Voorspellen van een hartaandoening

Een machine-learning model ontwerpen om een hartaandoening te kunnen voorspellen

Mark van de Streek – BFV3

Inhoudsopgave

Inleiding

In Nederland zijn er ruim 1.7 miljoen mensen met een hartaandoening *(Hart- en vaatziekten | Leeftijd en geslacht, z.d.)*. Denk hierbij aan chronische aandoeningen, hartritmestoornissen, hartklepafwijkingen en nog veel meer.

Iedereen kent wel iemand met een hartaandoening of iemand die is overleden aan een hart-gerelateerde oorzaak. Deze aandoeningen kunnen behoorlijk uiteenlopen, de ene patiënt heeft ook veel meer klachten dan de andere patiënt. In dit onderzoek zal er worden gekeken naar het ontwikkelen van een model dat op basis van klinische kenmerken, een hartaandoening kan voorspellen bij een patiënt.

Het doel van dit onderzoek is om met zo min mogelijk klinische kenmerken een goed model te ontwikkelen, die snel en eenvoudig een hartaandoening kan voorspellen bij een patiënt. Het model wordt ontwikkeld op basis van een machine-learning model. De onderzoeksvraag luidt daarom: “Welke medische eigenschappen zijn belangrijk om zo nauwkeurig mogelijk een hartaandoening of afwijking te voorspellen?”.

Voor het onderzoek zal er een dataset (*Predicting heart disease using clinical variables*, 2023) worden gebruikt waarin patiënten aanwezig zijn die een hartaandoening hebben, maar ook mensen die *geen* aandoening hebben. De dataset zal allereerst gebruikt worden om een model te ontwikkelen. Vervolgens zal dezelfde dataset worden gebruikt om het model te trainen. Dat wil zeggen dat de patiënten die geen aandoening hebben, ook worden gebruikt om het model te testen.

Voordat het model ontwikkeld wordt, moet er gekeken worden naar de juistheid van de data. Bevat de data genoeg verschillen? Zijn er afwijkende patronen zichtbaar? Welke klinische eigenschappen hebben op het eerste oog een hoge invloed? Dit wordt gedaan in een exploratieve data-analyse.

Verder moet er natuurlijk eerst duidelijk zijn wat precies een hartaandoening is en wat de gerelateerde onderwerpen hieraan zijn.

Een hartaandoening kan het volgende omvatten (*Heart disease - symptoms and causes - Mayo Clinic*, 2022):

- Bloedvatziekte
- Onregelmatige hartslag
- Aangeboren hartafwijkingen
- Beschadigde hartspier
- Hardklepziekte

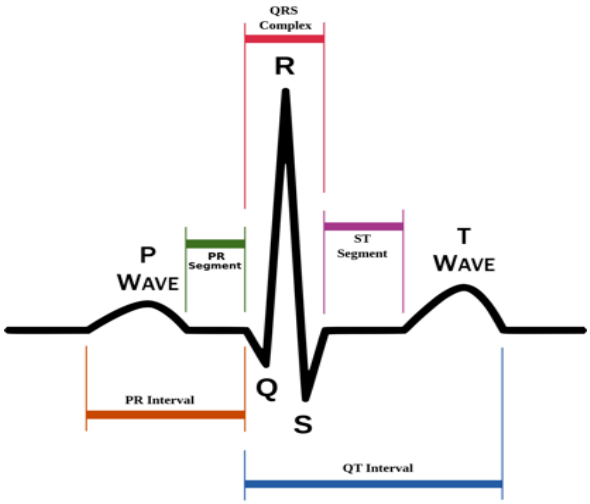
Al deze onderwerpen hebben onderling verschillende soorten ziektes. Al deze ziektes worden in dit onderzoek geclassificeerd als hartaandoening.

Materialen en Methoden

Korte beschrijving van de data

Er zijn vandaag de dag meerdere technieken/waarden waarmee afwijkingen van het hart in kaart kunnen worden gebracht.

Om te beginnen zijn de ECG-resultaten aanwezig in de dataset. Deze worden opgedeeld in drie sub-waarden (normaal, afwijking in specifieke sectie van de ECG en een waarschijnlijke of definitieve afwijking in de hoofdkamer van het hart).



Figuur 1 Diagram waarin een normaal sinusritme voor een menselijk hart wordt weergegeven

Hiernaast is een voorbeeld zichtbaar van een ECG. Een ECG bestaat uit verschillende segmenten. Het ST-segment kan een eventuele hartaandoening aangeven. De steilheid van dit segment bepaalt dit (oplopend, plat of dalend). In de dataset is een attribuut aanwezig met de steilheid (ST.depression).

Er kan ook gekeken worden hoe de bloedstroom rondom het hart is. Dit wordt gedaan met een techniek genaamd Thallium-scan. Thallium is een scheikundig element dat in je lichaam wordt gebracht als een tracer. Buiten het lichaam kan er vervolgens met een infraroodcamera worden gekeken waar deze thallium allemaal terecht komt. Hiermee kan dus worden gekeken in welke delen van het hart het bloed wellicht minder goed stroomt.

Verder is er een attribuut aanwezig die aangeeft of de bloedvaten die voedingsstoffen naar het hart brengt, vernauwd zijn (exercise angina).

Tot slot is het nog belangrijk om een attribuut te benoemen die het glucosegehalte weergeeft. Er is gekeken naar de het glucosegehalte in het bloed na acht uur niks te hebben gegeten en gedronken. Het attribuut heet FBS over 120. Mensen die na acht uur vasten een glucosegehalte hebben van boven de 120 mg/dl, worden gekenmerkt als “yes”.

Als een patiënt daadwerkelijk een glucose niveau heeft van boven de 120 mg/dl, betekent dit niet meteen dat deze persoon een hartaandoening heeft. Maar deze waarde zou daar wellicht iets over kunnen zeggen.

