Universiteit Leiden
Opleiding Informatica

Datamining in data van de Intensive Care

Wel of geen bloed transfusie?

Naam:	Mark Rasenberg
Studentnr:	1368281
Datum:	01/08/2016
1st supervisor:	Dr. S.G.R. Nijssen
2nd supervisor:	Prof.dr. A. Plaat

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Datamining in data van de Intensive Care: Wel of geen bloedtransfusie?

Mark Rasenberg

Abstract

Op de Intensive Care overlijden steeds minder vaak mensen door de technologische verbeteringen. Hierbij wordt echter nog geen gebruik gemaakt van datamining technieken. Door middel van datamining technieken hopen we een betere voorspelling te doen of een patiënt een bloedtransfusie nodig heeft of niet. Eerst hebben we er gekeken naar onderzoeken van andere mensen, om ideeën op te doen voor dit onderzoek. Vervolgens is er aan de hand van het testen van verschillende datamining algoritmen en diverse technieken om met missing values om te gaan, dat het RandomForest-algoritme als beste naar voren komt bij deze toepassing. Dat dit algoritme als beste naar voren is gekomen is gebaseerd op accuracy, precision en recall scores. Met een maximale precision van 95,23% bij een recall van 100% is dit algoritme op deze dataset een goede voorspeller. Echter is dit algoritme op het gebied van interpretatie en visualisatie ondermaats. Deze reden hiervoor is dat het algoritme een model bouwt op basis van meerdere decision trees met elk een grote hoeveelheid nodes.

Dankwoord

Graag wil ik Dr. S.G.R. Nijssen van de Universiteit van Leiden bedanken voor de assistentie, toezicht en opmerkingen tijdens het semester. Dit heeft deze scriptie naar een hoger niveau getild. Ook wil ik Dr. S.G.R. Nijssen bedanken voor de feedback op mijn concept scriptie. Zonder zijn supervisie en feedback had ik mijn scriptie niet op deze manier af kunnen ronden. Verder wil ik graag mijn tweede lezer Prof. Dr. A. Plaat bedanken voor zijn feedback en bereidheid om mee te denken over de vorm en inhoud van het onderzoek.

Ik wil graag het Leids Universitair Medisch Centrum (LUMC) bedanken voor het beschikbaar stellen van de dataset. Van het LUMC wil ik Dr. M.S. Arbous en Dhr. F.J. Kranenburg bedanken voor hun samenwerking, feedback en inzicht op het gebied van de data waarmee ik gewerkt heb.

Brandon Carter en Angie Boggust van MIT wil ik bedanken voor het werk dat zij in dit project hebben gestoken in de maand dat ze hier in Nederland waren. Zonder hun opzet van het programma had deze scriptie een stuk meer tijd en gepuzzel gekost.

Tot slot wil ik graag mijn ouders en mijn vriendin Tessa bedanken. Pap en mam, bedankt dat jullie altijd vertrouwen in mij hebben gehad ondanks dat ik wat uit ben gelopen en bedankt voor het bieden van hulp op alle momenten dat ik het nodig had. Verder wil ik jullie bedanken voor de rust en het plezier dat altijd bij jullie te vinden valt. Tessa, dankjewel voor alle ontspanning die ik bij je heb kunnen vinden en dat je me de tijd en ruimte hebt gegeven die ik voor mijn scriptie nodig had.

Mark Rasenberg - Januari 2017

Lijst van tabellen

Lijst van figuren

Inhoudsopgave

Hoofdstuk 1

Introductie

In dit hoofdstuk wordt de achtergrond van het onderzoek besproken. De motivatie achter het onderzoek en de opgestelde onderzoeksvraag.

1.1 Achtergrond

Op de Intensive Care afdeling van het Leids Universitair Medisch Centrum (LUMC) zijn 4 Intensive Care Units (ICU's) voor volwassenen en 1 ICU voor kinderen. Op de Intensive Care (IC) krijgen patiënten gespecialiseerde behandeling en zorg vanwege, mogelijk levensbedreigende, problemen met lichaamsfuncties zoals de ademhaling, hartslag en bloeddruk [7].

