# Inleiding Project 3 MIW

Voor project 3 moeten we kortgezegd een app maken die voor medisch personeel en patienten inzicht biedt in de gezondheid en adviezen geeft aan de patienten voor verbetering van de gezondheid. Deze adviezen worden gemaakt op basis van data.

# Onderverdeling Werkzaamheden

## Repository opzetten

Voor een geordende ontwerpproces moet ook een geordende codebase opgezet worden. Er is al een repository op github aangemaakt MakeAIWork3. Daarbinnen wordt een aparte map aangemaakt voor de healthapp.

De repository van de helathapp wordt ingedeeld in de volgende onderdelen.

* Webscraping
* Machine learning/ statistiek: deze gegevens worden opgesteld in
* App interface.

De codebase wordt gestart met cookiecutter. Deze structuur is gericht op het ontwikkelen van een Machine Learning model. De code van de

Voor de code ontwikkeling wordt een aparte virtuele Python environment aangemaakt. Als basis gebruik Ik de environment die aangemaakt wordt met create\_virtual\_env.sh

## Webscraping

De gezondheidsdata waarop de voorspellingen gemaakt worden staan op een webpagina We hebben toegang tot deze webpagina opgegeven gekregen, de gegevens moeten nog wel van de pagina afgehaald worden.

1 ) Lezen van de gegevens van de webpagina.

2) Omzetten data voor verdere verwerking.

Er zijn twee opties:

* Experimenten met Selenium en Beautiful Soup. Selenium is een bibliotheek voor interactie via webbrowsers. Met Beautiful Soup kun je html code naar een formaat vertalen die geschikt is voor verdere verwerking.
* Alternatief voor Beautiful Soup en Selenium is Scrapy. Deze bibliotheek biedt functionaliteit die

## Opslag Data, Database

De gegevens van de website worden omgezet naar een hanteerbaar formaat. Voor de had ligt een pandas dataframe. Als extra tussenstap kunnen de gegevens omgezet worden naar een database.

We hebben met twee database systemen gewerkt:

* Sqlite
* MongoDb

We gaan voor één of beide systemen uitproberen. Voor beide systemen doen het volgende

* data omzetten van pandas dataframe naar database.
* Toevoegen van nieuwe data aan de database
* Queries maken om data

## Machine Learning/ Statistiek

De gegevens uit de database moeten uiteindelijk vertaald worden naar een voorspelling van de gezondheid van een patient en een advies hoe deze te verbeteren. Met een ML model kan dit gerealiseerd worden.

* Vertaling van gegevens naar samengestelde gegevens. Bijvoorbeeld uit de lengte
* Keuze model. Gezien de gegevens lijken er twee modellen in aanmerking te komen. Regressiemodel, K-means. Kunnen meerdere modellen gecombineerd worden?

Op basis van metrics wordt bepaald welk model het beste is.

## App interface gebruiker

De gegevens uit het model moeten vertaald worden voor de gebruikers. De volgende vragen moet de app beantwoorden:

* Wat wil de gebruiker

Wat is de

* Bevragen machine learning model.
* De app moet uiteindelijk gebruiker kan een app interface maken die
* Aantrekkelijke visualisatie gegevens.

Django is een aantrekkelijk framework.

## Programmeer proces, bron-code- unit testen

Uiteindelijk wordt een app gemaakt waarin alle onderdelen samen komen. De code wordt object georiënteerd opgesteld. De bron code wordt in Python opgesteld.

De app kan in een app worden opgesteld.

Onderdeel van een geordend programmeer proces is de unit test. Voor een nuttige inzet van unit testen is het volgende gebeurd:

* Identificeren essentiële onderdelen app.
* Wat moet er getest worden?
* Stel de essentiele onderdelen.

## App in Docker

In voorafgaande projecten maakten is de code ontwikkeld op een laptop. Op deze manier kan snel code ontwikkeld worden. Een belangrijk nadeel van deze aanpak is dat de code op een andere computer niet direct werkt. Een Docker omgeving kan een

* Bepalen instellingen Docker omgeving
* Installatie script. Met dit script worden
  + Pakketten, bibliotheken geinstalleerd,
  + Omgevingsvariabelen ingesteld/vastgesteld.