Materialen

Om de juistheid van de data te toetsen is er een exploratieve data-analyse uitgevoerd. Deze werd uitgevoerd met behulp van de programmeertaal R *(R Core Team, 2022 version 4.2.2)*. In R zijn ook meerdere pakketten gebruikt. Voor het ontwikkelen van het model is de datamining software van Weka *(versie 3.8.6)* gebruikt.
De volgende pakketten zijn gebruikt:

Tabel 1 Overzicht van gebruikte pakketten.

Software	Pakket	Versie
R	Pander	0.6.5
R	Ggplot2	3.4.1
R	Ggcorrplot	2.1.2
R	gridExtra	2.3
R	dplyr	1.1.0
R	Data.table	1.14.8
R	tidyverse	2.0.0
Weka	attributeSelectionSearchMethods	1.0.7

Methoden

In de exploratieve data-analyse (EDA) is er onder andere gekeken naar de verdeling van de numerieke attributen. Hier werd zichtbaar dat de attributen cholesterolgehalte en ST-segment afdaling numeriek beter richting normaal verdeeld waren, na een log10 transformatie. Zonder transformatie is er een hoge piek te zien in het aantal nullen van het ST-attribuut. Met een log10 transformatie werden deze waardes allemaal -oneindig.

Met de software van Weka is gekeken naar het beste model. Hiervoor zijn alle algoritmen in een zogenaamd experiment gezet. Op dit experiment zijn vervolgens testen uitgevoerd om naar de significantie te kijken. De nauwkeurigheid van de modellen is het meest globale om naar te kijken, echter geeft de nauwkeurigheid geen hoog inzicht om prestaties van concepten met elkaar te kunnen vergelijken. Onderstaand een korte uitleg naar verdere kwaliteitsmaatstaven die zijn gebruikt voor het vergelijken van de concepten.

Ten eerste is er gekeken naar de verwarringsmatrix, ook wel “confusion matrix” genoemd. Deze vorm geeft bijvoorbeeld een veel beter inzicht in de aantal fouten die een algoritme maakt.

*Tabel 2 Voorbeeld van een confusion matrix. "Geclassificeerd als" geeft aan wat de voorspelling is volgens het algoritme. "a = presence" en "b = Absence" geeft aan wat de voorspelling in werkelijk is. Bijvoorbeeld als een algoritme een patiënt **classificeert** als presence, maar deze in **werkelijkheid** absence is, spreek je van een valse positieve.*

a	b	<-- classified as
106	14	a = Presence
46	04	b = Absence

In deze matrix zijn vier elementen zichtbaar: TP: echte positieven (worden geclassificeerd als positief en zijn daadwerkelijk ook positief), FN: valse negatieven (worden geclassificeerd als negatief, maar zijn daadwerkelijk positief), FP: valse positieven (worden geclassificeerd als positief, maar zijn daadwerkelijk negatief) en TN: echte negatieven (worden geclassificeerd als negatief en zijn daadwerkelijk ook negatief).

Met deze waardes kunnen zogenaamde ratio's worden berekend. Bijvoorbeeld de TPR (True Positive Rate of Gevoeligheid). Dit percentage geeft de daadwerkelijke positieve resultaten dat nauwkeurig is geïdentificeerd.

Voor het maken van het model is het belangrijk dat de TPR zo hoog mogelijk is en de FPR zo laag mogelijk is.

Een andere kwaliteitsmaatstaf waarnaar is gekeken is de ROC-curve. Om precies te zijn het gebied onder de curve. In deze curve wordt de TPR geplot tegen de FPR. Dit geeft de prestatie van een concept weer. Hoe meer de kromming naar linksboven gaat, hoe beter het model. Het gebied onder de kromming wordt de "Area under ROC" genoemd.

Verder is er ook nog gekeken naar de F-score. Dit is een uitgebreidere versie van de nauwkeurigheid maatstaf. De F-score wordt berekend aan de hand van de 'true positive rate' en de 'positive predictive value'.

Aan de hand van deze kwaliteitsmaatstaven is er in een Weka experiment gekeken naar de algoritmen. Een Weka experiment is een omgeving waarin verschillende modellen met elkaar worden vergeleken. Uit dit experiment zijn de beste modellen naar voren gekomen.

Nadat de beste algoritmen zijn verkregen (uit het experiment), is er als laatste nog gekeken naar zogenaamde 'meta-learners'. Meta-learning is een bandering van machine-learning waarbij algoritmen worden gebruikt om betere voorspellingen te doen door gebruik te maken van de uitgangen en metadata van andere machine-learning algoritmen. Met andere machine-learning modellen worden onze eerder (best) verkregen algoritmen bedoeld. Er is gekeken naar de drie meest voorkomende meta-learners. De resultaten van deze meta-learners worden in de resultaten-sectie bekeken. Bij één van deze learner is het mogelijk om een meta classifier te kiezen, dit is een classifier die de uiteindelijke beslissing maakt. Meerdere modellen zijn hiervoor getest.

Tenslotte is er met Weka ook gekeken naar de beste attributen. In de optie attribuut selectie wordt er een subset van de beste attributen berekent. Deze attributen zijn het belangrijkste voor het model (de hoogste scores worden behaald met deze subset van attributen).