In de loop der jaren is er een grote hoeveelheid data verzameld over de patiënten en hun verblijf op de IC. Ook in de toekomst zal deze data verzameld blijven worden en zo zal de beschikbare data alleen maar groeien. Onder deze data verstaan we features als leeftijd, geslacht en gewicht, maar ook waarden van metingen die gedaan zijn, zoals het hemoglobinegehalte en de pH waarde van het bloed. Dit zijn enkele voorbeelden van de vele data die opgeslagen wordt.

Met de data zoals hiervoor besproken, wordt nog niet veel gedaan. Aangezien datamining enorm in opmars is, is een voordehandliggende vraag of datamining ook op deze data van de IC toegepast kan worden. Datamining is al op vele gebieden toegepast en heeft bewezen vaak een toegevoegde waarde te hebben. Daarom wordt er bij dit onderzoek gekeken of er ook informatie uit de verkregen data van de IC te halen is. Meer precies zullen we bestuderen of er accuraat voorspeld kan worden of een patiënt wel of geen bloedtransfusie nodig heeft. De achterliggende gedachte is niet dat de computer de beslissing maakt of een patiënt een bloedtransfusie nodig heeft, maar wel een second opinion kan geven. Bovendien kan er een

melding gegeven worden als het er op lijkt dat de patiënt een bloedtransfusie nodig heeft. Bij dit onderzoek gaat het erom of er aan de hand van de statistieken van patiënten in de eerste 24 uur van de opname voorspeld kan worden of deze een bloedtransfusie nodig gaat hebben of niet. Dit is belangrijk omdat een voorspelling als deze het werk van de arts kan ondersteunen, en zo de werkdruk verlagen. Wellicht kunnen er hierdoor meerdere patiënten geholpen worden.

Deze scriptie bouwt voort op het onderzoek waar Brandon Carter en Angie Boggust van het Massachusetts Institute of Technologie aan begonnen zijn. Zij hebben hier in Nederland een maand aan gewerkt. In deze maand hebben ze de backbone van het programma dat gebruikt is om dit onderzoek te doen, grotendeels geschreven. Hun bevindingen waren dat er met de proefdataset en met bepaalde algoritmen een accuracy van maximaal 79% te behalen is. Dit percentage is gebaseerd op het aantal keer dat er correct voorspeld was dat er wel of geen bloedtransfusie nodig was ten opzichte van het totaal aantal patiënten.

In de korte tijd dat Brandon en Angie aan het onderzoek hebben gewerkt is er niet gekeken naar verschillende parameters bij de algoritmen, de te veranderen threshold na het opstellen van het model en er is niet verder gekeken dan de Accuracy van een algoritme. Om een betrouwbaarder model op te stellen is er in dit onderzoek wel naar de genoemde punten gekeken.

Zo is de volgende onderzoeksvraag naar voren gekomen:

"Kan het meest belovende datamining algoritme een nauwkeurigere voorspelling doen dan een voorspelling in eerdere studie gebleken en valt er te interpreteren bij welke waarden van features er wel of geen bloedtransfusie nodig is?"

Om structuur aan te brengen in het onderzoek naar deze onderzoeksvraag, zijn de volgende subonderzoeksvragen opgesteld:

- *"Wat is het meest belovende datamining algoritme om deze voorspelling te doen, gebaseerd op Accuracy, Recall en Precision scores bij de dataset?"*
- *"Is het gebruikte algoritme te interpreteren en valt hier zo uit op te maken welke features van de dataset belangrijk zijn bij de voorspellingen?"*

Ook stippen we kort het probleem van missing values aan. Missing values zijn de niet ingevulde waarden in de dataset. Bij dit onderzoek worden er een aantal basis methodes gebruikt om met dit probleem om te gaan.

1.2 Thesis overzicht

Hoofdstuk 1 bevat de introductie; hoofdstuk 2 geeft een samenvatting van de dataset; hoofdstuk 3 omvat het gerelateerde werk, hieronder valt literatuur, gebruikte algoritmen en statistieken; hoofdstuk 4 bevat de contributies van dit onderzoek; hoofstuk 5 staat de conclusie en future work.

