan elkaar verbonden zijn.

Add rows
Deze module wordt gebruikt wanneer de gebruiker een nieuwe rij data, bijvoorbeeld een nieuwe gebruiker, wilt toevoegen aan de dataset. Deze rows moeten uit een gelijke dataset komen als de originele dataset, anders is het toevoegen van een row niet mogelijk.

Apply SQL Transformation
Maakt gebruik van SQLite om een SQL transformatie uit te voeren op een dataset.

Clean Missing Data
Missende informatie kan voor sommige algoritmes funest zijn voor de training. Om deze informatie aan te vullen, of te verwijderen, kan deze module gebruikt worden. Dit kan gedaan worden door bijvoorbeeld het gemiddelde cijfer in te vullen, of door de hele rij of zelfs hele kolom te verwijderen.

Convert to Indicator Values
Deze module wordt gebruikt om een niet-binaire kolom om te zetten naar meerdere binaire kolommen. Dit wordt gedaan door voor iedere waarde uit de kolom een nieuwe kolom te maken, en dan met de waardes 0 en 1 aan te geven of deze waarde van toepassing is in die rij data. Er dient wel rekening mee gehouden te worden dat er geen constraint op deze tabel ontstaat. Dit betekent dat het mogelijk is om data die niet voldoet aan de verwachtingen van de dataset toe te voegen, door bijvoorbeeld het gebruik van de 'Add rows' module.

Auto	Kleur
Aston Martin	Zilver
Ferrari	Rood
BMW	Zwart

Tabel 3 Een voorbeeld van een multiclass tabel, waarin een auto drie kleuren kan hebben.

Auto	Is_Zilver	Is_Rood	Is_Zwart
Aston Martin	1	0	0
Ferrari	0	1	0
BMW	0	0	1

Tabel 4 Een voorbeeld waarin de Convert to Indicator Values module is toegepast op Tabel 3.

Auto	Is_Zilver	Is_Rood	Is_Zwart
Aston Martin	1	0	0
Ferrari	0	1	0
BMW	0	0	1
Nieuw	1	1	2

Tabel 5 Een voorbeeld waarin de Convert to Indicator Values module is toegepast op Tabel 3, waarna een nieuwe auto is toegevoegd met verkeerde data. Er zijn geen constraints gelegd op de dataset, dus dit is mogelijk.

Group Categorical Values
Vermindert het aantal categorieën in een kolom. Dit is bijvoorbeeld handig als een postcode te accuraat is voor Machine Learning en dus minder specifiek gebruikt kan worden. (5612 SE, 5612 DF en 5612 ZK zijn allemaal postodes in Eindhoven).

Join Data
Voegt twee sets van data samen in een enkele set data. Dit kan op iedere manier zoals een standaard SQL database dat zou kunnen.

Metadata Editor
Deze module wordt gebruikt om de metadata van één of meer kolommen aan te passen. Zo kun je bijvoorbeeld de soort inhoud (string, numeric) aanpassen, mits deze conversie mogelijk is.

Project Columns
maakt een nieuwe dataset gebaseerd op welke kolommen geselecteerd zijn in deze module.

Remove Duplicate Rows
Deze module verwijdert dubbele rijen uit een dataset.

Auto	Kleur
Aston Martin	Zilver
~~Aston Martin~~	~~Zilver~~
Ferrari	Rood
BMW	Zwart

Tabel 6 Een tabel waarin de module Remove Duplicate Row wordt gebruikt. De dubbele waarde van Aston Martin wordt verwijderd.

Select Columns Transform
Geeft, net als Project Columns, de mogelijkheid om een select aantal kolommen in te voeren voor het experiment. De output is een interface die gebruikt kan worden in andere modules.

SMOTE
SMOTE, de afkorting voor 'Synthetic Minority Oversampling Technique, is een module die een waarde uit een kolom kan oversamplen. Dit houdt in dat een waarde uit die kolom meermalen gekopieerd wordt, met waardes die logisch zijn voor die kolom.

3.2.4 Sample and Split

Om algoritmes te testen moet eenzelfde soort dataset gebruikt. Maar het is niet verstandig om exact dezelfde data te gebruiken als bij het trainen: dan zou het "leren" neerkomen op herkennen van bestaande informatie. Daarom moet je data splitsen in verschillende sets. Er zijn twee mogelijkheden om dit uit te voeren in AML zelf, namelijk:

Partition and Sample
Splitst de data op in een aantal partities, voor het partitioneren van je dataset in verschillende kleinere datasets. Zo is het mogelijk om een x aantal sets te maken gebaseerd op een percentage (bijv. 10 sets die elk 10% van de data bevatten).

Split Data
Splitst de dataset op een willekeurige manier in twee verschillende datasets. De verhouding waarin deze gesplitst wordt bepaald door de gebruiker. Zo kan deze ervoor kiezen om 70% naar dataset 1 te splitsen en de overige 30% naar dataset 2.

3.2.5 Scale and Reduce

Irrelevante of onlogische data mogen bij Machine Learning niet voorkomen, omdat onjuiste informatie in de dataset funest kan zijn voor de nauwkeurigheid van het algoritme. De volgende opties zijn er om data te verwijderen of aan te passen:

Clip Data
Identificeert en verwijdert data als de waarde boven of onder een bepaalde drempel ligt. Dit is nuttig voor het detecteren van afwijkende of buiten de scope liggende data.

Normalize Data
Normaliseert de data naar een standaard variabele. Hiermee versimpelt de machine de data.

Principal Component Analysis
Deze module wordt gebruikt om data uit een set te analyseren en te comprimeren in een kleinere dataset met minder features. Dit doet het door te kijken welke data zeer relevant is voor het trainen en welke data meer opgekropt kunnen worden.

Quantize Data
Verdeelt een numerieke kolom in verschillende categorieën. Zo kunnen er bijvoorbeeld vier kolommen aangemaakt worden waarin ieder 25% van de data gerepresenteerd wordt. Dit valt te vergelijken met de 'Group Categorical Values' module.

3.3 Selecteren van Features

Machine Learning gebruikt features om de algoritmes te trainen. Het selecteren van de juiste features is dan ook van extreem belang om de testresultaten betrouwbaar te houden. Daarvoor zijn de volgende modules beschikbaar:

Filter Based Feature Selection
Deze module scant alle kolommen en kijkt dan welke features de grootste voorspellingskracht hebben. De gebruiker stelt het aantal features in.

Fisher Linear Discriminant Analysis
Creëert een groep waardes die een combinatie van features kan herkennen. Deze kunnen het best gebruikt worden om twee of meer kolommen te onderscheiden.

Permutation Feature Importance
Controleert in hoeverre de prestaties veranderen als de data uit de dataset willekeurig gehusseld worden.

3.4 Machine Learning Modules

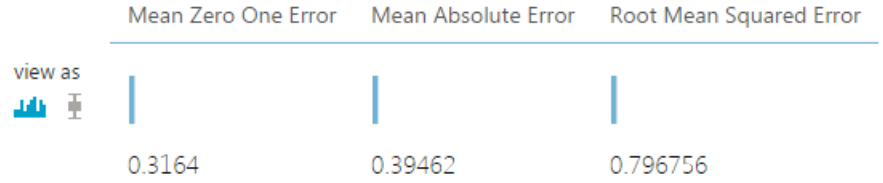
Het gros van AML gebeurt in deze modules. De algoritmes worden besproken in Machine-Learning algoritmes.

3.4.1 Evaluate

De modules in deze categorie worden gebruikt om een getraind algoritme te evalueren, en dus de nauwkeurigheid van dit algoritme te bepalen.

Cross Validate Model
Deze module neemt een ongetraind 'Classification' of 'Regression' model en een dataset als input. Het bouwt een model en geeft een nauwkeurigheids statistiek terug die aangeeft hoe nauwkeurig dit algoritme zal zijn op deze dataset.

Evaluate Model
De standaard evaluatie module. Deze wordt gebruikt om de precisie van een model weer te geven. Afhankelijk van het type algoritme geeft deze module andere waarden.



Figuur 22 Voorbeeld van een Evaluate op een 'Regression' algoritme.

Evaluate Recommender
Wordt gebruikt om de precisie van voorspellingen te bepalen door een 'Recommendation model'. Dit is een model dat streeft naar het voorspellen van de voorkeur van de gebruiker naar een bepaald item uit een dataset.

3.4.2 Score

De modules in deze categorie worden gebruikt om een getraind algoritme toe te passen op data.

Apply Transformation
Wordt gebruikt om een transformatie uit te voeren op input voordat deze gebruikt wordt tijdens het trainen. Sommige modules geven enkel een transformatie terug, die in deze module dus gebruikt kan worden. Een voorbeeld hiervan is de 'Select Columns Transform' module.

Assign Data to Clusters
Gebruikt een getraind 'Clustering' algoritme om data uit een dataset onder te verdelen in de clusters uit het algoritme.

Score Matchbox Recommender
Maakt een voorstel voor een bepaald item uit een dataset, gebaseerd op de gebruiker die meegegeven wordt.

Score Model
Doet een voorspelling door gebruik te maken van een getraind algoritme en input van een dataset.

3.4.3 Train

Deze modules trainen een algoritme.

Sweep Clustering
Deze module traint een cluster algoritme door gebruik te maken van input van de gebruiker. De gebruiker voert in welke parameters benodigd zijn tijdens het trainen, waarna er een getraind cluster model uitkomt.

Sweep Parameters
Deze module traint een algoritme door gebruik te maken van input van de gebruiker. De gebruiker voert in welke parameters benodigd zijn tijdens het trainen, waarna er een getraind cluster model uitkomt.

Train Anomaly Detection
Traint een 'Anomaly Detection' algoritme.

Train Clustering Model
Traint een 'Clustering'algoritme.

Train Matchbox Recommender
Traint een 'Recommender' algoritme.

UserId	MovieId	Rating
1	1	2
1	2	7
2	3	1

Tabel 7 Een voorbeeld van een tabel van gebruikers, en de rating die zij geven aan een item (film). Dit soort datasets zijn ideaal voor Recommender algoritmes.

Train Model
Traint een 'Regression'/'Classification' algoritme door gebruik te maken van een dataset.

3.5 Gebruik van OpenCV

OpenCV is een library die ondersteuning biedt voor het herkennen van afbeeldingen en het verwerken van deze afbeeldingen. OpenCV in AML zorgt ervoor dat je afbeeldingen kan importeren en er een voorgetraind algoritme op los kan laten die bijvoorbeeld gezichten kan herkennen.

3.6 Gebruik van Python

Het is mogelijk om Python scripts te importeren in AML, en deze uit te voeren. AML maakt gebruik van de Anaconda omgeving, dat een aantal belangrijke Python packages bevat.

3.7 Gebruik van R

R is een programmeertaal die veel gebruikt wordt in statistische- data-analyse. Deze taal kan ook in AML gebruikt worden.

3.8 Statistieken

De statistiek van AML maakt het mogelijk om wiskundige formules los te laten op de voorgaande modules. Zo kun je bijvoorbeeld een rapportage maken over data, waarbij het gemiddelde berekend wordt van een feature. De volgende statistische opties zijn beschikbaar in AML:

Apply Math Operation
Deze module wordt gebruikt om berekeningen uit te voeren op een kolom, zoals het afronden van een getal.

Descriptive Statistics
Genereert een simpel statistisch rapport voor iedere kolom in de dataset. Hiermee kun je bijvoorbeeld het gemiddelde van iedere (numerieke) kolom bekijken.

Compute Elementary Statistics
Genereert een rapport waarin meer informatie gevonden kan worden over kolommen die numerieke waardes bevatten, zoals bijvoorbeeld het gemiddelde

Compute Linear Correlation
Deze module wordt gebruikt om een *Pearson Correlation* grafiek te genereren voor iedere mogelijke connectie tussen twee of meer features in een dataset.

Evaluate Probability Function
Genereert een statistiek die gebruikt kan worden voor kansberekeningen.

Replace Discrete Values
Maakt non-numerieke features numeriek door gebruik te maken van een andere kolom (deze kan zowel *Categorical* als *Non-categorical* zijn).

3.9 Tekst analyses

AML biedt de gebruiker de mogelijkheid om een stuk tekst te pakken en dit in AML te stoppen, waar dan analyses op uitgevoerd kunnen worden. Zo kan er bijvoorbeeld een dataset uit tekst gehaald worden. AML maakt hiervoor gebruik van de Vowpal Wabbit library. Deze is gemaakt om tekst te analyseren en daar Machine Learning op toe te passen. De volgende opties kunnen op tekst uitgevoerd worden:

Feature Hashing
Leest features uit een stuk tekst. Zo kan het bijvoorbeeld de frequentie van een woorden achterhalen.