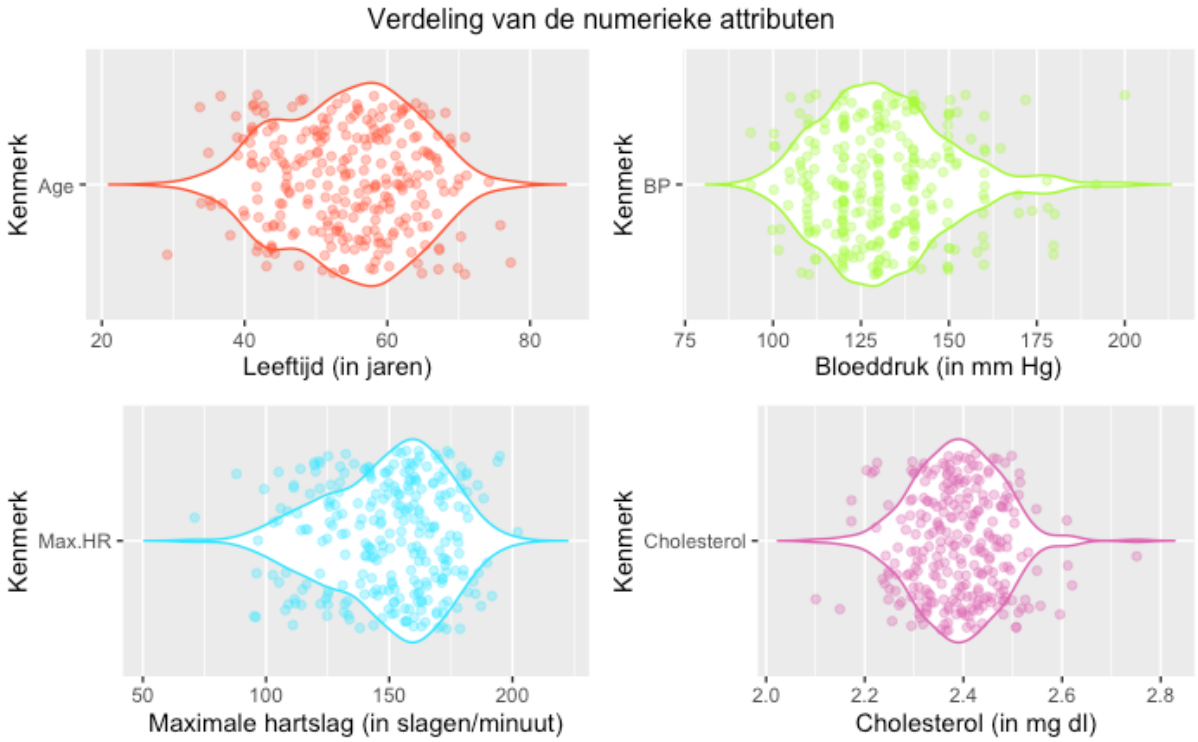
Het beste model is niet alleen maar gemaakt, maar ook gepubliceerd. Dit is gedaan in de vorm van een Java-applicatie. Via deze applicatie is het mogelijk om zelf een aantal patiënten op te geven en de applicatie een voorspelling te laten maken. Een uitgebreide beschrijving van het project is te vinden in de referenties *(MarkStreek, z.d.-a)*. Tevens is hier ook een installatiehandleiding en gebruiksaanwijzing te vinden.

Resultaten

Uit de EDA zijn een aantal interessante bevindingen aan het licht gekomen. Om een zo respectief mogelijk antwoord te kunnen geven op de onderzoeksvraag is het van belang om even veel klasse-aantallen te hebben. Je wil dus evenveel mensen met als zonder een hartaandoening. Er zijn voor dit onderzoek 15 (medische) kenmerken van 270 patiënten gebruikt. In figuur 2 is zichtbaar dat de groep met een hartaandoening bestaat uit 120 mensen. De groep zonder een aandoening bestaat uit 150 mensen.

Figuur 2 Cirkeldiagram van de verdeling van de klasse labels. De groep die een aandoening heeft is in het rood afgebeeld. De groep zonder een aandoening in het blauw.

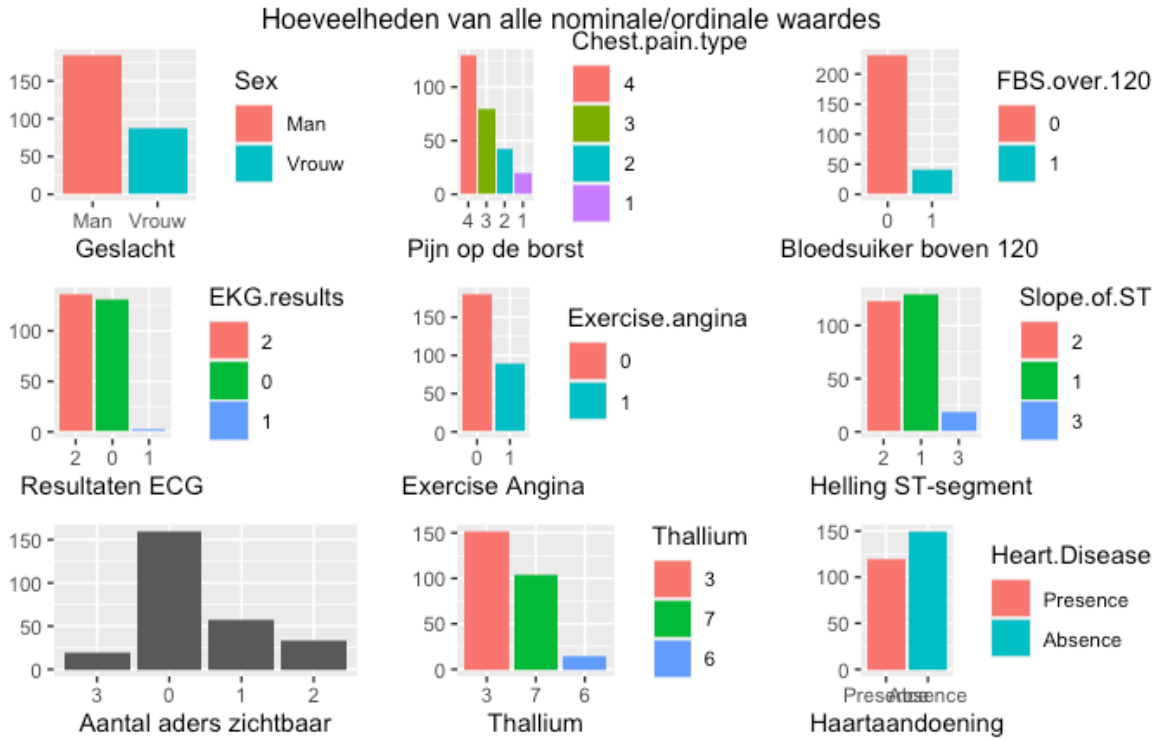
Om de verdeling van de (numerieke) attributen te tonen is er een Violin-plot geconstrueerd. In dit plot zijn de punten geplot om uitschietende waardes te lokaliseren.



