

Projectbeschrijving Project 1

“De gezondheidszorg”

Inhoudsopgave

Inleiding	3
Opdracht	3
Doel	3
Onderdelen	4
<i>Data verzamelen</i>	4
Aanvulling en duiding	4
<i>Data transformatie</i>	4
<i>Data interpretatie</i>	4
Exploratie	4
Modelleren	5
<i>Datapresentatie</i>	5
Deliverables	6
<i>Projectevaluatie</i>	7
IT voor AI	7
Toegepaste wiskunde	7
Professionele vaardigheden	7
<i>Toetsing</i>	8
Tips & Tricks	9
Bijlagen	10
<i>Daily Kata Starter Questions</i>	10
Bronnen	10

Inleiding

Een aantal wetenschappers uit Rotterdam, Leiden en Delft maakt al gebruik van AI om diagnoses en prognoses preciezer te maken dan ooit tevoren. Een bijkomend voordeel is dat hierbij ook minder medisch ingrijpend onderzoek nodig zal zijn. Daarnaast kan AI gebruikt worden om processen in de gezondheidszorg te verbeteren. Hierbij kan gedacht worden aan het voorspellen van *no shows* bij afspraken of de ligduur van patiënten op de IC-afdeling. Zo heeft Davy van de Sande, promovendus Intensive Care Medicine, onlangs samen met het Erasmus MC en SAS een machine learning model ontwikkeld dat voorspelt welke patiënten na een operatie in het ziekenhuis moeten blijven. Het is duidelijk dat AI veel kan betekenen voor de gezondheidszorg.

Opdracht

Medisch Centrum Randstad is een van de grote ziekenhuizen in Regio Rijnmond. Dagelijks wordt door artsen veel data verzameld over patiënten. Het ziekenhuis wil met een deel van deze informatie een pilot doen om te kijken of ze op basis van levensstijl een voorspelling kunnen doen hoe oud iemand wordt, maar ook of ze gerichte adviezen kunnen geven hoe de levensverwachting te verbeteren. Voor deze pilot is de data over de levensstijl van een groot aantal patiënten beschikbaar gesteld. Deze voorspelling en adviezen moeten op een begrijpelijke manier zichtbaar zijn in een applicatie voor patiënten. Omdat het project nog volop in ontwikkeling is, is het belangrijk dat er constant nieuwe data en variabelen aan de dataset toegevoegd moeten kunnen worden.

Doel

Dit project bestaat uit meerdere onderdelen:

1. Verzamelen van gegevens vanaf een dynamisch gegenereerde webpagina
2. Transformeren van die gegevens, zodat deze geschikt worden voor modelleren
3. Achterhalen hoe de levensverwachting afhangt van aanleg en levensstijl
4. Presenteren en voorspellen van de levensverwachting

Het eerste doel is om een kwalitatief model te ontwikkelen dat het verband tussen leefstijl en leeftijd beschrijft. Als er nieuwe data beschikbaar komt, moet het model eenvoudig bijgewerkt kunnen worden. Er moet dus een automatische data pipeline gebouwd worden.

Rondom dit model wordt modulaire applicatie gebouwd die geschikt genoeg is om in een artspraktijk of ziekenhuis gebruikt te kunnen worden. De applicatie moet dus zelfstandig kunnen draaien op een pc of laptop.

Onderdelen

Data verzamelen

Op een [fictieve site van het Medisch Centrum Randstad](#) vinden jullie de data van gegenereerde gegevens betreffende levensstijl en levensverwachting. Aan jullie de taak om deze data te *scrapen*.

Aanvulling en duiding

De data beschrijft de levensstijl van volwassenen. We maken hierin geen onderscheid in man/vrouw/...

- **Genetic** is de genetische leeftijd in jaren (zonder invloed van levensstijl)
- **Length** is de lengte in centimeters
- **Mass** is de massa in kilogram
- **Exercise** is de hoeveelheid beweging in uren per dag
- **Smoking** beschrijft het aantal sigaretten per dag
- **Alcohol** beschrijft het aantal glazen alcoholische drank per dag
- **Sugar** beschrijft de hoeveelheid suiker in klontjes per dag
- **Lifespan** is de leeftijd afhankelijk van invloed levensstijl

We gaan ervan uit dat de genetische leeftijd bepaald kan worden met een genetische test. Hierin wordt het DNA van een patiënt geanalyseerd op aanleg en defecten. Dit is verder een gegeven.