Named Entity Recognition
Deze module wordt gebruikt om de naam van een locatie, organisatie of persoon te herkennen uit een stuk tekst. Het genereert een model van 5 door komma's gescheiden variabelen. Namelijk de rangorde van voorkomen in de module (0 is eerste artikel in de module), de naam van locatie, de startindex van de naam, de lengte van de naam, en wat het representeert.
De drie waardes die een naam kan hebben zijn:

- LOC(ation)
- PER(son)
- ORG(anisation)

Input	Output
Eindhoven is een mooie stad.	0,Eindhoven,0,9,LOC
Daar staat Robert!	1,Robert,11,6,PER
In Sogeti is innovatie top-prioriteit	2,Sogeti,3,6,ORG

Tabel 8 Een voorbeeld van Named Entity Recognition, waarin alle drie de mogelijkheden van herkenning voorkomen.

In de eerste rij van Tabel 8 staat de output "0, Eindhoven, 0, 9, LOC". Dit houdt in dat Eindhoven de eerste naam is die voorkomt (0, Eindhoven). De naam begint op index 0, en is 9 karakters lang (0, 9). De naam is van een locatie, namelijk de stad Eindhoven (Loc).

Score/Train Vowpal Wabbit
Traint of verifieert een Vowpal Wabbit module.

3.10 Conclusie

AML heeft veel verschillende mogelijkheden om data uit een dataset te verwerken en aan te passen. Veel van deze modules zijn echter zeer specifiek en zullen in de meeste experimenten niet gebruikt worden. Voornamelijk het gebruik van R-scripts en Manipulation modules gezien zal worden in een experiment. Dit komt doordat de datasets hiermee gemanipuleerd kunnen worden naar de wens van de gebruiker, en niet specifiek voor één doelstelling ontwikkeld zijn.

4 Machine-Learning algoritmes

Het combineren van de verschillende modules zorgt ervoor dat AML bezig gaat met het leren door het uitlezen en toepassen van ingevoerde data. Er is een bijna eindeloos aantal mogelijkheden om hiermee aan de slag te gaan. Die zijn echter wel te categoriseren in een viertal die van toepassing zijn op Machine Learning.

1. Afwijkingdetectie: als AML getraind is in datatransacties is het mogelijk om hier een test op uit te voeren om te kijken of de data afwijkend gedrag vertonen. Dit is bijvoorbeeld heel handig als er gecontroleerd dient te worden op fraude door financiële instellingen.
2. Multiclass-classificatie (waaronder binaire classificatie): AML kan door middel van Machine Learning een voorspelling te maken op een enkele waarde binnen een dataset: binair of de waarde true/false is, of multiclass wanneer één van de opgegeven waardes in de dataset voorspeld wordt.
3. Clustering: het clusteren van gevallen in een dataset die ongeveer dezelfde karakteristieken hebben. Deze groeperingen kunnen dan gebruikt worden om afwijkingen te detecteren of eventuele voorspellingen te maken.
4. Regressie: wordt gebruikt om de numerieke waarde van een feature te voorspellen. Dit is bijvoorbeeld handig om het weer te voorspellen.

AML maakt gebruik van veel verschillende algoritmes om data te herkennen en daar voorspellingen op te doen. Welk algoritme wanneer gebruikt dient te worden hangt af van wat voor data er gebruikt worden, wat de uitkomst kan zijn (binair / multiclass / numeriek), en wat voor eindresultaat je verwacht.

4.1 Detectie van afwijkingen

AML heeft de mogelijkheid om afwijkingen in data te herkennen door die te analyseren en te kijken of een groep van waardes opmerkelijk zijn of niet voldoen aan de verwachtingen erover. Er zijn twee verschillende modules voor het detecteren van afwijkingen in AML, namelijk:

4.1.1 One-class support vector machine

Een 'Support Vector Machine' (of SVM) is een model dat een analyse uitvoert op ingevoerde data en dan gebruikt kan worden voor classificatie of regressie. Het 'One-class'-gedeelte slaat op het idee dat AML één waarde pakt van een binaire classificatie (bijv. 0 op 0/1) en die als normaal beschouwt, en de ander als afwijkend.

Voordelen
SVM is ideaal om te gebruiken als veel data één van de binaire waardes heeft en weinig tot geen andere. Dit is normaal gedrag bij afwijkingen, aangezien er meestal weinig tot geen resultaten van bijvoorbeeld fraude aanwezig zijn.

Nadelen
SVM heeft echter wel een beperking bij het trainen van data. Het kost veel tijd om een SVM-model te trainen en SVM is niet goed schaalbaar op de hoeveelheid data. Als één van deze factoren aanwezig is, is het verstandig om de andere module toe te passen, 'PCA-Based Anomaly Detection' die ik verderop behandel.

Hoe gaat dit te werk?
Een One Class Support Vector Machine analyseert een gegeven dataset en maakt een voorspelling voor een binaire waarde. Het algoritme creëert een datamatrix die de uitkomsten van elkaar distantieert op grond van de gegeven data. Hoe groter de afstand is tussen de waarden in de matrix, des te accurater de voorspelling is.

Voorbeeld
Bank A gebruikt dit algoritme om fraude te detecteren. Dit doet ze door zoveel mogelijk 'correcte' transacties in het model te stoppen, waarna deze correcte transacties als zodanig herkent. Als SVM een afwijkende set data ontvangt zal hij die als frauduleus beschouwen en een andere waarde geven dan normaal.

4.1.2 PCA-Based Anomaly Detection

Principal Component Analysis (PCA) kan gebruikt worden om data te classificeren. Het wordt voornamelijk gebruikt om uitgebreide data te analyseren vanwege de mogelijkheid tot het herkennen van de diversiteit van de data. (Dallas, 2003)

Voordelen
Dit algoritme werkt uitstekend als de gebruikte features verbindingen hebben met elkaar.

Nadelen
Als de data niet gerelateerd is kan het zijn dat de resultaten uit dit algoritme zeer inaccuraat en afwijkingen herkent terwijl deze er niet zijn.

Hoe gaat dit te werk?
PCA maakt gebruik van de data die aangereikt worden, en past de waardes dusdanig aan dat onnodige data niet gerepresenteerd worden en de relevante data beter weergegeven kunnen worden. Dit doet het door naar "*eigenwaarden*" en "eigenvectoren" in de data te zoeken. De gevonden vectoren en waarden genereren één of meer "generieke lijnen" die je door de data kan trekken: de 'principal components'. Deze worden in plaats van de grote hoeveelheden oorspronkelijke data gebruikt om analyses uit te voeren. Op deze manier destilleert PCA de relevante aspecten van de data om te analyseren, terwijl die toch allemaal gebruikt worden.

Voorbeeld
Door de grote hoeveelheid verschillende soorten data die Bank A ontvangt is ze overgestapt op PCA. De diversiteit van de data is groot, maar er zijn wel relaties tussen. De bank krijgt nu nauwkeurigere resultaten dan eerst.

4.2 Classificatie

Het bepalen tot welke klasse een set data hoort valt onder de categorie classificatie. Dit kan bijvoorbeeld bepalen tot welk geslacht een werknemer hoort, of voorspellen wat voor een opleiding ideaal is voor een student. Ofwel, het voorspellen op basis van een bepaalde categorie in je data, behoort onder classificatie.
Er zijn twee categorieën waarin geclassificeerd kan worden:

- Two class, waarbij er een binair antwoord mogelijk is (man/vrouw)
- Multi-class, waarbij diverse klassen mogelijk zijn als antwoord (bijvoorbeeld kleur: rood / wit / blauw / groen)

De volgende algoritmes zijn beschikbaar voor deze twee categorieën:

- Averaged Perceptron (alleen Two-class)
- Bayes Point Machine (alleen Two-class)
- Boosted Decision Tree (alleen Two-class)
- Decision Forest
- Decision Jungle
- (Locally-Deep) Support Vector Machine
- Logistic Regression
- Neural Network
- One-vs-All Multiclass

4.2.1 Averaged Perceptron

Averaged Perceptron (AP) is een oudere/simpelere vorm van een Neural Network, waarbij de input wordt geclassificeerd naar de verschillende mogelijke outputs, waarbij er gewicht wordt gelegd op een bepaalde waarde in de input.

Voordelen
Dit algoritme is te gebruiken als er een lineair verschil aanwezig is tussen de twee verschillende klassen. Als dit niet het geval is kan er beter een Neural Network algoritme gebruikt worden, die dieper in constructie zijn, maar wel veel meer tijd kosten om te trainen dan een Perceptron netwerk.

Nadelen
Werkt alleen naar behoren als er een lineair verschil aanwezig is.

Hoe werkt dit?
Een perceptron algoritme pakt een paar waardes en baseert daar zijn voorspelling op. Het gaat alle trainingsdata één voor één af en controleert of de huidige voorspellingswaarden correct zijn. Als dat niet zo is past het zijn warden dusdanig aan dat het op de gefaalde voorspelling een betere voorspelling kan doen. Zodra alle data gecontroleerd zijn, loopt het algoritme nog een keer vanaf invoer 1 naar de laatste, tot een gegeven aantal iteraties gedaan zijn.

Doordat het algoritme er zo vaker doorheen loopt zal het zichzelf vaker aanpassen, wat betekent dat het steeds accurater wordt naarmate het meer iteraties uitvoert.

Voorbeeld
Bank B maakt gebruik van classificatie om te controleren of een persoon wellicht een klant wil worden voordat ze contact met hem of haar opneemt. De bank verwacht dat de informatie die ze invult een lineair verband heeft met de vraag of hij wel of geen klant wil worden. Dit algoritme geeft met output Ja / Nee aan of de persoon wellicht klant wil worden.

4.2.2 Bayes Point Machine

Bayes Point Machine (BPM) is een Bayesiaanse vorm van kansberekening. Dat betekent dat het een subjectieve vorm van kansberekening is. In plaats van dat dit algoritme objectieve feiten pakt en beslist wat het beste resultaat is, maakt BPM daadwerkelijk een gok naar de beste mogelijkheid, door middel van kansberekeningen.

Voordelen
Er zijn geen parameters die ingesteld dienen te worden om dit algoritme te laten functioneren. *Overfitting* is ook geen risico bij dit algoritme.

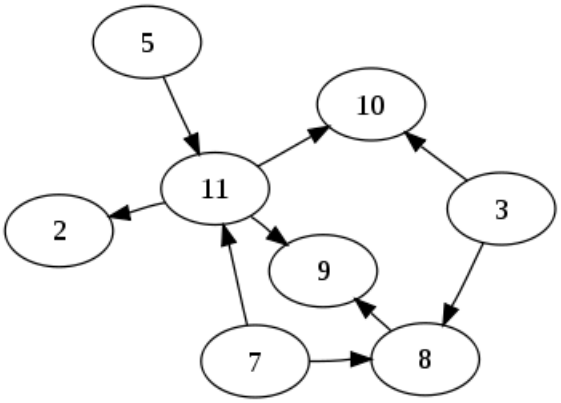
Nadelen
BPM is subjectief en gebaseerd op uitkomsten van de vorige voorspellingen. Het kan hierdoor rare uitkomsten vertonen. (100 keer een munt opgooien, waarbij het precies 50/50 verdeeld is, de kans hierop is heel klein).

Hoe werkt het?
BPM gebruikt het 'propagation-message-passing'-algoritme. Dit maakt geen gebruik van het antwoord uit zijn algoritmes, maar eerder van de kennis of veronderstellingen die het trekt uit de mogelijkheden van het antwoord. Het doet een schatting van wanneer welk antwoord correct is, waarna het de data bestudeerd en deze schatting nog een keer doet.

Voorbeeld
Bank B merkt dat de informatie die ze meegaf aan het Averaged Perceptron-algoritme veel te grof was om accuraat te gebruiken. Omdat Bank B niet weet welke kolommen nuttig zijn heeft ze de hele dataset gebruikt. Bayes Point is niet gevoelig voor overfitting, dus vond ze dit een goede keuze.