Figuur 3 Vier Violin plots die de verdeling van de kenmerken weergeven in combinatie met de buiten liggende waardes. Van links naar rechts onder elkaar afgebeeld: leeftijd (rood), bloeddruk (lichtgroen), maximale hartslag (blauw) en cholesterol (paars). Op de x-assen staan de attribuut-waardes.

In figuur 3 (bovenstaand) valt op dat de kenmerken bloeddruk en maximale hartslag de meeste aantal uitschietende waardes hebben. De verdeling van de bloeddruk heeft een piek rond de 125 mm Hg. Voor de maximale hartslag ligt de piek van de verdeling net iets boven de 150 slagen per minuut.

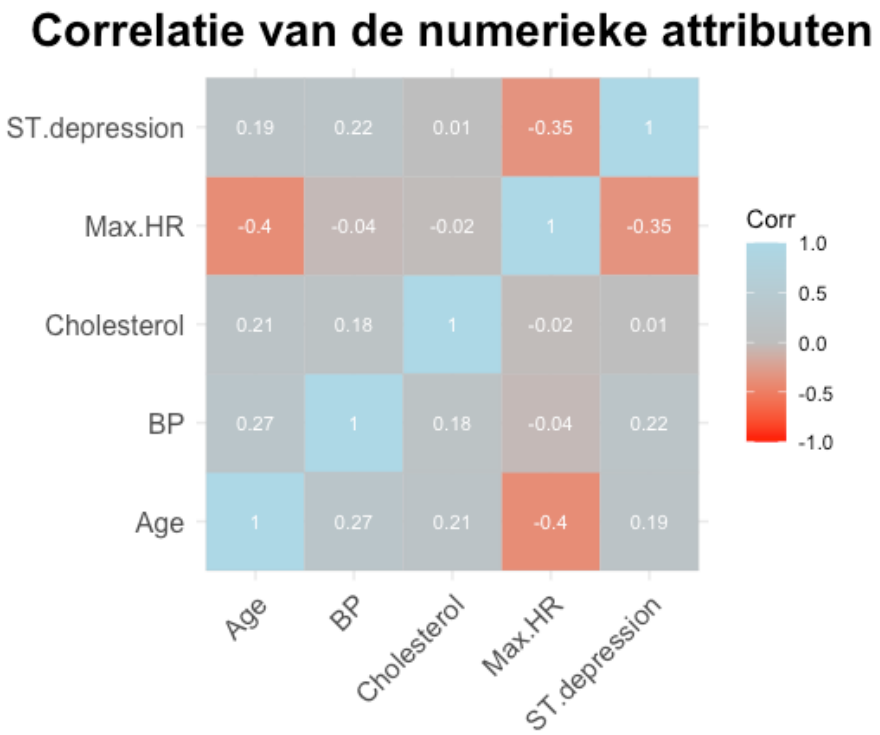
Uiteraard is er ook gekeken naar de nominale attributen. De hoeveelheden van de kenmerken is zichtbaar in figuur 4.



Figuur 4 Negen barplots van alle nominale/ordinale attributen. Van links naar rechts en van boven naar beneden zijn de volgende attributen weergegeven: geslacht, pijn op de borst, bloedsuiker boven de 120, resultaten ECG, Exercise angina, Helling ST-segment, Aantal aders zichtbaar, Thallium en Hartaandoening. De categorieën van de ordinale waardes staan op alle x-assen en de hoeveelheden van deze waarde op de y-as. De legenda's zijn aanwezig naast de figuren.

In figuur 4 is te zien dat er meer mannen aanwezig zijn dan vrouwen. De resultaten lijken meer naar mensen zonder een aandoening te "leunen". Dit blijkt uit bijvoorbeeld pijn op de borst, bloedsuikerspiegel, helling ST-segment en Thallium.

Voor een ontwikkelen van een goed model is er ook gekeken naar samenhang tussen verschillende attributen. In figuur 5 is een warmtebeeld zichtbaar. Hierin is te zien dat er geen grote waardes zijn, dat wil zeggen, er zijn geen attributen die een heel hoge correlatie hebben.



Figuur 5 Warmtebeeldplot van de numerieke waardes. Alle attributen staan op zowel de x- als y-as. Het cijfer geeft de correlatie (samenhang) tussen de attributen weer. Bij een correlatie van 1 is er een (zeer) sterke samenhang tussen deze attributen (blauw). Bij een correlatie van -1 is er (vrijwel) geen samenhang tussen de attributen (rood).

Verder valt er in figuur 5 te zien dat de vergelijking tussen maximale hartslag (Max.HR) en helling in ST-segment (ST.depression) de laagste correlatie heeft. Dit wordt gevolgd door Max.HR en leeftijd. Tussen leeftijd en bloeddruk is de hoogste correlatie aanwezig.