Hoofdstuk 2

De gebruikte dataset

In dit hoofdstuk wordt de door het LUMC beschikbaar gestelde dataset in detail beschreven. Deze dataset is gebruikt voor alle experimenten in deze scriptie.

2.1 Introductie tot de dataset

De dataset die gebruikt wordt in dit onderzoek is een oefen-dataset. De dataset is gebaseerd op echte meetgegevens vervaardigd op de IC van het LUMC. Het gaat hier om 1934 patiënten. Elke patiënt heeft statische en tijdreeks features. Van de 1934 patiënten hebben er 1823 één of meerdere tijdreeks punten. Alleen deze patiënten zijn meegenomen in het onderzoek. De statische features van de patiënt zijn bekend op het moment dat deze op de IC binnenkomt en blijven tijdens het gehele verblijf hetzelfde. Verder zijn er tijdreeks features waarvan de waarden tijdens het verblijf op de IC gemeten worden. Elk van deze features hebben tijdstempels. Als laatst zijn er nog de statisch_24 features. Deze groep wordt gemeten en berekend in de eerste 24 uur van de opname op de IC en veranderen daarna ook niet meer. In de dataset zijn zodoende de volgende features beschikbaar: 34 statische, 8 statische_24 en 38 tijdreeks features. Voor dit onderzoek maken we van sommige features geen gebruik. De redenen waarom we hier geen gebruik van maken staan beschreven in de informatiekolommen van de tabellen die later in dit verslag staan.

2.2 Uitleg beschikbare features

In tabel 2.1 staan de 34 statische features, wat voor feature het is en een stukje informatie over deze feature.

Static_0_Features	feature type	Informatie
age	Float	Leeftijd in jaren
gender	String	Geslacht
weight	Float	Gewicht
height	Float	Lengte
bmi	Integer	Body Mass Index
adm_type	String	Opname type
refer	String	Afdeling die de patiënt heeft verwezen
plan_adm	Boolean	Was de opname gepland?
adm_sur	Boolean	Was de opname na een operatie?
adm_surg_pacu	Boolean	Was de patiënt afkomstig van de operatiekamer of de post - anesthesia care unit?
adm_other_hos	Boolean	Was de opname vanuit een ander ziekenhuis?
readm	Boolean	Was het een heropname?
mech_vent_0	Boolean	Werd de patiënt beademd bij de opname?
aids	Boolean	Heeft de patiënt AIDS?
diabetes	Boolean	Heeft de patiënt diabetes?
copd	Boolean	Heeft de patiënt COPD?
cva	Boolean	Heeft de patiënt een CVA gehad?
cirrhosis	Boolean	Heeft de patiënt levercirrose?
imm_insuf	Boolean	Heeft de patiënt immunologische insufficiëntie?
ac_ren_fail	Boolean	Heeft de patiënt acute nierinsufficiëntie?
chron_dialysis	Boolean	Wordt de patiënt regelmatig gedialiseerd?
dysrhytmia	Boolean	Heeft de patiënt een ritmestoornis?
confirm_infection	Boolean	Heeft de patiënt een bevestigde infectie?
gastro_bleed	Boolean	Heeft de patiënt een gastro-interne bloeding?
intracran_mass	Boolean	Heeft de patiënt intracraniële mass?
burns	Boolean	Heeft de patiënt brandwonden?
thrombo_therap	Boolean	Heeft de patiënt trombolytische therapie gehad na een hartaanval?
cario_vasc_insuf	Boolean	Heeft de patiënt chronische cardiovasculaire insufficiëntie?
chr_renal_insuf	Boolean	Heeft de patiënt chronische nierinsufficiëntie?
cpr	Boolean	Is de patiënt gereanimeerd voor opname?
resp_insuf	Boolean	Heeft de patiënt respiratoire insufficiëntie?
mi_cabg	Boolean	Heeft de patiënt een hartaanval gehad voor een bypass operatie?
neoplasm	Boolean	Heeft de patiënt uitgezaaide kanker?
hem_malign	Boolean	Heeft de patiënt Leukemie?