4.2.3 Boosted-Decision-Trees

Boosted Decision trees zijn een 'Directed Acyclic Graph', vanaf nu DAG genoemd, wat inhoudt dat het een 'lineair' algoritme is. Dit maakt gebruik van keuzemomenten om tot een resultaat te komen. Voor iedere set data die door het getrainde model gaat loopt hij een reeks keuzes af, waarna het door het maken van deze keuzes, een antwoord geeft. (Sink, 2011)



Figuur 23 Voorbeeld van een Directed Acyclic Graph

Er wordt aangeraden om BDT enkel toe te passen als de verschillende features aan elkaar gekoppeld zijn en een duidelijke onderscheid te maken tussen de keuzes die het algoritme moet maken.

Voordelen

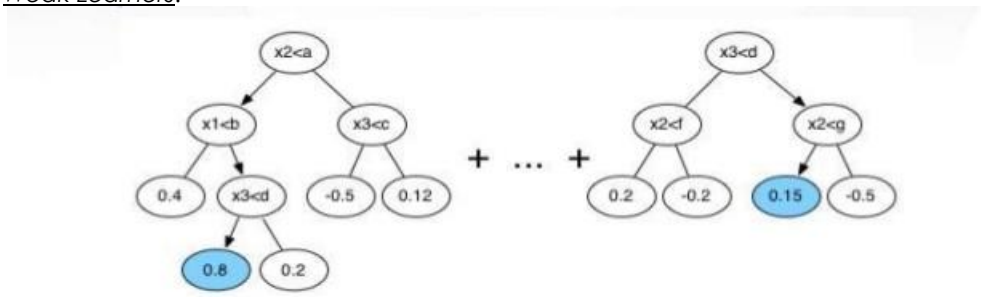
Dit algoritme is snel getraind en zeer accuraat als de data gerelateerd is aan elkaar. Het is zeer leesbaar voor een gebruiker, met eenvoudige regels. (Wilku, 2012)

Nadelen

BDT is gevoelig voor overfitting. Non-numerieke data zijn moeilijker voor een Decision tree te begrijpen. (Wilku, 2012) Het verbruikt veel geheugen in de training en tijdens de voorspelling, waardoor dit algoritme niet aan te raden is voor grote datasets. (Microsoft, 2015)

Hoe werkt het?

De Boosted Decision Tree analyseert de trainings-data en kijkt naar overeenkomsten in deze data in combinatie met andere gegeven features. Hierop maakt het een reeks aan decision trees, zogenaamde *Weak Learners*.



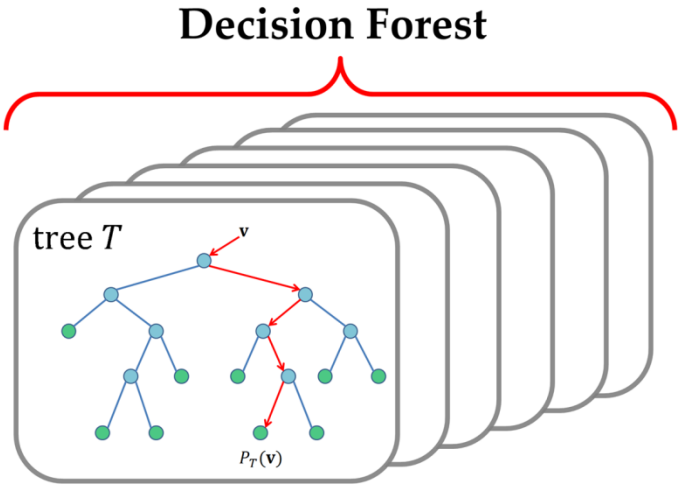
Figuur 24 Een reeks van decision trees, die gebruikt worden voor het boosted tree algoritme.

Voorbeeld

Bank B heeft ontdekt welke parameters meegegeven kunnen worden om beter te herkennen of een persoon klant wil worden. Omdat deze data goed met elkaar gerelateerd waren besloot de bank om een 'Boosted Decision Tree' algoritme te gebruiken.

4.2.4 Decision Forest

Een Decision Forest maakt een aantal DAG's aan. Voor iedere set data die door het getrainde model gaat controleert het welke van deze DAG's het beste werkt en loopt het een reeks keuzes af, waarna het een antwoord geeft gebaseerd op de DAG en het pad dat de data is afgegaan in deze DAG.
Er wordt aangeraden om deze methode enkel toe te passen als de verschillende features aan elkaar gekoppeld zijn, en een duidelijk onderscheid maken tussen de keuzes die het algoritme moet maken.
Het grote verschil tussen Decision Forests en Boosted-Decision-Trees is dat de eerste parallel berekend kan worden Dit betekent dat iedere tree in theorie in een andere service gebruikt kan worden. De Boosted-Decision-Trees maken echter gebruik van diverse kleine DAG's, wat ervoor zorgt dat minder ingewikkelde datasets hierdoor efficiënter voorspeld kunnen worden. (Xu, 2013)



Figuur 25 Een voorbeeld van een decision Forest, waarbij meerdere decision trees gebruikt worden om een voorspelling te maken.

Voordelen
Decision Forests zijn efficiënt tijdens het trainen en het voorspellen door het weinige gebruik van geheugen en processorkracht van dit algoritme. Ook is het zeer robuust als het aankomt op het zeer willekeurige features. (Microsoft, 2015)

Nadelen
Decision Forests maken gebruik van een DAG en kunnen dus niet goed werken met non-numerieke features. (Wilku, 2012)

Hoe werkt het?
De Decision Tree analyseert de trainings- data en kijkt naar overeenkomsten in deze data in combinatie met andere gegeven features. Zo creëert het een aantal 'decision trees' waarin de data getest kan worden.

Voorbeeld
Het energielabel heeft een aantal labels van A tot G, en de overheid wil op basis van de informatie van de woningen voorspellingen doen over de energielabels. Er zijn numerieke features aanwezig waardoor het mogelijk is voor dit algoritme om efficiënt te werken.

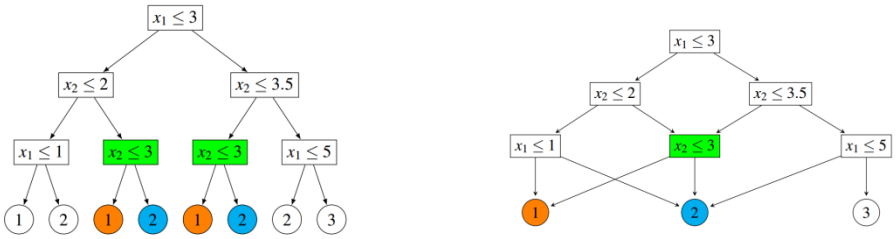
4.2.5 Decision Jungle

Een "Decision Jungle" is een DAG waarbij nodes van de tree samengevoegd kunnen worden om geheugen te besparen tijdens het uitvoeren van een voorspelling.

Voordelen
Een Decision Jungle verbruikt minder geheugen tijdens het testen door het vermengen van de decision trees in een compactere status. Dit zorgt voor een betere prestatie, zij het ten koste van langere trainingstijd. (Microsoft, 2015)

Nadelen
Door het vermengen van de trees is een langere trainingstijd vereist. Decision Jungles maken gebruik van decision tree, en kunnen dus ook niet goed werken met non-numerieke features. (Microsoft, 2015)

Hoe werkt dit?
Een Decision-Jungle-algoritme is bijna exact hetzelfde als een decision tree, met de uitzondering dat een leaf in meer dan 1 boom voor kan komen. (Pohlen, 2015)



Figuur 26: Het verschil tussen een decision tree(Links) en decision jungle (Rechts)

Voorbeeld

De overheid merkt dat er teveel geheugen wordt gebruikt tijdens het berekenen van de energielabels. Hierom hebben ze besloten om een Decision Jungle te gebruiken. De training duurt dan wel langer, maar is efficiënter voor het gebruik, aangezien deze voorspelling vaak uitgevoerd gaat worden.

4.2.6 (Locally-deep) support vector machine

SVM is praktisch gezien hetzelfde als de One-class variant, waarbij Locally-deep enkel sneller en minder accuraat is dan de voorgenoemde.

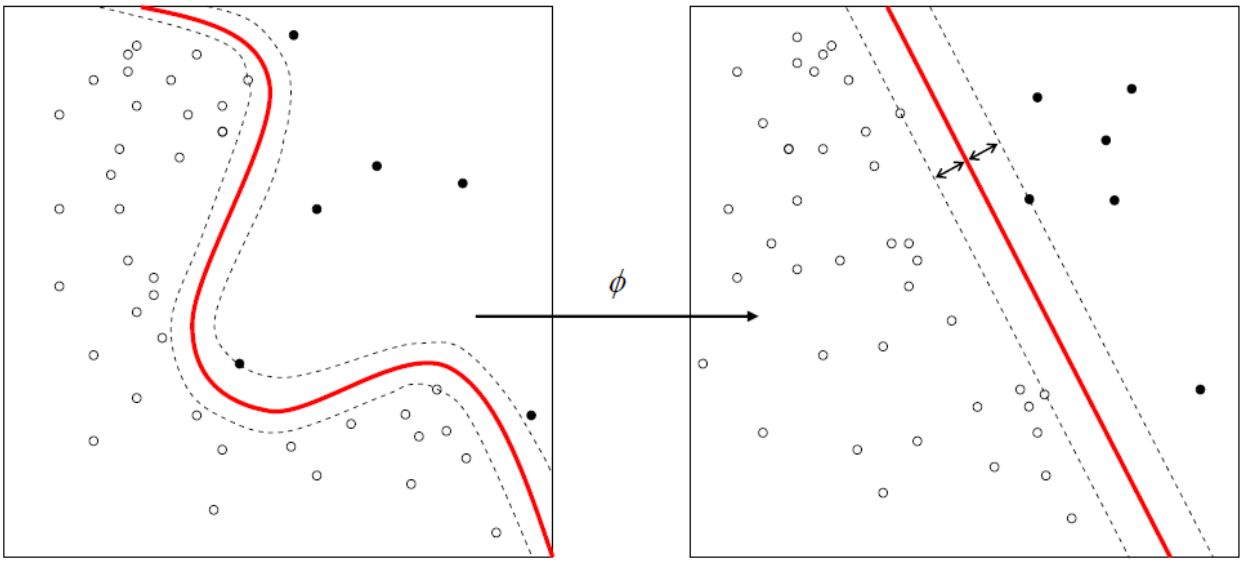
Dit algoritme is prima te gebruiken als er een binair probleem is, waarbij lineaire classificaties (bijv. PCA) niet goed presteren. Er dient wel rekening gehouden te worden met het verlies van voorspellingsnauwkeurigheid doordat de trainingstijd korter is dan andere modellen. (Bernhard Schölkopf A. J., 2000)

Voordelen
Extreem snelle prestaties en korte trainingstijden.

Nadelen
Voorspellingskracht is niet het sterkste punt van dit algoritme.

Hoe werkt dit?
Indien er geen lineaire classificatie aanwezig is zal het de non-lineaire gegevens lineair proberen neer te zetten voor het classificeren van twee verschillende uitkomsten op een enkele soort data. Deze data worden dus vergeleken alsof er een ideale lineaire classificatie aan te geven is. (Statsoft, 2016)

Als er wel een lineaire classificatie mogelijk is zal het deze stap overslaan, en de grootste marge pakken tussen de twee verschillende uitkomsten, zoals te zien is in Figuur 27



Figuur 27: Een non lineaire classificatie wordt omgezet naar een lineaire classificatie door de algoritmes van SVM

Voorbeeld
De overheid merkt dat de Decision jungle niet goed werkt voor de voorspellingen en wil een tijdelijke snelle oplossing voor het probleem. Ze maakt gebruik van SVM en offert daarmee voorspellingskracht op totdat ze een ander algoritme gevonden heeft wat past bij het voorspellen van het label.