Tot slot is er in de exploratieve data-analyse gekeken naar de samenhang tussen de attributen en het wel/niet aandoening attribuut. Hiervoor zijn statistische toetsen uitgevoerd. Uit elke toets kwam een p-waarde. Hoe lager de p-waarde, hoe hoger de samenhang tussen deze attributen. Onderstaand een gesorteerde tabel van alle p-waardes.

Tabel 3 Overzicht van alle p-waarden. De p-waarden zijn verkregen door statistische toetsen uit te voeren tussen elke attribuut en het wel/niet ziek attribuut. De waardes zijn gesorteerd van klein naar groot.

Attribuut	**P-waarde**
Thallium	6.419×10^{-17}
Pijn op borst	8.561×10^{-15}
Aantal aders	1.437×10^{-13}
Maximale hartslag	2.604×10^{-12}
Exercise Angina	5.585×10^{-12}
ST-segment ECG	1.601×10^{-11}
ST-segment ECG numeriek	1.713×10^{-9}
Sex	9.979×10^{-7}
Leeftijd	3.526×10^{-4}
ECG resultaten	1.122×10^{-2}
Bloeddruk	1.196×10^{-2}
Cholesterol	4.971×10^{-2}
Bloedsuiker over 120	7.886×10^{-1}

In tabel 2 is zichtbaar dat het attribuut Thallium de laagste p-waarde heeft. Dit betekent dat dit attribuut de meeste samenhang heeft met de wel/niet ziek groep. Van de 13 attributen zijn er 12 die een p-waarde lager dan 0.05 hebben. Bij een p-waarde lager dan 0.05, wordt er gesproken van een significante samenhang.

Nadat alle inzichten zijn verkregen over de data, is vervolgens het model gecreëerd. Zoals in de materialen & methoden beschreven, is het beste algoritme verkregen door te kijken naar de beschreven maatstaven. Zie bijlage A voor de uitkomsten/maatstaven van het Weka experiment.

In het project (MarkStreek, z.d.-a) is een logboek te vinden waar alle resultaten van het experiment uitgebreid worden gepresenteerd. Tevens wordt hierin ook vermeld hoe het experiment is opgesteld.

Uit het experiment kwam 'Logistic' en 'NaiveBayes' als hoogste naar voren. Deze algoritmen zijn vervolgens gebruikt om de meta-learners te testen.

Tabel 4 Overzicht van de Meta learners. Alle geteste meta learners zijn onder elkaar gezet en de nauwkeurigheid en het gebied onder de ROC-curve is gegeven voor elke test.

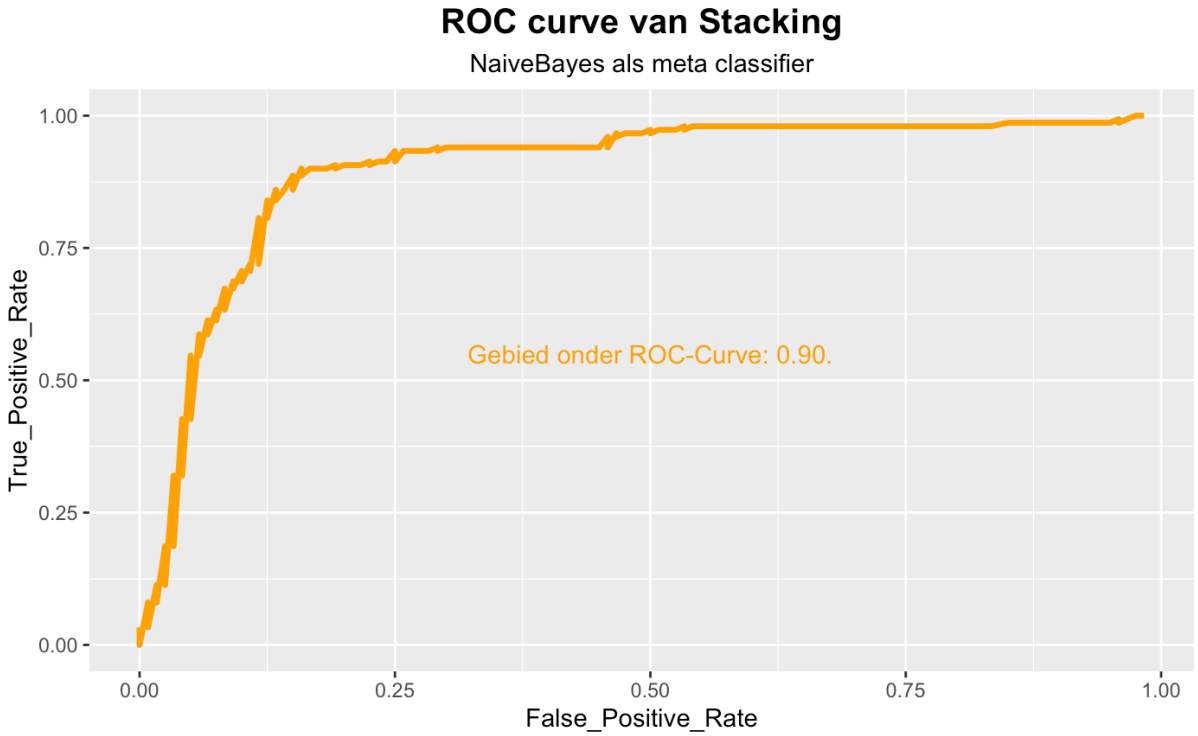
Meta-learner	**Nauwkeurigheid**	**Gebied onder ROC-curve**
Voting	86.2963%	0.90
Stacking met NaiveBayes als Meta classifier	86.6667%	0.90
Stacking met Logistic als Meta classifier	87.037%	0.90
Bagging met Logistic	85.1852%	0.90
Bagging met NaiveBayes	83.7037%	0.90

In tabel 3 zijn de resultaten van de meta-learners zichtbaar. Stacking met NaiveBayes als meta classifier behaald de hoogste score. Het gebied onder de ROC-curve is voor alle resultaten gelijk.