Tabel 2.1: De statische feature die gelijk beschikbaar zijn bij opname

Bij de statische features in Figuur 2.1 worden een aantal features omgezet naar getallen, namelijk de string features. Aangezien de gebruikte datamining algoritmen niet goed weten om te gaan met strings [13], worden strings omgezet in integers. Een voorbeeld hiervan is de adm_type feature:

'Nursing ward' ⇒ 0
'Other hospital, nursing ward'⇒ 1
'Emergency Department'⇒ 2
'Other ICU and/or hospital' ⇒ 3
'Other High Care department'⇒ 4
'Other' ⇒ 5
'Exclude (non-adult)'⇒ 6

Figuur 2.1: De mapping van de adm_type feature naar integer waarden

Ook bij de static_24_features in Tabel 2.2 en timeseries_features in Tabel 2.3, zijn de string features omgezet naar integers. Omdat de strings naar integers worden omgezet, wordt er automatisch gebruik gemaakt van een ratio schaal. Hierover wordt meer beschreven in hoofdstuk 4. In Figuur 2.2 zijn de features te zien die statisch zijn na 24 uur na opname.

Static_24_Features	**feature type**	**Informatie**
mech_vent_24	Boolean	Heeft de patiënt mechanische beademing gehad binnen 24 uur?
surg_24	Boolean	Heeft de patiënt een operatie ondergaan binnen 24 uur?
apache_ii	Integer	Wat is de apache II score van de patiënt?
apache_iv	Integer	Wat is de apache IV score van de patiënt?
apache_ii_adm_diag1	String	Wat is de apache II diagnose 1? (Niet gebruikt, teveel mogelijkheden)
apache_iv_adm_diag1	String	Wat is de apache IV diagnose 1? (Niet gebruikt, teveel mogelijkheden)
apache_ii_adm_diag_sub1	String	Wat is de apache II diagnose subgroup 1?
apache_iv_adm_diag_sub1	String	Wat is de apache IV diagnose subgroup 1?

Tabel 2.2: De statische features die beschikbaar zijn na de eerste 24 uur na opname

Op de volgende pagina staat Tabel 2.3. Hierin staan alle gebruikte tijdreeks features. Onder andere lab-waarden van het bloed, zoals de hoeveelheid thrombocyten en leucocyten. Ook bijvoorbeeld toegediende medicatie en opnames van de hartslag.