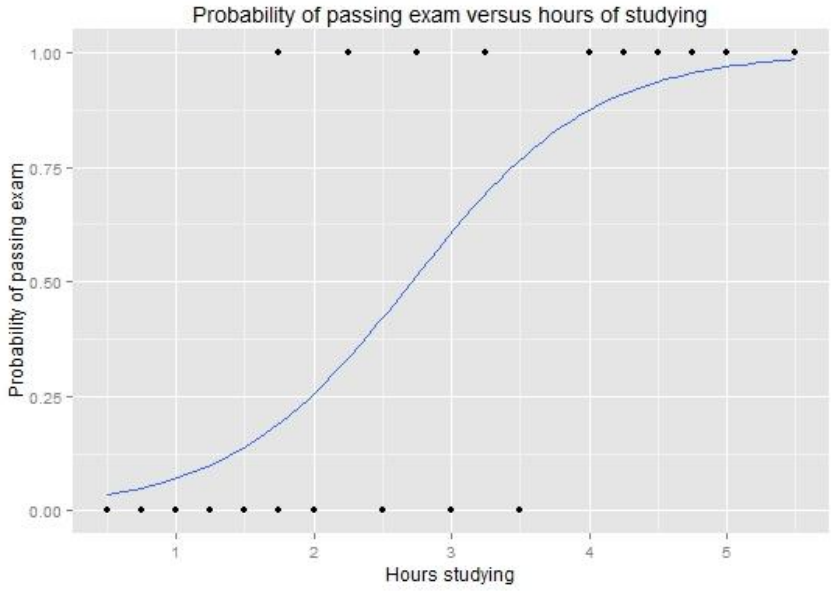
4.2.7 Logistic Regression

Dit algoritme meet de relatie tussen de te voorspellen waarde en de onafhankelijke overige waardes in de dataset. Hier maakt het een diagram van waarin de kans op een bepaalde waarde voorspeld wordt op grond van de gegeven waardes.

Voordelen
Logistic Regression geeft een numerieke uitkomst, waarbij de kans groot is dat deze dicht bij de correcte waarde ligt.

Nadelen
Logic Regression werkt alleen op numerieke waardes.

Hoe werkt het?
In het geval van 'Two Class Classification' is dit een *binomiale verdeling*, waarbij het algoritme de kans aangeeft dat de gevraagde waarde x is (bijv. 0.85 = 85% kans op x).



Figuur 28 Regressie berekent hier de kans dat iemand slaagt voor een test, vergeleken met hoeveel uur hij/zij studeert.

Bij 'Multiclass Classification' geeft Regression een numerieke waarde terug met wat het voorspelt dat de correcte waarde in dat geval is.

Voorbeeld
Na wat onderzoek heeft de overheid ontdekt dat het wellicht een mogelijkheid is om regressie te gebruiken om te voorspellen of een bepaald energielabel mogelijk is voor een huis. Logistic Regression is een goede oplossing, maar de data dienen dusdanig gemanipuleerd te worden dat tijdens de training de energielabels als een nummer meegegeven worden i.p.v. een letter. De uitkomst zal dan ook een nummer zijn. (Bijv. A = 1, B = 2, etc.)

4.2.8 Neural network

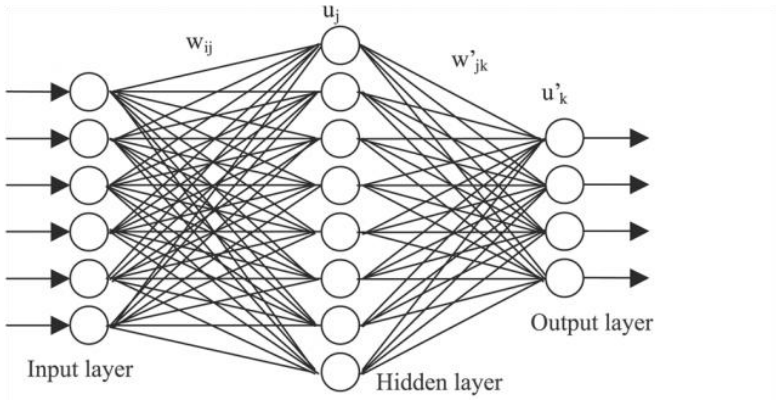
Een neural network algoritme maakt een netwerk van nodes aan die een bepaald gewicht hebben voor het bepalen van de waarde van de te voorspellen feature. Het gewicht van een node wordt bepaald tijdens het trainen van het algoritme.

In een node bekijkt het algoritme bepaalde eigenschappen van de data en geeft die dan door aan de volgende node. Een node kan input ook verwerken naar output (de uitkomst), of er berekeningen op uitvoeren.

Voordelen
Een Neural network is en zeer efficiënt en nauwkeurig alternatief voor decision trees.

Nadelen
De trainingstijd is lang in vergelijking met andere algoritmes. Het algoritme is extreem gevoelig voor overfitting.

Hoe werkt dit?
De werking van dit algoritme is ongeveer hetzelfde als een Decision Jungle, met als uitzondering dat een neural node naar meer dan twee andere nodes kan verwijzen. Verder is een Neural network in staat om zichzelf nieuwe dingen aan te blijven leren, net zoals het menselijk brein nieuwe connecties kan leggen. (Templeton, 2015)



Figuur 29: Een voorbeeld van een neuraal netwerk, met 6 vormen van input, 8 nodes, en 4 outputs.

Voorbeeld
De overheid heeft de dataset die voor training wordt gebruikt geoptimaliseerd voor Machine Learning. De features zijn beperkt zodat overfitting wordt voorkomen, en de overheid is bereid om een lange trainingstijd tegemoet te zien. Het Neural Network is nu een goed alternatief.

4.2.9 One-vs-All Multiclass

Dit algoritme is heel simpel in theorie, maar vereist redelijk wat rekenkracht. Er wordt een Two-class classificatie model gekoppeld en voert deze in dit algoritme uit.

Voordelen
Bij one-vs-all-multiclass kun je een two-class-model gebruiken voor multiclass-classificatie.

Nadelen
Dit algoritme is niet aan te raden als de te voorspellen feature veel verschillende waardes heeft.

Hoe werkt dit?
Dit algoritme voert een aantal binaire algoritmes uit op de data die ingevoerd worden en optimaliseert dan het algoritme voor iedere klasse. Hierna worden alle modellen samengevoegd in één model.

4.3 Clustering

Clustering maakt gebruik van iteratieve technieken om een dataset in verschillende delen op te delen, waarbij de data van iedere 'cluster' ongeveer hetzelfde zijn of dezelfde karakteristieken vertonen. Door middel van deze clusters is het makkelijker om een relatie te vinden tussen de data. Clustering is daarom een goed alternatief in de eerste fasen van machine learning.

Op het moment van schrijven is 'K-Means Clustering' de enige optie om clusters te gebruiken met AML. Dit maakt gebruik van 'centroids', een gemiddeld punt, voor iedere cluster, waarmee de input wordt vergelijken. De cluster waar de data het beste bij passen is de uitkomst van dit algoritme.

Voorbeeld
De overheid heeft besloten dat ze ieder energielabel als een 'cluster' van data zien. Ze heeft nu een cluster voor ieder energielabel en kan dus clustering toepassen.

4.4 Regressie

Regressie-algoritmes worden gebruikt om een (voornamelijk) numerieke waarde te voorspellen aan de hand van gegeven input. Zo kun je bijvoorbeeld de prijs van een huis voorspellen op grond van bekende prijzen van huizen in de buurt, met ongeveer dezelfde grootte en conditie.
De volgende algoritmes zijn beschikbaar voor regressie in Azure:

- Linear Regression
- Bayesian linear Regression
- Boosted decision tree Regression
- Decision Forest Regression
- Fast Forest Quantile Regression
- Neural network Regression
- Ordinal Regression
- Poisson Regression

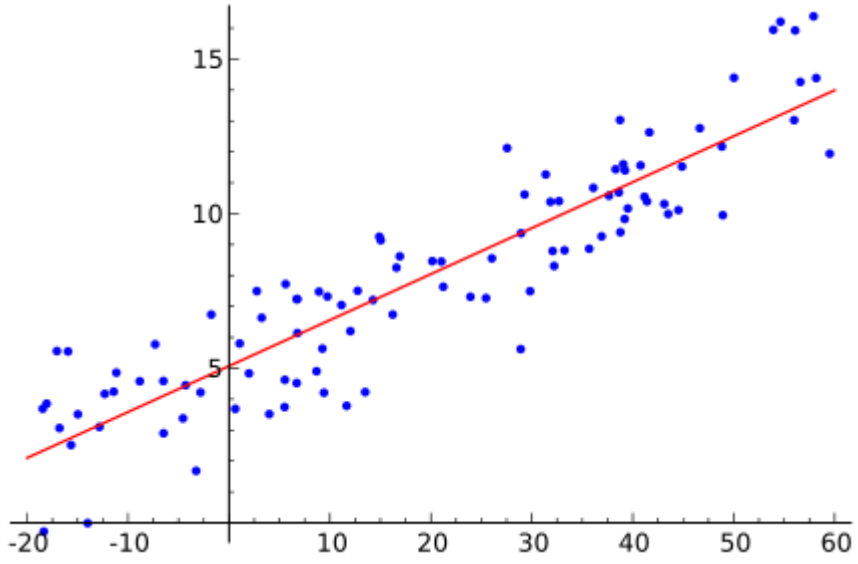
4.4.1 Linear regression

Dit algoritme wordt gebruikt wanneer er een lineair verband aanwezig is tussen de gegeven data en de te voorspellen feature.

Voordelen
Geeft een numerieke waarde gebaseerd een gemiddelde waarde op een grafiek. Als er een lineair verband is in de data is dit een zeer nauwkeurig algoritme.

Nadelen
Afwijkende data vormen een probleem voor dit algoritme. Veel informatie is nodig om een nauwkeurige lineair verband te maken.

Hoe werkt het?
Dit algoritme probeert een lineair verband te leggen tussen de afhankelijke waarde (de voorspelling) en de onafhankelijke waarde (bekende data). Dit doet het door alle data op een grafiek te leggen en hier een lijn door te trekken naar wat een logische uitkomst is. (Linear regression)



Figuur 30: Voorbeeld van een lineaire regressielijn.

Voorbeeld
De overheid stapt over van classificatie naar lineaire regressie, omdat dit de eenvoudigste vorm van regressie is.

4.4.2 Bayesian linear Regression

Dit algoritme maakt een voorspelling gebaseerd op kennis van eerder ingevoerde data

Voordelen
Er zijn geen parameters die ingesteld dienen te worden om dit algoritme te laten functioneren. *Overfitting* is ook geen kwestie bij dit algoritme.

Nadelen
De voorspelling is subjectief doordat het systeem een 'gok' maakt op de uitkomst. Dit is niet per se inaccurater, maar kan wel vreemde resultaten leveren.

Hoe werkt het?
Met gebruik van subjectieve kansberekeningen wordt er een regressie model gemaakt, wat inhoudt dat gebaseerd op de data die ingevoerd wordt, er gegokt wordt wat de uitkomst zal zijn, gebaseerd op wat het algoritme berekend heeft tijdens het trainen met test-data.

Voorbeeld
De overheid stapt over van lineaire regressie naar Bayesian linear regression, omdat ze toch nog een gevoel heeft dat overfitting een probleem kan worden.

4.4.3 Boosted Decision Tree Regression

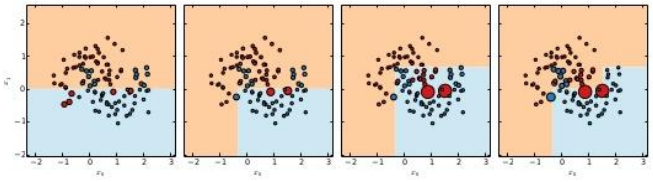
Boosted Decision Tree Regression maakt gebruik van DAG's om een voorspelling te maken naar de waarde van de feature.

Voordelen
De training gaat snel en de uitkomst is zeer accuraat als er correlatie is tussen de ingevoerde data en de te voorspellen feature. Niet-lineair verband tussen data en te voorspellen feature kan herkend worden. (Prettenhofer & Louppe, 2014)

Nadelen
Het algoritme maakt gebruik van DAG's en is daarom onbetrouwbaar als er veel verschillende te voorspellen categorieën zijn. Er is intensieve voorbereiding nodig voor het instellen van het algoritme en het trainen duurt lang. (Prettenhofer & Louppe, 2014)

Hoe werkt het?
Door het toepassen van 'Multiple Additive Regression Trees' (MART) Gradient Boosting creëert dit een reeks van decision trees.

Gradient boosting houdt in dat er over een grafiek als het ware een transitie wordt gemaakt tussen twee verschillende waardes. Dit kan vergeleken worden met een gradiënt leggen over de data (Figuur 31). Aangezien Gradient boosting diverse gradients kan maken is het mogelijk om in tegenstelling tot lineaire vergelijkingen een aantal vlakken van een grafiek te raken. (Li)



Figuur 31 Laat de mogelijkheden zien van gradient boosting voor het differentiëren van data.

Figuur 31 geeft een aantal voorbeelden van gradient boosting. Er wordt gekeken naar de hoeveelheid data als dit over een grafiek uitgestreken zou worden, en trekt als het ware een lijn door de data die waarde 1 van waarde 2 splitst. Het verschil met lineaire algoritmes is dat deze lijn niet recht hoeft te zijn.