Data transformatie

De ruwe data van de website is op zichzelf onbruikbaar: de data bevat fouten en hiaten. Aan jullie de taak om dit om te zetten naar een bruikbare set. Bedenk ook hoe deze stap past in je data pipeline en hoe je dit kunt automatiseren.

Data interpretatie

Exploratie

Kies op basis van de data één of meerdere klassieke machine learning technieken om de data te interpreteren. Neem de tijd om de data te verkennen. Maak hierbij gebruik van de tools die beschikbaar zijn. Denk hierbij aan ScikitLearn, Numpy, Matplotlib en Pandas.

Modelleren

Bouw met deze tools een model om de levensverwachting te voorspellen op basis van *genetics*, *length*, *mass*, *exercise*, *smoking*, *alcohol* en *sugar*. Dit model kan de arts gebruiken om patiënten te wijzen op het belang van een gezonde levensstijl. Ook kan de arts inzichtelijk maken hoe de levensstijl van invloed is op de verzekeringspremie.

De factor voor de premie wordt berekend via de formule:

$$PremieFactor = \frac{genetic}{lifespan}$$

Dus als de genetische leeftijd 75 jaar is en de levensverwachting 80 jaar, dan is de premiefactor gelijk aan 0,9375. Dit betekent een korting van 6,25% op de premie.

Houd ook in deze stap rekening dat er nieuwe data beschikbaar kan komen.

Datapresentatie

De arts kan via een console applicatie de benodigde gegevens invoeren, en krijgt dan de resultaten te zien. Om de patiënt inzicht te geven in een verandering van levensstijl kan de arts de stijl aanpassen, bijvoorbeeld het aantal sigaretten per dag aanpassen. De patiënt krijgt dan te zien hoeveel invloed dit heeft op de levensverwachting. Na een sessie kan de arts via de escape toets terug naar het startpunt. Zorg ervoor dat alle stappen logisch verlopen. Laat je applicatie daarom testen door je mede-cursisten en door de docenten.

Deliverables

- Vastlegging van inzichten en experimenten in één of meerdere Jupyter Notebooks
- Korte onderbouwing van gemaakte keuzes (ontwerp, implementatie)
- Overzicht van de testresultaten (nauwkeurigheid, foutmarges) en advies (bruikbaarheid?)
- Uitdraai van de git historie (geeft een beeld van het software development proces)
- De uiteindelijke broncode moet als zip-bestand worden ge-upload naar je eigen Teams kanaal. Maak duidelijk onderscheid tussen de software modules die je bij een klant kunt deployen (run) en de modules die je zelf gebruikt om een model te trainen (build).

Upload bovenstaande onderdelen **vóór de deadline van maandag 20 maart, 17:00 uur** naar:

Teams > Applied Artificial Intelligence – Make IT Work >
Jouw Private Channel > Files > Deliverables periode 1 > Project 1

Zorg daarbij voor de volgende mappenstructuur:

- Notebooks
- Onderbouwing
- Advies
- Git-historie
- Broncode

Projectevaluatie

Het doel van de projectevaluatie is tweeledig. Aan de ene kant wordt in dit document aandacht besteed aan de meta-functionele overwegingen die gemaakt worden tijdens het project, daarnaast wordt in dit document aandacht besteed aan de individuele ervaringen gedurende dit project. Meta functionele eisen: het gaat hier om bedrijfsmatige, ethisch-maatschappelijke, juridische, regulerings- en technische kaders die de directe functionele eisen overstijgen. Bij toetsing ligt de nadruk op de volgende onderwerpen c.q. activiteiten:

IT voor AI

- 1) Online beschikbare data verzamelen
- 2) Data transformeren en interpreteren en indien nodig verrijken
- 3) Gebruik van Python libraries (Numpy, Matplotlib, Pandas, enz.)
- 4) Het opsplitsen van een applicatie in modules
- 5) Het verwerken van de verzamelde data in een database zodat je ze met query's kunt filteren en combineren
- 6) Zet alle `imports` bovenaan met daaronder alle globale variabelen
Gebruik `snake_case` voor bestandsnamen en directories
Gebruik `./relatieve/linux/paden`

Toegepaste wiskunde

- 1) Lineaire en niet-lineaire regressie
- 2) Decision Tree & Random Forest
- 3) K-means / -nearest clusering

Professionele vaardigheden

- 1) Kennis putten uit onvolmaakte bronnen door kritisch te lezen en meerdere bronnen te combineren
- 2) Planmatig en zelfstandig werken
- 3) Het vinden van een werkbare balans tussen snel versus beheersbaar ontwikkelen bij een proof of concept.