Als laatste de beste attributen. In weka zijn zoals vermeld de beste attributen berekend. Uit deze berekening komt een subset van attributen. Deze attributen zijn nodig om de hoogste score te behalen.

De volgende subset kwam naar voren: Leeftijd, geslacht, Pijn op borst, bloeddruk, bloedsuikerspiegel over 120, maximale hartslag, exercise angina, helling ST-segment, helling ST-segment numeriek, aantal vaten zichtbaar en thallium.

Van het definitieve model is vervolgens de ROC-curve geplot, deze is zichtbaar in figuur 6.



Figuur 6 ROC-curve van het definitieve Model: stacking met Naivebayes als metaclassifier. Op de x-as staat de false positive rate en op de y-as de true positive rate. Een gebied onder de ROC-curve van 1.0 is maximaal, dan topt de grafiek helemaal uit naar linksboven. Het gebied onder de ROC-curve is ook in het figuur weergegeven (als tekst).

Zoals bovenstaand zichtbaar, heeft deze ROC-curve heeft een piek naar linksboven.

Discussie

In figuur 1 is de verdeling van de klasse labels zichtbaar. Hierin valt te zien dat de groep met een hartaandoening een grotere vertegenwoordiging heeft dan de groep zonder een hartaandoening. Uiteraard was het voor dit onderzoek het beste geweest om een dataset te hebben waarbij de verdeling omme en nabij de 50% zou zijn geweest. Echter is de groep niet velen malen groter en lijkt dit prima te zijn voor dit onderzoek.

In figuur 3, kijkend naar de verdeling van de numerieke attributen, valt op dat het attribuut met de maximale hartslag het meeste aantal uitschieters heeft. Er is geen logische verklaring hiervoor. Neem bijvoorbeeld de verdeling, deze laat geen extreem afwijkend patroon zien.

In figuur 4 werd opgemerkt dat de meest waarden “leunen” richting gezonde mensen. Dit zou kunnen komen doordat (1) de dataset meer gezonde mensen bevat of (2) mensen met een aandoening zijn lastig te onderscheiden door alleen te kijken naar ordinale waardes.

In tabel 4 zijn er een aantal zaken die opvallen. Om te beginnen het gebied onder de ROC-curve. Deze gebieden zijn voor alle meta-learners gelijk. Dit wijst erop dat alle learners even accuraat zijn. Verder valt op de nauwkeurigheid behoorlijk uiteenloopt. Er is 3.3% verschil tussen het hoogste en laagste percentage.

Tot slot is er nog een opvallende zaak in de attribuut selectie. Van de 14 attributen zijn er 11 nodig om de beste voorspelling te kunnen maken. Dit zijn (nagenoeg) alle belangrijke attributen in de dataset. Het is natuurlijk voor de hand liggend dat voor het verzamelen van deze dataset al de “juiste” attributen zijn verzameld door de onderzoekers, de benodigde attributen om een goede voorspelling te kunnen maken zijn allemaal aanwezig in de dataset.

Conclusie

Het doel van dit onderzoek was om met zo min mogelijk klinische kenmerken een goed model te ontwikkelen. De onderzoeksvraag luidde: “Welke medische eigenschappen zijn belangrijk om zo nauwkeurig mogelijk een hartaandoening of afwijking te voorspellen?”.

De EDA toonde aan dat de dataset weinig aanpassingen nodig had voor het ontwerpen van een goed model. Met de helling van het ST-segment als uitzondering, lieten alle numerieke attributen een goede verdeling zien. De categorieën van de ordinale waardes werden goed gerepresenteerd en meer dan 80% van de attributen lieten een significante samenhang zien met het classificatie attribuut.

Kijkend naar een model waarbij er zomin mogelijk medische eigenschappen nodig zijn, is Stacking met Logistic en NaïveBayes (met NaïveBayes als metalearner) het beste model om een hartaandoening te kunnen voorspellen. De volgende medische eigenschappen worden hierin meegenomen: leeftijd, geslacht, pijn op borst, bloeddruk, bloedsuikerspiegel over 120, maximale hartslag, exercise angina, helling ST-segment, helling ST-segment numeriek, aantal vaten zichtbaar en thallium.

Uiteraard zijn er een aantal zaken die verbeterd kunnen worden bij een vervolgonderzoek. Om te beginnen had de dataset wellicht beter gekund. 270 patiënten is niet veel voor deze dataset. Zeker omdat de data afkomstig is van 4 verschillende bronnen. Verder hadden er nog attributen kunnen worden toegevoegd. Denk bijvoorbeeld aan de lichaamslengte of gewicht van de patiënt. Hiermee kan de BMI ook bepaald worden.

Een verbeterpunt voor het machine-learning gedeelte zou kunnen zijn dat er meer gebruik kan worden gemaakt van de parameters. Er zijn weliswaar verschillende opties getest voor algoritmen, maar dit kan altijd nog uitgebreider. Dit is ook zeker het geval voor bijvoorbeeld Logistic.

Verder is er weinig verteld over de Java publicatie van het model. Hier zou wellicht meer over kunnen worden verteld. Maar dit is uiteraard afhankelijk van het doel wat er mee bereikt wil worden.

Referenties

Atrial Fibrillation: Resources for Patients. (2020, 27 augustus). *Understanding the EKG Signal - Atrial Fibrillation: Resources for patients*. https://a-fib.com/treatments-for-atrial-fibrillation/diagnostic-tests-2/the-ekg-signal/

Hart- en vaatziekten | Leeftijd en geslacht. (z.d.). Volksgezondheid en Zorg. https://www.vzinfo.nl/onderwerpen/hart-en-vaatziekten/leeftijd-en-geslacht#:~:text=In%202021%20waren%20er%20naar%20schatting%201.704.100%20mensen%20met,%2C0%20per%201.000%20vrouwen).