Net als 'Boosted Decision Tree Classification' maakt deze methode gebruik van decision trees, ofwel 'weak learners', om het algoritme te trainen. Hierbij bepaalt het een zogenaamde '*loss function* gradient'.

Gebaseerd op deze gradiënt genereert het algoritme de volgende tree. Als het aangegeven aantal trees is gegenereerd pakt het de decision tree met de laagste loss function.

Voorbeeld
De overheid heeft ontdekt dat de data non-lineair is maar wil toch gebruik maken van regressie. Hierom

gebruikt ze nu Gradient Boosting, omdat deze non-lineaire data makkelijker kan herkennen dan andere algoritmes.

4.4.4 Decision Forest Regression

Decision forests zijn vrijwel identiek aan de classificatievariant en maken gebruik van een aantal trees om een voorspelling te doen voor de ingevoerde data.

Voordelen
Efficiënt tijdens het trainen en het voorspellen door het weinige gebruik van geheugen en processorkracht van dit algoritme. Ook is het zeer niet gevoelig voor willekeurige features.

Nadelen
Forest Regression maakt gebruik van DAG's en kan dus niet goed werken met niet-numerieke features.

Hoe werkt het?
Forest Regression kijkt naar overeenkomsten in de trainingsdata in combinatie met andere gegeven features. Zo creëert het diverse 'decision trees' waarin de data getest kan worden.

4.4.5 Fast Forest Quantile Regression

Deze methode lijkt sterk op Decision Forest, maar stopt de data in een categorie. Een voorbeeld is de conversie van het energieverbruik van een huis naar een energielabel.

4.4.6 Neural Network Regression

Regressienetwerken zijn vrijwel identiek aan de classificatievariant. Dit maakt gebruik van een netwerk van nodes om een voorspelling te maken.

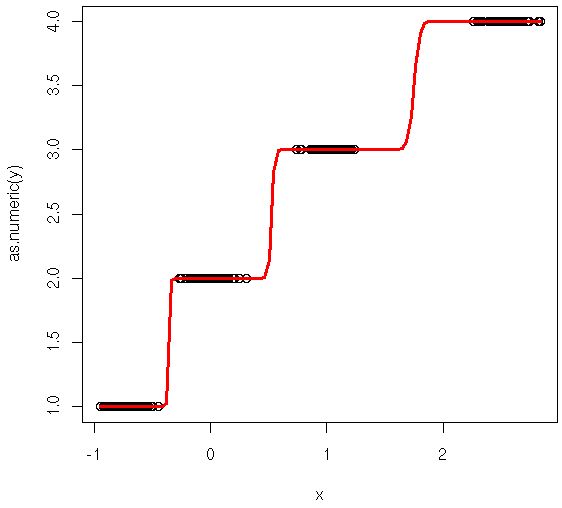
4.4.7 Ordinal Regression

Deze vorm van regressie wordt gebruikt als er sprake is van een rangschikking in de waarden, zoals in een competitie, waarbij nummer 1 voor de 1e plaats staat, 2 voor de 2e plaats, etc.

Voordelen
Ordinal Regression is zeer efficiënt in het genereren van een ranking-algoritme.

Nadelen
Ordinal Regression werkt alleen als de te voorspellen feature een rangorde kent.

Hoe werkt het?
Het algoritme traint zichzelf om antwoord te geven op de vraag: 'verwacht ik dat deze data hoger is dan plaats x?'. Indien het antwoord ja is zal het de vraag nog eens stellen met een hogere waarde voor x. Zodra het antwoord nee is wordt x als voorspelde waarde gegeven.



Figuur 32 Een voorbeeld van een grafiek waarin duidelijke rangen aanwezig zijn.

Voorbeeld
De overheid heeft besloten dat het energielabel als een ranking gezien kan worden en wil gebruik maken van Ordinal Regression. Dit geeft vrijwel altijd het beste resultaat voor rangschikkingen en is daarom een goede keuze.

4.4.8 Poisson Regression

Poisson Regression maakt gebruik van de Poissonverdeling om te voorspellen wat de frequentie is dat iets gaat gebeuren. Er mogen in de trainingsdata geen negatieve waarden voorkomen.

Voordelen
Kan accurate voorspellingen maken gebaseerd op frequenties van informatie.

Nadelen
De methode werkt alleen met frequenties.

Hoe werkt het?
Het algoritme probeert voor iedere feature de optimale waardes te vinden door de '*log-likelyhood*' voor iedere parameter te maximaliseren.

Voorbeeld
Het voorspellen van de hoeveelheid brieven die er bezorgd worden op een bepaalde dag.

4.5 Conclusie

Er zijn veel verschillende algoritmes om voorspellingen te maken gebaseerd op data. Het is aan de ontwikkelaar om een keuze te maken welk algoritme hij kiest, gegeven de voor- en nadelen ervan en de uitkomst die hij wenst. De vier verschillende categorieën maken de keuze makkelijker, maar die verschilt uiteindelijk per experiment.

Voor het VEL-project binnen Sogeti is Ordinal Regression in theorie de beste aanpak, omdat het gaat om een rangschikking van energielabels.

5 Toepassen AML in applicaties

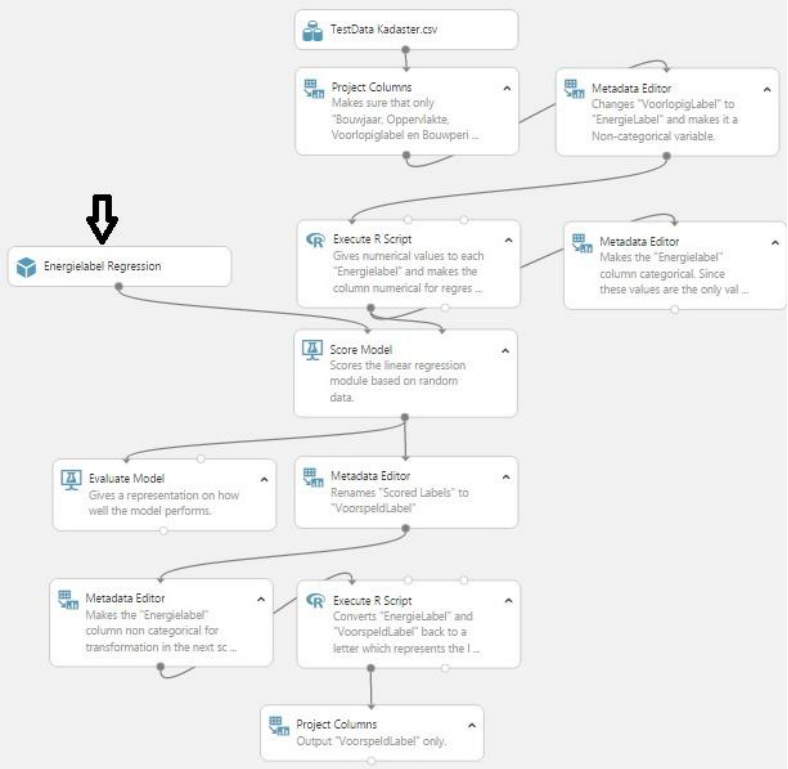
Na de training is een algoritme in staat een voorspelling te doen. Dit hoofdstuk beantwoordt de deelvraag: hoe is AML toepasbaar in applicaties?
De enige mogelijkheid die AML biedt om te gebruiken in code, is door je model te deployen als webservice. De input en output van het model worden dan dusdanig aangepast zodat het mogelijk is om het te gebruiken in applicaties. Het model moet worden geprepareerd voor de gebruiksfase, aangezien het algoritme zich anders gedraagt tijdens de training dan in productie.

Figuur 33 Een afgerond experiment dat klaar is om omgezet te worden naar een Webservice. Iedere keer dat dit experiment draait wordt het algoritme opnieuw getraind.

Dit is het model dat aangepast gaat worden voor productie. Het krijgt als input informatie over huizen en gaat na manipulatie van de data een voorspelling doen over het energielabel ervan. Het model

5.1 Opslaan van een getraind model

Het is mogelijk om een model dat getraind is op te slaan als een model, dat altijd gebruikt kan worden in andere experimenten. Het grote voordeel hiervan is dat het model dan niet opnieuw getraind wordt.

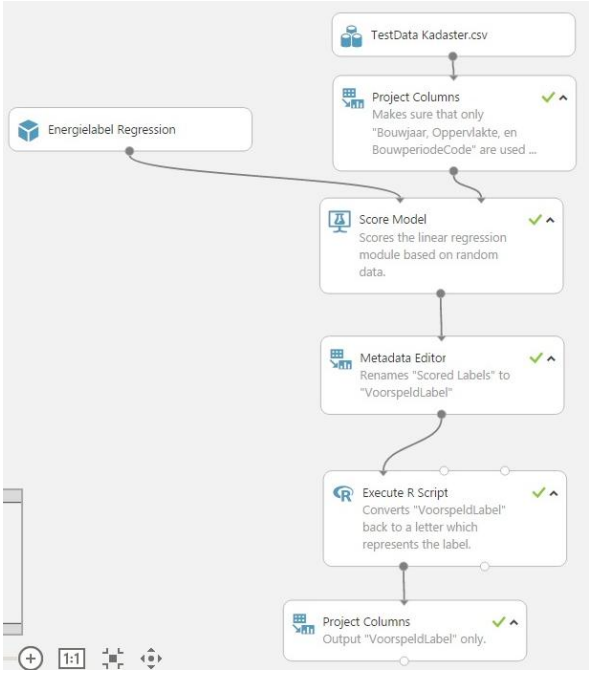


Figuur 34: Het model is opgeslagen, en opnieuw geplaatst in het experiment. Het algoritme is klaar voor gebruik en hoeft dus niet opnieuw getraind te worden.

Hier zijn het "Linear Regression"-algoritme en de "Train model"-module weggehaald nadat het getrainde model is opgeslagen. Omdat er nu een getraind algoritme is is er geen splitsing van data meer nodig. Deze module is daarom ook verwijderd.

5.2 Verwijderen van label in experiment

Het *label* is onbekend bij de gebruiker als hij een voorspelling wil doen. Deze dient dan ook uit de aangeboden dataset verwijderd te worden. Hiermee wordt er gegarandeerd dat het model geen voorkennis heeft van de data die het voorspelt en kan AML ook geen problemen veroorzaken omdat één van de input-features niet aanwezig is.



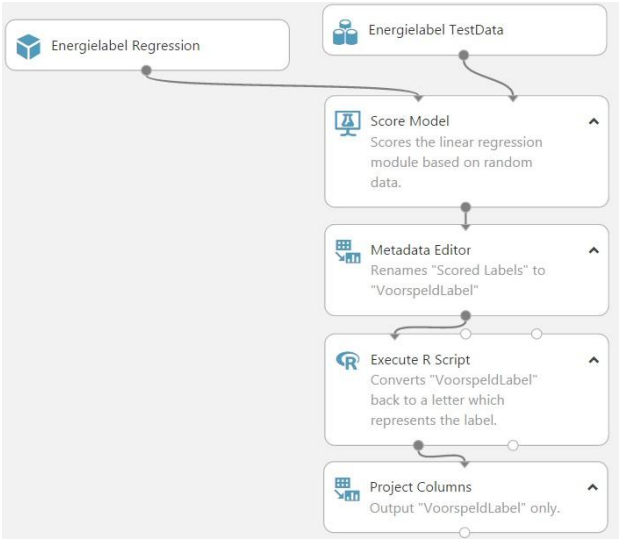
Figuur 35 Alle referenties en modules die werken met het label "Energielabel" zijn verwijderd of aangepast.

Als eerste is het "Energielabel" uit de "Project Columns" module gehaald. Deze feature werd aangepast in de "Metadata-Editor" en "Execute R Script" module. Omdat dit de enige feature was in deze modules zijn deze twee ook verwijderd.
Na het scoren van het algoritme is ook nog een "Metadata-Editor" die zich specifiek richt op het "Energielabel". Deze module is ook verwijderd.

Als laatste is de "Execute R Script" module aangepast zodat de kolom "Energielabel" niet meer gebruikt wordt.

5.3 Opslaan van de dataset

Om enkel de nuttige informatie als input te gebruiken kan de dataset opgeslagen worden als een kleinere dataset waarin alleen de gebruikte features zitten.



Figuur 36 Het experiment met de nieuwe dataset.