Toetsing

De toetsing is formatief. Het doel is niet beoordeling, maar het geven van feedback en adviezen door de docent(en). Hierbij ligt de nadruk op de volgende zaken:

- 1) Uitwerking van opdrachten die onderdeel zijn van de lessen
- 2) Uitwerking van de projectopdracht, inclusief inhoud onderzoeksrapportage (Jupiter Notebook), broncode, git historie, testresultaten en advies
- 3) Formatieve toetsing vindt plaats tijdens gesprekken in de les en indien nuttig tijdens een afrondend gesprek. Het gaat daarbij om het geven van feedback en adviezen door de docenten, niet om beoordeling.
- 4) Bijzondere aandacht wordt in dit project besteed aan de manier waarop je je broncode in onafhankelijk testbare onderdelen hebt verdeeld. Dit kunnen modules of classes zijn, of allebei.
- 5) Daarnaast komen juist in dit project meta-functionele aspecten aan de orde. Gezondheidszorg neemt hierbij een bijzondere plaats in. Ethische vragen rond privacy, betaalbaarheid, beschikking over eigen leven, maar ook gedifferentieerde verzekeringspremies zijn actueel.
- 6) Inventariseer voor jezelf hoe je tegen dit soort zaken aankijkt. Er is geen “voorgebakken” goed antwoord. Het is echter wel een maatschappelijke discussie waar je als ontwikkelaar mee te maken krijgt, actief of passief. Hoever vind je dat jouw verantwoordelijkheid daarin gaat? Vroeg of laat ben je behalve ontwikkelaar mogelijk ook consument. Hoe vind jij dat deze zaken geregeld moeten zijn en waarom?
- 7) Een laatste aspect is de regelgeving. Denk van tevoren eens na over testbaarheid en betrouwbaarheid. Zijn er zaken die je hier in een professionele omgeving anders wilt doen dan bij het maken van bijvoorbeeld een computer-game? Welke? Hoe zou dat in het ontwikkelproces tot uiting kunnen komen, o.a. wat betreft samenwerking met collega's?

Tips & Tricks

- Maak een persoonlijk Kanbanboard om daarin al je taken bij te houden. Hoe breek je de opdracht op in kleine taken? Geef de docenten toegang.
- Commit en push elke dag je code en maak handig gebruik van branches voor nieuwe experimenten zodat je altijd een stabiele versie hebt. Geef de docenten toegang tot jouw Github repository.
- Maak een ontwerp en gebruik daarbij de UML-diagrammen die we behandeld hebben. Tip: [VSCode Mermaid plugin](#).
- Gebruik een debugger en plaats tactische breakpoints zodat je de 'flow' van het programma kunt volgen.
- Gebruik logging en stem het loglevel af op de ontwikkelfase.
- Maak voor elke essentiële functie een unit test zodat je meteen een specificatie hebt en controleert of nieuwe code de huidige code niet breekt.
- Wanneer je met nieuwe libraries (andere dan in het installatiescript) gaat experimenteren, maak dan eerste een nieuwe virtual Python environment aan.
- Let op de *hardening* van je code; probeer de import van libraries zoveel mogelijk te beperken tot wat je echt gebruikt.
- Gebruik Mark Down voor het aanbrengen van structuur in Jupyter notebooks.

Bijlagen

Daily Kata Starter Questions

- 1) Wat is het einddoel? Wat heb en kun je aan het einde van het project?
- 2) Waar sta je nu?
- 3) Zijn er obstakels die je verhinderen om het doel te bereiken?
Zo ja, van welke ondervind je op dit moment hinder?
- 4) Gegeven het einddoel en waar je nu staat, wat is de volgende stap?
- 5) Hoe kun je de stap dusdanig klein maken dat je die vandaag in zijn geheel kunt nemen?

Zie <https://traccsolution.com/blog/toyota-kata> voor achtergrondinformatie.

Bronnen

[Trimbos - roken](#)

[Trimbos - alcohol](#)

[Nidi - obesitas](#)

[Voedingscentrum - BMI](#)