Heart disease - symptoms and causes - Mayo Clinic. (2022, 25 augustus). Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118

MarkStreek. (z.d.-a). *GitHub - MarkStreek/Heart-Disease-ML: This repo contains a project where a machine learning model is created to predict a heart disease. The predicting is based on clinical variabeles*. GitHub. https://github.com/MarkStreek/Heart-disease-ML

MarkStreek. (z.d.). *GitHub - MarkStreek/PredictingHeartDiseaseJavaWrapper: This repository contains a Java wrapper where a heart disease can be predicted, using clinical attributes. See the README for further information about this project.* GitHub. https://github.com/MarkStreek/PredictingHeartDiseaseJavaWrapper

Predicting heart disease using clinical variables. (2023, 12 januari). Kaggle. https://www.kaggle.com/datasets/thedevastator/predicting-heart-disease-risk-using-clinical-var

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical

Computing, Vienna, Austria. URL https://www.R-project.org/.

WEKA 3 - Data mining with open source machine learning software in Java. (z.d.). https://www.cs.waikato.ac.nz/ml/weka/

Bijlage A

Tabel 1: Nauwkeurigheid van de algoritmen

```
# Dataset                    (1) 01-Clean | (2) 02-to (3) 03-Nu (4) 04-Si
# ----------------------------------------------------------------------
# rules.ZeroR '' 4805554146(100)   55.56 |   55.56     55.56     55.56
# rules.OneR '-B 6' -345942(100)   71.63 |   73.07     71.63     51.41 *
# rules.OneR '-B 15' -34594(100)   71.63 |   73.07     71.63     60.81 *
# trees.J48 '-C 0.25 -M 2' (100)   76.41 |   80.96 v   76.44     65.44 *
# trees.J48 '-U -M 2' -2177(100)   75.81 |   81.07 v   75.93     65.22 *
# trees.J48 '-C 0.25 -M 15'(100)   73.81 |   73.00     73.81     63.07 *
# trees.J48 '-C 0.25 -M 35'(100)   72.44 |   73.00     72.44     64.63 *
# trees.J48 '-C 0.025 -M 10(100)   73.89 |   76.52     73.89     63.22 *
# trees.RandomForest '-P 10(100)   81.33 |   79.33     82.22     64.96 *
# trees.RandomTree '-K 0 -M(100)   73.00 |   74.59     73.89     62.48 *
# lazy.IBk '-K 10 -W 0 -A \(100)   80.85 |   80.89     80.93     64.22 *
# lazy.IBk '-K 35 -W 0 -A \(100)   83.67 |   83.37     83.30     67.26 *
# lazy.IBk '-K 35 -W 0 -A \(100)   83.44 |   83.19     83.41     67.04 *
# functions.SMO '-C 1.0 -L (100)   84.07 |   83.70     84.04     62.67 *
# functions.Logistic '-R 1.(100)   83.96 |   83.81     83.85     66.04 *
# bayes.NaiveBayes '' 59952(100)   84.89 |   83.44     84.15     68.26 *
# meta.ClassificationViaClu(100)   78.07 |   80.81     78.15     62.11 *
# ----------------------------------------------------------------------
#                                (v/ /*) |  (2/15/0)  (0/17/0)  (0/1/16)
```

Tabel 2: Gebied onder de ROC-curve

```
# Dataset                    (1) 01-Clea | (2) 02-t (3) 03-N (4) 04-S
# ----------------------------------------------------------------
# rules.ZeroR '' 4805554146(100)   0.50 |   0.50     0.50     0.50
# rules.OneR '-B 6' -345942(100)   0.71 |   0.73     0.71     0.50 *
# rules.OneR '-B 15' -34594(100)   0.71 |   0.73     0.71     0.60 *
# trees.J48 '-C 0.25 -M 2' (100)   0.76 |   0.83 v   0.76     0.70
# trees.J48 '-U -M 2' -2177(100)   0.76 |   0.82     0.76     0.70
# trees.J48 '-C 0.25 -M 15'(100)   0.78 |   0.78     0.78     0.68 *
# trees.J48 '-C 0.25 -M 35'(100)   0.74 |   0.74     0.74     0.68
# trees.J48 '-C 0.025 -M 10(100)   0.76 |   0.79     0.76     0.66 *
# trees.RandomForest '-P 10(100)   0.89 |   0.84 *   0.89     0.71 *
# trees.RandomTree '-K 0 -M(100)   0.73 |   0.75     0.74     0.62 *
# lazy.IBk '-K 10 -W 0 -A \(100)   0.89 |   0.88     0.90     0.71 *
# lazy.IBk '-K 35 -W 0 -A \(100)   0.91 |   0.89     0.90     0.73 *
# lazy.IBk '-K 35 -W 0 -A \(100)   0.90 |   0.89     0.90     0.74 *
# functions.SMO '-C 1.0 -L (100)   0.83 |   0.83     0.83     0.63 *
# functions.Logistic '-R 1.(100)   0.90 |   0.90     0.90     0.72 *
# bayes.NaiveBayes '' 59952(100)   0.91 |   0.90     0.91     0.72 *
# meta.ClassificationViaClu(100)   0.78 |   0.80     0.78     0.64 *
# ----------------------------------------------------------------
#                                (v/ /*) | (1/15/1) (0/17/0) (0/4/13)
```