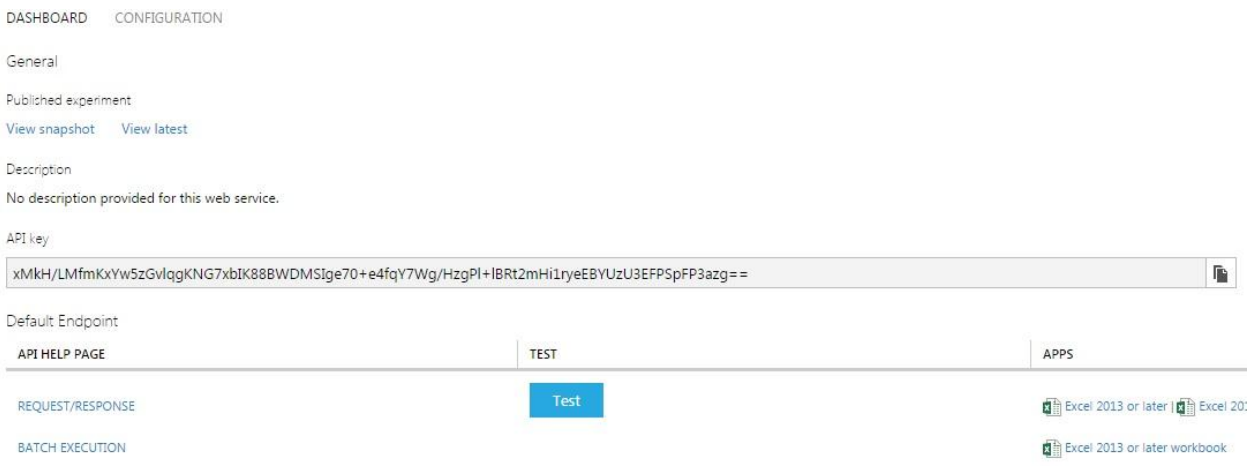
De output die uit de "Project Columns" module komt is opgeslagen als een dataset, waarna deze dataset gebruikt wordt in het experiment. Hierdoor zijn enkel de gebruikte features beschikbaar.

5.4 Deployen van de Webservice

Als laatste wordt de webservice gedeployd en kan deze ingesteld en genoemd worden naar wens van de gebruiker. De webservice heeft nu een in- en output die gebruikt kunnen worden door externe applicaties.

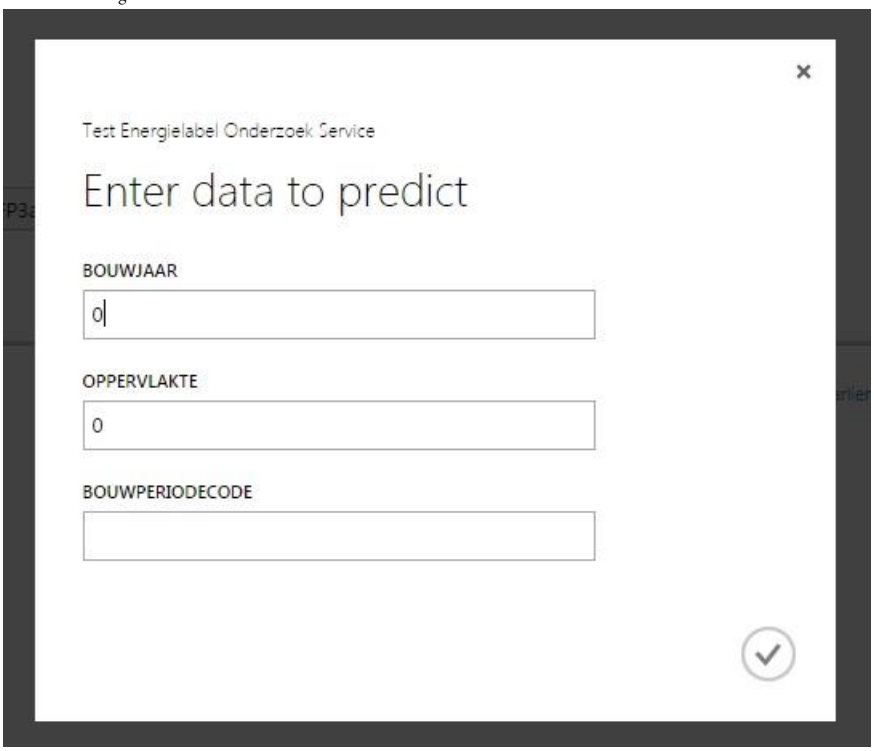
Figuur 37 Het experiment met de webservice input en output.

5.5 De webservice



Figuur 38 Het webservice configuratie scherm.

Zodra de service geconfigureerd is kan de webservice geopend worden, waarna deze het bovenstaande scherm toont. Hierin staat de API key die gebruikt wordt om deze webservice aan te roepen.
Het is ook mogelijk om op deze pagina de webservice te testen. Dit wordt gedaan door op de knop test te drukken en dan informatie in te voeren in de popup die verschijnt.



Figuur 39 Nadat op de test knop is gedrukt wordt de mogelijkheid geboden om testdata in te voeren.

5.6 Conclusie

AML is een sterke tool om data te voorspellen. Er zijn echter wat configuraties nodig als de gebruiker het wil gebruiken in een applicatie. Zodra deze aanpassingen zijn toegepast is het gebruik in de applicatie een kwestie van het aanroepen van een *Request-Response service*.
Voor de applicatie die geschreven wordt voor het VEL project is het toepassen van webservices prima te gebruiken. Dit levert verder geen problemen op voor de applicatie.

6 Toepasbaarheid in applicaties

AML is een tool die gebruikt kan worden om data te voorspellen. Er zijn echter wat beperkingen die ervoor zorgen dat deze tool niet voor iedere applicatie toepasbaar is.

Voorspellingen
Het is zeer onwaarschijnlijk dat AML een model genereert dat 100% accurate antwoorden geeft. Net als andere analyses op zogenaamde *"Big Data"* zal een applicatie die zekerheid nodig heeft (bijvoorbeeld een banktransactie) geen gebruik moeten maken van AML.

Hoeveelheid data
Machine Learning heeft data nodig om een algoritme te trainen naar een model. De hoeveelheid data die hiervoor nodig is, hangt af van de verschillende categorieën in de te voorspellen feature.

Gevoeligheid data
De data worden gestuurd naar Azure. Dit betekent dat een extra partij als 'data processor' aanwezig is, wat kan betekenen dat een contract met een klant niet toestaat dat AML gebruikt wordt voor deze doelstellingen. Het bewaren van privacy van de cliënt is wettelijk verplicht en dient te allen tijde gegarandeerd te worden.
Indien er een mogelijkheid geboden wordt door Azure om deze berekeningen lokaal uit te voeren zal dit geen kwestie zijn, doordat de data niet naar een derde partij verzonden wordt.

6.1 Conclusie

AML kan gebruikt worden wanneer de beperkingen die hierboven genoemd zijn niet van belang zijn voor de applicatie waarin AML toegepast gaat worden.
Voor VEL is het geen probleem, maar er dient wel rekening mee gehouden te worden dat de data gevoelig kunnen zijn. Er zijn om deze reden geanonimiseerde data gebruikt, zodat de data niet gekoppeld kunnen worden aan een persoon of instantie.

7 Antwoord op de hoofdvraag

AML is een goede optie om analyses uit te voeren op data en toe te passen in applicaties. Zolang de klant geen gevoelige data gebruikt of de output van een applicatie op een voorspelling niet kan of wil vertrouwen, is AML een krachtige tool als extensie voor een applicatie.
Nu is de vraag of er een andere manier van Machine Learning is die deze gebreken kan omzeilen. Dit vereist een onderzoek naar andere ML tools, en de mogelijkheden hiervan.
Voor het VEL project is AML bruikbaar als referentiekader voor applicaties. Het kan gebruikt worden als referentiekader om te verifiëren of de berekende energielabels nog kloppen met de verwachte resultaten.

8 Bibliografie

Bernhard Schölkopf, A. J. (2000). *Microsoft Research.* Opgehaald van New support vector algorithms: http://www.stat.purdue.edu/~yuzhu/stat598m3/Papers/NewSVM.pdf
Bernhard Schölkopf, J. C.-T. (2000, September 18). *Estimating the support of a high-dimensional distribution.* Opgeroepen op November 29, 2015, van Microsoft Research: http://research.microsoft.com/pubs/69731/tr-99-87.pdf
Dallas, G. (2003). *Georgemdallas.wordpress.com*. Opgehaald van Principal Component Analysis 4 Dummies: https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/
Jose, C. (sd). *Microsoft research.* Opgeroepen op January 4, 2016, van Local Deep Kernel Learning for Efficient Non-linear SVM Prediction: http://research.microsoft.com/en-us/um/people/manik/pubs/Jose13.pdf
Li, C. (sd). *A Gentle Introduction to Gradient Boosting.* Opgeroepen op Januari 29, 2016, van ccs.neu.edu: http://www.ccs.neu.edu/home/vip/teach/MLcourse/4_boosting/slides/gradient_boosting.pdf
Linear regression. (sd). Opgeroepen op Januari 29, 2016, van Stat.yale.edu: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm
Michaelg2015. (2015, August 17). Exam pass logistic curve.jpeg.
Microsoft. (2015, Augustus 24). *MSDN Machine Learning Studio*. Opgeroepen op 2015, van MSDN: https://msdn.microsoft.com/en-us/library/azure/dn905974.aspx
Microsoft, M. L. (2015, Februari 17). *Big Learning made easy, with counts!* Opgeroepen op 2015, van Technet.
Milivojevic, Z. (2009). *Digital Filter Design.*
Minka, T. P. (2004, Januari 30). *Microsoft Research.* Opgehaald van Expectation Propagation for Approximate Bayesian Inference: http://research.microsoft.com/en-us/um/people/minka/papers/ep/minka-ep-uai.pdf
Pohlen, T. (2015, Maart 8). *Geekstack.* Opgeroepen op December 23, 2015, van geekstack.net/resources/public/downloads/tobias_pohlen_decision_jungles_slides.pdf
Prettenhofer, P., & Louppe, G. (2014, Februari 24). *Gradient Boosted Regression Trees in sckit-learn.* Opgeroepen op Februari 2, 2016, van Slideshare: http://www.slideshare.net/DataRobot/gradient-boosted-regression-trees-in-scikitlearn
Sink, E. (2011). *Directed Acyclic Graph.* Opgeroepen op Januari 29, 2016, van ericsink: http://ericsink.com/vcbe/html/directed_acyclic_graphs.html
Smith, J. O. (2007). *Introduction to Digital Filters with Audio Applications.* Opgehaald van Introduction to Digital Filters with Audio Applications: Linear-Phase Filters: https://ccrma.stanford.edu/~jos/fp/Linear_Phase_Filters_Symmetric_Impulse.html
Statsoft. (2016). Support Vector Machines. In *Statsoft Textbook* (p. Web). Statsoft.
Templeton, G. (2015, October 12). Artificial neural networks are changing the world. What are they?
Wang, Q. (2014, Maart 10). *Decision Tree and Decision Forest - File Exchange - MATLAB Central.* Opgeroepen op Januari 29, 2016, van Mathworks: http://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/39110/versions/6/screenshot.jpg
Wilku, J. S. (2012, October 18). *Decision Trees.* Opgeroepen op Januari 29, 2016, van Slide share: http://www.slideshare.net/jagjitsinghwilku/decision-trees-14788315
Xu, T. (2013, Mei 31). How do random forests and boosted decision trees compare?

Bijlage C: Algoritme training

Versie informatie

Versie	Datum	Bijzonderheden	Auteur
0.1	16-03-2016	Ontwikkeling document	Robert Donner
1.0	31-03-2016	Verwerken feedback, Spelling en grammatica check	Robert Donner

Inhoudsopgave

1 Inleiding

Dit document beschrijft de ondernomen stappen van het gebruik van Azure Machine Learning (AML) en de kadaster dataset die gebruikt wordt in het VEL project. Het legt stapsgewijs uit wat er gedaan werd om het huidige algoritme te trainen en de bruikbaarheid van dit algoritme.

Het doel van dit document is om te veronderstellen welke algoritmes toe te passen zijn op de dataset van het VEL project en worden de benodigdheden voor de algoritmes besproken.

Dit document veronderstelt dat de lezer voorkennis heeft van AML.