Timeseries_Features	feature type	Informatie
hb_gdl	Float	Wat is het Hemoglobine gehalte in het bloed van de patiënt?
hb_cat	String	In welke categorie valt de Hemoglobine concentratie van de patiënt
ery_trans_6h	Boolean	Heeft de patiënt een bloedtransfusie gehad binnen 6 uur na een Hb-gdl meting? (wordt niet gebruikt aangezien dit zo goed als het antwoord is)
previous_hb	String	Wat was de vorige Hemoglobine waarde van de patiënt?
next_hb	Float	Wat is de volgende Hemoglobine waarde van de patiënt?
minutes_to_next_hb	Integer	Hoe lang tot de volgende Hemoglobine meting van de patiënt (in minuten)?
hb_change	Float	Wat is de verandering van de Hemoglobine waarde van de patiënt ten opzichte van de vorige meting?
day_adm_icu	Integer	Hoeveel dagen is de patiënt opgenomen op de Intensive Care?
ht	Float	Wat is de fractie rode bloedcellen van de patiënt in het bloed?
type_blood_gas	String	Waar is het bloed van de patiënt geprikt?
ph	Float	Wat is de pH waarde van het bloed van de patiënt?
pco2	Float	Wat is de pCO_2 (koolstofdioxide spanning) waarde van het bloed van de patiënt?
po2	Float	Wat is de pO_2 (zuurstofs panning) waarde van het bloed van de patiënt?
baseexcess	Float	Wat is de Base Excess (zuurbase evenwicht) waarde van het bloed van de patiënt?
alkali_reserve	Integer	Wat is de Alkali Reserve waarde van het bloed van de patiënt?
pao2	Float	Wat is de pAO_2 (arteriële zuurstofspanning) waarde van het bloed van de patiënt?
epi	Float	Hoeveel Adrenaline is er via het infuus toegediend?
nor	Float	Hoeveel Noradrenaline is er via het infuus toegediend?
dobu	Float	Hoeveel Dobu (Dobutamine) is er via het infuus toegediend?
dopa	Float	Hoeveel Dopa (Dopamine) is er via het infuus toegediend?
furo	Float	Hoeveel Furo (Furosemide) is er via het infuus toegediend?
thrombo	Float	Wat is de hoeveelheid thrombocyten (bloedplaatjes) in het bloed van de patiënt?
leuco	Float	Wat is de hoeveelheid leucocyten (witte bloedlichaampjes) in het bloed van de patiënt?
bili	Integer	Wat is de hoeveelheid bilirubine (afvalstof van voornamlijk afbraak oude rode bloedcellen) in het bloed van de patiënt?
albumin	Integer	Wat is de hoeveelheid albumine (eiwitmolecuul in het bloedplasma) in het bloed van de patiënt?
lactate	Float	Wat is de lactaat (zuurrest van melkzuur) waarde in het bloed van de patiënt?
creat	Integer	Wat is de creatinine (test om nierfunctie te berekenen) waarde in het bloed van de patiënt?
ck	Integer	Wat is de creatinekinase (enzym) waarde in het bloed van de patiënt (maat voor spierschade)?
tropo	Float	Wat is de troponine (eiwit betrokken bij samentrekken en ontspannen van spieren) waarde in het bloed van de patiënt?
inr	Float	Wat is de international normalized ratio (bloedstolling) waarde in het bloed van de patiënt?
fluid_balance_24h	Integer	Wat is de vochtbalans in het bloed van de patiënt?
spo2	Float	Wat is de zuurstof saturatie waarde in het bloed van de patiënt?
pao2fio2	Float	Wat is de pAO_2/FiO_2 ratio in het bloed van de patiënt?
temp	Float	Wat is de temperatuur van de patiënt?
vent	Boolean	Wordt de patiënt beademd?
ventilation_mode	String	Wat voor beademing heeft patiënt?
fio2	Integer	Wat is de FiO_2 (Fractie ingeademde zuurstof) waarde van de patiënt?
ecmo	Boolean	Heeft de patiënt een ECMO (Extra Corporele Membraam Oxygenatie) behandeling gehad?
hvad	Boolean	Heeft de patiënt een HVAD (steunhart) gehad?
iabp	Boolean	Heeft de patiënt een IABP (Intra-Aortale BallonPomp) behandeling gehad?
cvvh	Boolean	Heeft de patiënt een CVVH (Continue Veno-Veneuze Hemo/Tijdelijke nier dialyse) gehad?
heart_rate	Integer	Wat is de hartslag van de patiënt?
heart_rate_mean	Float	Wat is de gemiddelde hartslag van de patiënt?
abp_systole	Integer	Wat is de systolische bloeddruk van de patiënt?
abp_systole_mean	Float	Wat is de gemiddelde systolische bloeddruk van de patiënt?
abp_diastole	Integer	Wat is de diastolische bloeddruk van de patiënt?
abp_diastole_mean	Float	Wat is de gemiddelde diastolische bloeddruk van de patiënt?
resp_rate	Integer	Wat is de ademhalingsfrequentie van de patiënt?
resp_rate_mean	Float	Wat is de gemiddelde ademhalingsfrequentie van de patiënt?

Tabel 2.3: De tijdreeks features tijdens de opname worden waargenomen [1] [2] [3]

Omdat de gebruikte algoritmen niet weten hoe om gegaan moet worden met tijdreeks features, berekenen we voor elk van de tijdreeks features de volgende features. Dit is te zien in tabel 2.4.