Tabel 3: Vals negatieven

```
# Dataset                    (1) 01-Clea | (2) 02-t (3) 03-N (4) 04-S
# ----------------------------------------------------------------
# rules.ZeroR '' 4805554146(100)   1.00 |   1.00     1.00     1.00
# rules.OneR '-B 6' -345942(100)   0.32 |   0.30     0.32     0.60 v
# rules.OneR '-B 15' -34594(100)   0.32 |   0.30     0.32     0.47 v
```

```
# trees.J48 '-C 0.25 -M 2' (100)   0.30 |   0.25     0.30     0.46 v
# trees.J48 '-U -M 2' -2177(100)   0.29 |   0.24     0.28     0.47 v
# trees.J48 '-C 0.25 -M 15'(100)   0.30 |   0.34     0.30     0.50 v
# trees.J48 '-C 0.25 -M 35'(100)   0.29 |   0.26     0.29     0.52 v
# trees.J48 '-C 0.025 -M 10(100)   0.26 |   0.24     0.26     0.50 v
# trees.RandomForest '-P 10(100)   0.25 |   0.27     0.23     0.39 v
# trees.RandomTree '-K 0 -M(100)   0.33 |   0.31     0.30     0.41
# lazy.IBk '-K 10 -W 0 -A \(100)   0.20 |   0.21     0.20     0.33 v
# lazy.IBk '-K 35 -W 0 -A \(100)   0.22 |   0.25     0.22     0.43 v
# lazy.IBk '-K 35 -W 0 -A \(100)   0.23 |   0.26     0.23     0.44 v
# functions.SMO '-C 1.0 -L (100)   0.23 |   0.23     0.23     0.30
# functions.Logistic '-R 1.(100)   0.22 |   0.23     0.22     0.38 v
# bayes.NaiveBayes '' 59952(100)   0.19 |   0.22     0.21     0.38 v
# meta.ClassificationViaClu(100)   0.24 |   0.26     0.23     0.24
# ------------------------------------------------------------------
#                                (v/ /*) | (0/17/0) (0/17/0) (13/4/0)
```

Tabel 4: Vals positieven

```
# Dataset                   (1) 01-Clea | (2) 02-t (3) 03-N (4) 04-S
# ------------------------------------------------------------------
# rules.ZeroR '' 4805554146(100)   0.00 |   0.00     0.00     0.00
# rules.OneR '-B 6' -345942(100)   0.26 |   0.24     0.26     0.40 v
# rules.OneR '-B 15' -34594(100)   0.26 |   0.24     0.26     0.33
# trees.J48 '-C 0.25 -M 2' (100)   0.18 |   0.14     0.18     0.25
# trees.J48 '-U -M 2' -2177(100)   0.21 |   0.15     0.21     0.25
# trees.J48 '-C 0.25 -M 15'(100)   0.23 |   0.21     0.23     0.26
# trees.J48 '-C 0.25 -M 35'(100)   0.26 |   0.28     0.26     0.22
# trees.J48 '-C 0.025 -M 10(100)   0.26 |   0.23     0.26     0.26
# trees.RandomForest '-P 10(100)   0.14 |   0.16     0.13     0.32 v
# trees.RandomTree '-K 0 -M(100)   0.22 |   0.21     0.23     0.35 v
# lazy.IBk '-K 10 -W 0 -A \(100)   0.18 |   0.17     0.18     0.38 v
# lazy.IBk '-K 35 -W 0 -A \(100)   0.12 |   0.10     0.12     0.24 v
# lazy.IBk '-K 35 -W 0 -A \(100)   0.11 |   0.10     0.11     0.24 v
# functions.SMO '-C 1.0 -L (100)   0.11 |   0.11     0.10     0.43 v
# functions.Logistic '-R 1.(100)   0.11 |   0.11     0.11     0.30 v
# bayes.NaiveBayes '' 59952(100)   0.12 |   0.12     0.12     0.27 v
# meta.ClassificationViaClu(100)   0.21 |   0.14     0.21     0.49 v
# ------------------------------------------------------------------
#                                (v/ /*) | (0/17/0) (0/17/0) (10/7/0)
```

Tabel 5: F-score

```
# Dataset                   (1) 01-Clea | (2) 02-t (3) 03-N (4) 04-S
# ------------------------------------------------------------------
# rules.ZeroR '' 4805554146  (0)        |
# rules.OneR '-B 6' -345942(100)   0.68 |   0.69     0.68     0.42 *
# rules.OneR '-B 15' -34594(100)   0.68 |   0.69     0.68     0.53 *
# trees.J48 '-C 0.25 -M 2' (100)   0.72 |   0.78 v   0.72     0.57 *
# trees.J48 '-U -M 2' -2177(100)   0.72 |   0.78     0.72     0.57 *
# trees.J48 '-C 0.25 -M 15'(100)   0.70 |   0.68     0.70     0.53 *
# trees.J48 '-C 0.25 -M 35'(100)   0.69 |   0.71     0.69     0.54 *
# trees.J48 '-C 0.025 -M 10(100)   0.71 |   0.74     0.71     0.54 *
# trees.RandomForest '-P 10(100)   0.78 |   0.76     0.79     0.60 *
# trees.RandomTree '-K 0 -M(100)   0.69 |   0.71     0.70     0.58 *
# lazy.IBk '-K 10 -W 0 -A \(100)   0.79 |   0.78     0.79     0.62 *
# lazy.IBk '-K 35 -W 0 -A \(100)   0.81 |   0.80     0.80     0.60 *
# lazy.IBk '-K 35 -W 0 -A \(100)   0.80 |   0.79     0.80     0.60 *
# functions.SMO '-C 1.0 -L (100)   0.81 |   0.81     0.81     0.63 *
```

```
# functions.Logistic '-R 1.(100)   0.81 |   0.81     0.81     0.61 *
# bayes.NaiveBayes '' 59952(100)   0.83 |   0.81     0.82     0.63 *
# meta.ClassificationViaClu(100)   0.75 |   0.77     0.75     0.64 *
# ------------------------------------------------------------------
#                              (v/ /*) | (1/15/0) (0/16/0) (0/0/16)
```