2 Keuze van het algoritme

Als eerste dient er een algoritme gekozen te worden voordat er aanpassingen van data toegepast kunnen worden om dit algoritme te trainen. Welk algoritme er gekozen is wordt beschreven in dit hoofdstuk.

2.1 Algoritme types

Er zijn vier verschillende types algoritmes in AML waar uit gekozen kan worden om voorspellingen uit te voeren:

- Classification
- Anomaly Detection
- Regression
- Clustering

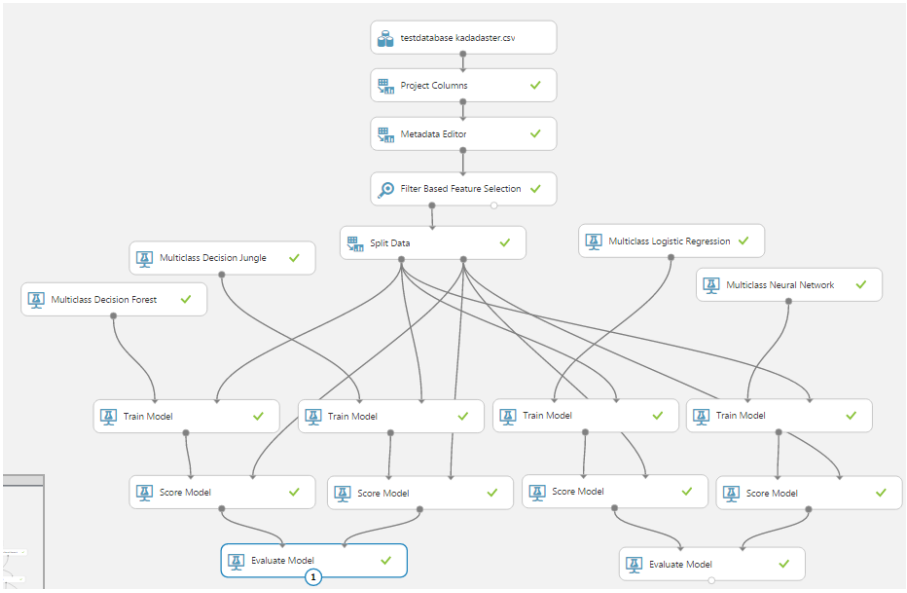
VEL vereist een algoritme dat een label voorspelt. Er zijn meerdere labels die voorspeld kunnen worden. Aangezien er geen afwijkingen zijn in de data valt de categorie 'Anomaly Detection' af. Dit zorgt ervoor dat de keuze blijft tussen de overige drie types.

2.1.1 Classification

Classificeert de data en toont dan ook specifiek dat een rij data bij een specifiek label zal horen. Het voordeel hiervan is dat er een specifiek label gekozen wordt.

De volgende algoritmes zijn gekozen voor verdere analyse:

- Decision Forest
- Decision Jungle
- Logistic Regression
- Neural Network



Figuur 40 Experiment dat de vier genoemde Classification algoritmes test.

2.1.2 Regression

Regressie betekend dat er een numerieke waarde voorspelt wordt. Dit zorgt ervoor dat een afwijking in de data er niet voor zorgt dat er een afwijkend label voorspeld wordt. Regressie werkt beter met numerieke data en er dient van te voren een modificatie gemaakt te worden in de dataset zodat de labels gerepresenteerd worden met een cijfer.

De volgende algoritmes zijn gekozen voor verdere analyse:

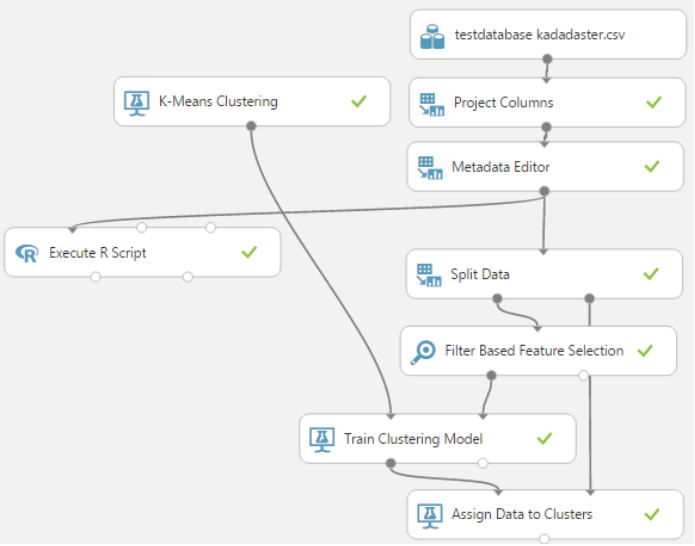
- Ordinal Regression
- Decision Forest Regression

Figuur 41 Experiment dat de vier genoemde Regression algoritmes test.

2.1.3 Clustering

Clustert de data gebaseerd op het label. Hiermee wordt een rij data onder een categorie geclusterd. In het geval van het VEL project zijn het de energielabels die als een categorie data gelden.

Er is maar één vorm van clustering in AML, namelijk K-Means Clustering.



Figuur 42 Experiment dat Clustering algoritme test.

3 Kiezen van de dataset

Om te kunnen werken met de gegeven data van het kadaster, is het nodig om kolommen van de dataset of aan te passen, of niet te gebruiken in de training van één van bovenstaande algoritmes.

rows	columns
100000	22

Id	BAGId	Postcode	Huisnummer	Toevoeging	Straatnaam	Plaats	EigenaarId	EigenaarType	Bouwjaar

Figuur 43 Een klein deel van de originele dataset, die 22 kolommen aan data bevat.

22 kolommen aan data zorgt voor overfitting van een algoritme, wat aangeeft dat deze dataset te groot is voor gebruik in de training. Door gebruik te maken van 'Filter Based Feature Selection' zijn er kolommen geselecteerd die passend zijn voor het trainen van een algoritme om het energielabel te voorspellen.

VoorlopigLabel	Bouwjaar	WoningTypeId	Huisnummer	Oppervlakte

Figuur 44 De kolommen gebruikt uit de originele dataset

Figuur 9 toont de geselecteerde kolommen van 'Filter Based Feature Selection'. Er is gekozen voor vier kolommen voor training.

In de praktijk zijn er maar één of twee kolommen data nodig, dit kan de gebruiker echter de illusie geven dat er niet genoeg data ingevuld wordt voor een goede voorspelling.

VoorlopigLabel	Bouwjaar	WoningTypeId

Figuur 45 De eigenlijke kolommen waarmee het label het nauwkeurigst voorspeld wordt.

4 Resultaat

Het toepassen van de kolommen op ieder algoritme zoals vermeld in 2. Keuze van het algoritme leverde de volgende resultaten op.

4.1 Regression

Hier kwamen wisselende resultaten uit. Het ordinale regressie algoritme heeft een accuratie van 100%, wat ongebruikelijk is voor een voorspellingsalgoritme. Verder ziet de Decision Forest Regression er ook uit als een alternatief, met een afwijking van slechts 0.0066%.

Ordinal Regression

Mean Zero One Error	Mean Absolute Error	Root Mean Squared Error
0	0	0

Decision Forest Regression

Mean Absolute Error	Root Mean Squared Error
0.000066	0.005894

4.2 Classification

Hier kwamen onverwachte resultaten uit. Zo zijn er twee algoritmes met een verdacht hoog voorspelgehalte (Decision Forest en Decision Jungle met 100%) een algoritme met een verwachte voorspellingskracht (Neural Network met 87.6%) en een algoritme met teleurstellende precisie (Logistic Regression met 43.7%). De eerste twee algoritmes vertonen een ernstig verdachte nauwkeurigheid tegenover de twee andere algoritmes.

Decision Forest

Overall accuracy	1
Average accuracy	1
Micro-averaged precision	1
Macro-averaged precision	1
Micro-averaged recall	1
Macro-averaged recall	1

Decision Jungle

Overall accuracy	1
Average accuracy	1
Micro-averaged precision	1
Macro-averaged precision	1
Micro-averaged recall	1
Macro-averaged recall	1

Logistic Regression

Overall accuracy	0.429657
Average accuracy	0.837045
Micro-averaged precision	0.429657
Macro-averaged precision	NaN
Micro-averaged recall	0.429657
Macro-averaged recall	0.283496

Neural Network

Overall accuracy	0.875515
Average accuracy	0.964433
Micro-averaged precision	0.875515
Macro-averaged precision	0.882326
Micro-averaged recall	0.875515
Macro-averaged recall	0.854136

4.3 Clustering

Clustering is niet werkend gekregen met de dataset. Er ontstonden foutmeldingen die het gebruik van dit algoritme ingewikkelder maakten. En gebaseerd op de vorige resultaten is er ook besloten om hier geen verdiepend onderzoek naar te verrichten.

Conclusie

Het is duidelijk dat de VEL dataset in staat is om voorspellingen uit te voeren. Een nauwkeurigheid van 100% is echter wel een voorspelling waar de resultaten twijfelachtig zijn. Dit is doordat er niet verwacht wordt van een voorspelling om zo nauwkeurig te zijn.

Zowel Regression als Classification zijn een ideaal hulpmiddel om de voorspellingskracht toe te passen van AML. In deze situatie is Classification echter beter te gebruiken, omdat voor deze algoritmes minder manipulatie van de dataset nodig is.

Het advies is om Classification algoritmes te gebruiken. En de twee beste algoritmes zijn de ‘Decision Jungle’ en ‘Decision Forest’, maar door de verdacht hoge waardes is het advies om ‘Neural Network’ te gebruiken als algoritme. Dit heeft een lagere nauwkeurigheid (87.5%), maar ligt meer in de richting van wat er verwacht wordt van een dusdanig algoritme.

Bijlage D: Requirements document

SOGETI

Versie informatie

Versie	Datum	Bijzonderheden	Auteur
1.0	15-01-2016	Aanmaak requirements document	Robert Donner

Inhoudsopgave

1 Inleiding

1.1 Doel

Dit document beschrijft de Requirements die gesteld zijn door Sogeti, in samenwerking met de ontwikkelaar waar de Azure Machine Learning Sogeti (ARMLEtS) applicatie aan zal moeten voldoen. In dit document worden de User requirements van deze applicatie besproken. Deze eisen zullen fungeren als de acceptatiepunten voor de applicatie. Verder zullen deze eisen meegenomen worden in het ontwerp van de applicatie, alvorens deze geïmplementeerd dient te worden.

1.2 Scope

De applicatie is bedoelt als Proof of Concept (PoC) dat aantoont dat het mogelijk is om Azure Machine Learning (AML) in een .Net omgeving te gebruiken. Het zal een object versturen naar Azure, die dan een voorspelling uitvoert en deze terugstuurt naar de applicatie. ARMLEtS zal hierna met deze data een nieuw object creëren.

2 Algemene beschrijving

2.1 Product perspectief

Er zal gebruik gemaakt worden van een dataset van het Kadaster. Deze dataset bevat woninginformatie. Deze informatie wordt gebruikt door AML om een algoritme te trainen om een label te kunnen voorspellen. Deze dataset wordt enkel gebruikt voor de training en zal geen invloed uitoefenen op het PoC.

2.2 Functies

De applicatie zal de volgende functies kunnen uitvoeren:

- Functies waarmee de gebruiker informatie van gebouwen kan invullen en reviewen. Meer informatie over deze functies is te vinden in PARAGRAAF
- Functies waarmee de gebruiker een gebouw kan labelen, door middel van AML. Dit update de interface zodat ook deze gebouwen bekeken kunnen worden door de gebruiker. Informatie over deze functies is te vinden in PARAGRAAF
- Functies waarmee gebouwen verwijdert kunnen worden uit de applicatie. Meer informatie kan gevonden worden in PARAGRAAF

2.3 Kenmerken gebruiker

Aangezien de applicatie een PoC is zal de gebruiker de applicatie niet regelmatig gebruiken. Het wordt gebruikt als bewijsstuk voor latere implementaties van AML in projecten binnen Sogeti. Er wordt geen kennis van de gebruiker verwacht over de mogelijkheden van AML.

2.4 Beperkingen

Er zijn geen beperkingen opgelegd voor de gebruiker. Alle functionaliteit is bereikbaar voor iedere gebruiker.

2.5 Aannames en afhankelijkheden

De applicatie vereist een internetverbinding om volledig te functioneren. Zonder deze verbinding is het namelijk niet mogelijk om een verbinding te maken met AML. Voor de rest is de applicatie onafhankelijk van andere applicaties.

3 Requirements

3.1 User Requirements

3.1.1 Gebouwen zonder label

- **UGZ1:** De gebruiker kan een gebouw zonder label aanmaken.
  - **UGZ1.1:** Het gebouw krijgt standaard informatie mee.
- **UGZ2:** De gebruiker kan een gebouw zonder label verwijderen.
  - **UGZ2.1:** De gebruiker moet bevestigen dat deze het gebouw wilt verwijderen.
- **UGZ3:** De gebruiker kan de informatie van een gebouw zonder label aanpassen.
  - **UGZ3.1:** De gebruiker kan geen foutieve informatie invullen
    - **UGZ3.1.1:** De gebruiker wordt geïnformeerd over de foutieve data.
    - **UGZ3.1.2:** De gebruiker krijgt een voorbeeld van correcte data.
- **UGZ4:** De gebruiker kan een gebouw zonder label versturen naar AML.

3.1.2 Gebouwen met label

- **UGL1:** De gebruiker kan een gebouw met label verwijderen.
  - **UGL1.1:** De gebruiker moet bevestigen dat deze het gebouw wilt verwijderen.

3.2 System Requirements

3.2.1 Azure Machine Learning

- **SAML1:** Ontvangen data wordt gebruikt om een label te voorspellen voor het verzonden gebouw.
  - **SAML1.1:** Indien de data incorrect is wordt de gebruiker geïnformeerd
  - **SAML1.2:** Indien de data niet verzonden kan worden wordt de gebruiker geïnformeerd over deze kwestie.
- **SAML2:** Het voorspelde label wordt met het gebouw teruggestuurd naar de applicatie.

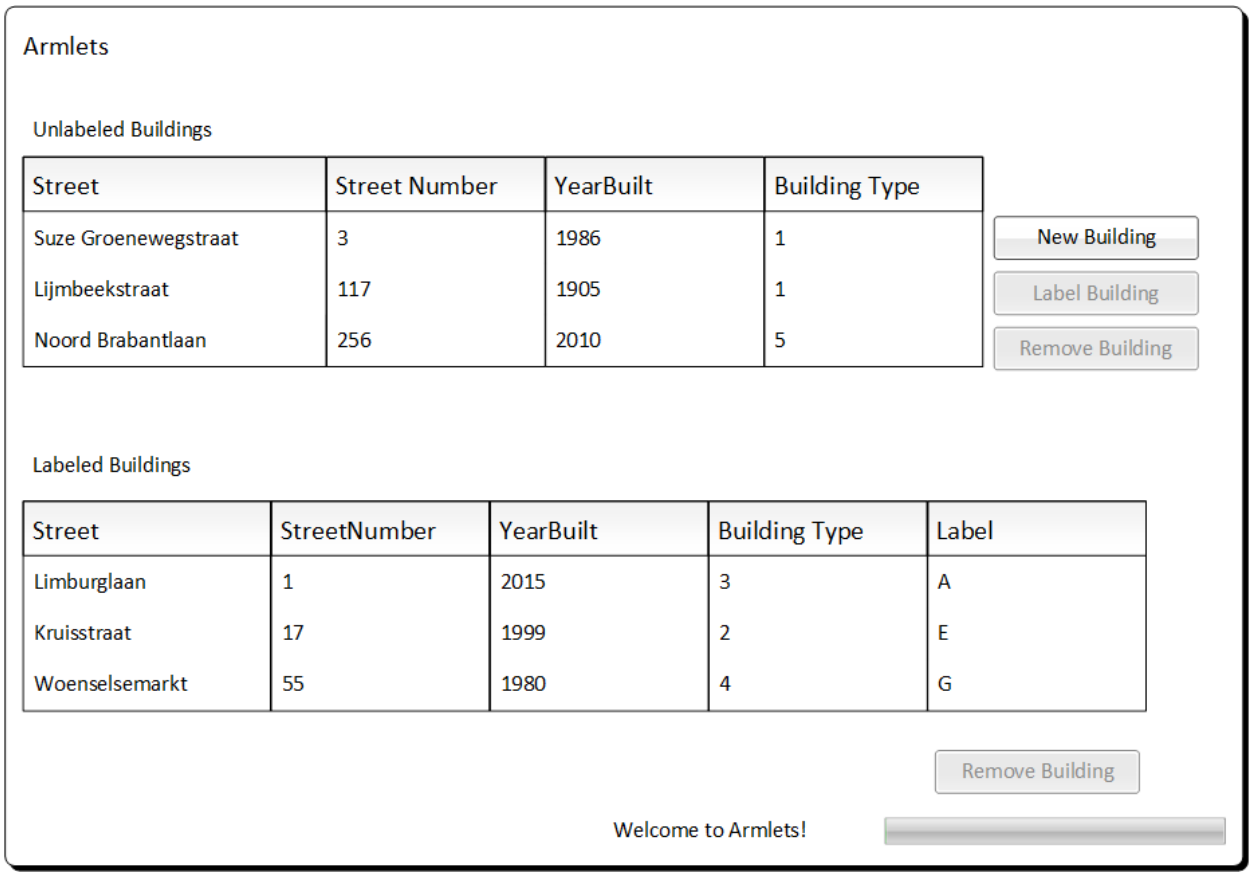
3.2.2 Notificaties

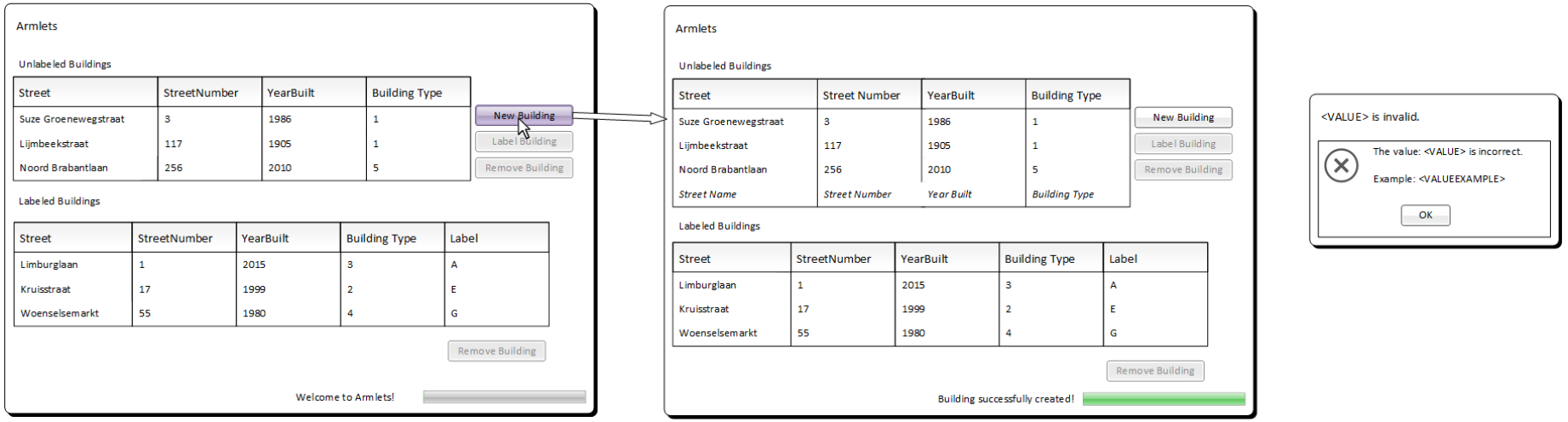
- **SNO1:** De gebruiker wordt geïnformeerd van veranderingen in de applicatie.

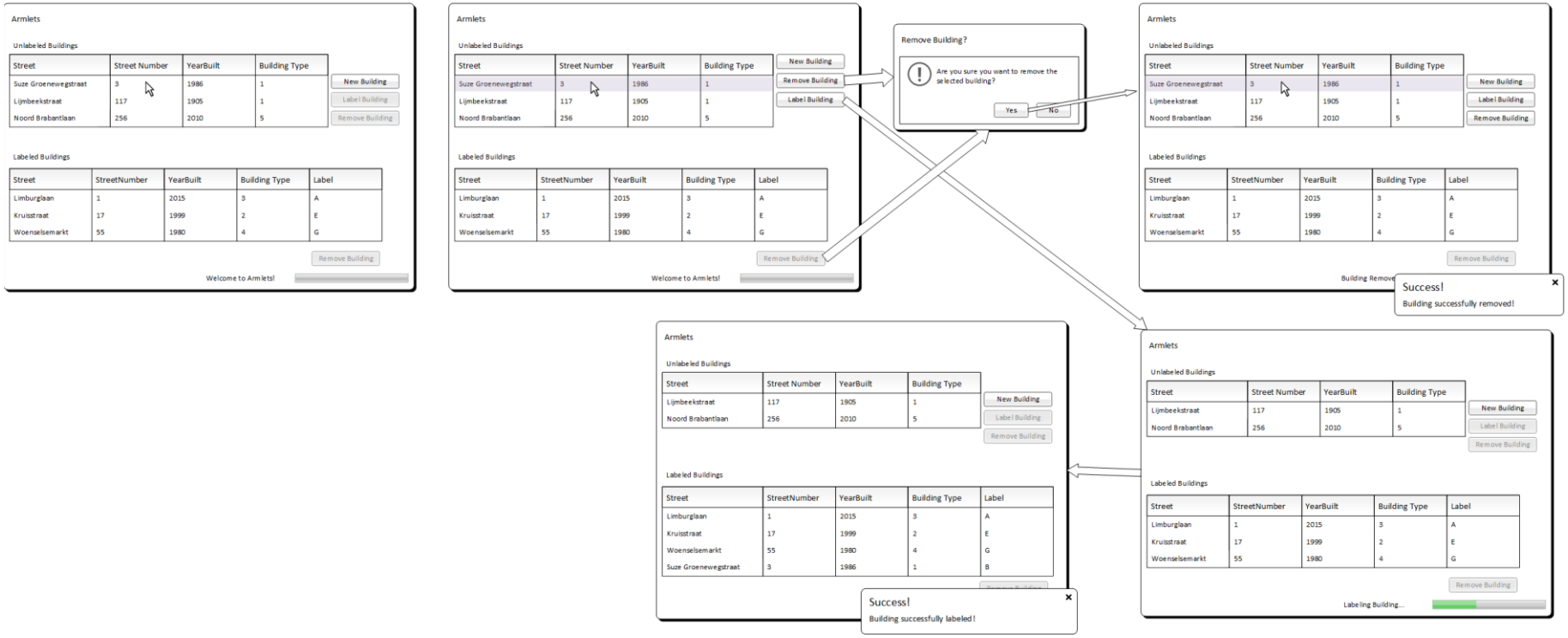
3.2.3 Gebouwen

- **SGO1:** Een gebouw heeft een bouwjaar.
  - **SGO1.1:** Het bouwjaar mag niet hoger zijn dan het huidige jaar.
- **SGO2:** Een gebouw heeft een oppervlakte.
  - **SGO2.1:** De oppervlakte kan geen negatieve waarde hebben.
- **SGO3:** Een gebouw heeft een type.
  - **SGO3.1:** Het type is een getal tussen de 1 en de 6.
- **SGO4:** Een gebouw heeft een huisnummer
  - **SGO4.1:** Het huisnummer kan geen negatieve waarde hebben.

Bijlage E: Wireframes




Armlets
Unlabeled Buildings
Street
StreetNumber
YearBuilt
Building Type
Suze Groenewegstraat 3 1986 1
Lijmbeekstraat 117 1905 1
Noord Brabantlaan 256 2010 5
New Building
Label Building
Remove Building
Labeled Buildings
Street
StreetNumber
YearBuilt
Building Type
Label
Limburglaan 1 2015 3 A
Kruisstraat 17 1999 2 E
Woenselsemarkt 55 1980 4 G
Remove Building
Welcome to Armlets!
Armlets
Unlabeled Buildings
Street
Street Number
YearBuilt
Building Type
Suze Groenewegstraat 3 1986 1
Lijmbeekstraat 117 1905 1
Noord Brabantlaan 256 2010 5
Street Name
Street Number
Year Built
Building Type
New Building
Label Building
Remove Building
Labeled Buildings
Street
StreetNumber
YearBuilt
Building Type
Label
Limburglaan 1 2015 3 A
Kruisstraat 17 1999 2 E
Woenselsemarkt 55 1980 4 G
Remove Building
Building successfully created!
<VALUE> is invalid.
The value: <VALUE> is incorrect.
Example: <VALUEEXAMPLE>
OK



[3] Bijlage E: Wireframes