Berekend tijdreeks feature	**feature type**	**Informatie**
min	Float	De minimale waarde van de tijdreeks
max	Float	De maximale waarde van de tijdreeks
mean	Float	Het gemiddelde van de tijdreeks
median	Float	De mediaan van de tijdreeks
stdev	Float	De standaardeviatie van de tijdreeks
lingreg_yint	Float	De intercept van de lineaire regressie, dit is de waarde van y bij $x = 0$
linreg_slope	Float	De helling van de lineaire regressie
expreg_A	Float	e^{linreg_slope}
expreg_B	Float	e^{linreg_yint}

Tabel 2.4: De berekende features van de tijdreeks features

De bovenstaande features zijn geïntroduceerd om van de tijdreeks features gebruik te kunnen maken en zo de voorspelkracht van het programma te vergroten.

Hoofdstuk 3

Gerelateerd Werk

Er is in het verleden al enig onderzoek gedaan naar het gebruik van datamining in onderzoek naar bloedfalen. In de meeste onderzoeken gaat het er echter niet om of er wel of geen bloedtransfusie nodig is, maar gaat het om het voorspellen van de ernst van het bloedfalen [12], om een voorspelling van de ernstige vorm van bloedfalen [27] of een voorspelling van een shock ontstaan uit bloedfalen [17].

Echter zijn er een aantal werken geschreven waarin wel soortgelijk onderzoek wordt gedaan, deze worden hieronder beschreven.

3.1 Gelijkend statistisch onderzoek

Een van de onderzoeken die gedaan is, is de vroege detectie van bloedfalen. Dit is in grote lijnen hetzelfde onderzoek als waar deze scriptie over geschreven is [16]. Ook hier zijn de maximale, minimale, gemiddelde, mediaan en standaard deviatie van tijdreeks features gemeten over de eerste 24 uur van de opname. Echter is er niet naar lineaire en exponentieële regressie gekeken in dat onderzoek. In dat onderzoek is een sample groep van 380 patiënten gekozen en omdat er meer dan 95% van de data gevuld was, zijn de missing values overgeslagen. Verder is er een minimaal verblijf op de Intensive Care van 10 uur gekozen voor een degelijke berekening van de features bij tijdreeks features. In dat onderzoek is er geen gebruik gemaakt van datamining methoden.

Uit dat onderzoek is de conclusie getrokken dat voornamelijk een lichaamstemperatuur van meer dan 38 °C en een Mean Arterial Pressure van minder dan 70 mm Hg belangrijke features zijn of een patiënt wel of geen bloedfalen zal krijgen. De kans is 4,63 keer hoger wanneer er aan deze waarden werd voldaan. Aan de hand van bloeddruk en temperatuur kunnen volgens dit onderzoek bijna 80% van de patiënten correct

geïdentificeerd worden. In tabel 3.1 is te zien dat deze methode een 78,9% correcte identificatie van bloedfalen geeft. Als we verder kijken in de tabel, is te zien dat de voorspelling van het niet hebben van bloedfalen erg laag is [16], wat betekent dat er een grote hoeveelheid onnodige bloedtransfusies gegeven worden als dit onderzoek gebruikt wordt in de praktijk.

	Predicted		
Observed	**No Sepsis**	**Sepsis**	**% Correct**
No Sepsis	157	191	45,1
Sepsis	75	280	78,9
Overall Percentage			62,2

Tabel 3.1: De onderzoeks resultaten van het onderzoek op basis van statistiek [16]

3.2 Decision Tree, Artificial Neural Network, Support Vector Machine en Lineaire Regressie

In een ander onderzoek is gebruik gemaakt van een dataset van 23.446 patiënten van de intensive care. Hiervan is de helft gebruikt om het model te bouwen en de andere helft om het te testen. In dit onderzoek werd de score, die door middel van datamining technieken gevonden werd, vergeleken met de APACHE III score[1] [20]. In dit onderzoek is gebruik gemaakt van een aantal datamining technieken waarvan de uitkomsten te zien zijn in tabel 3.2. Voor missing values is in dit onderzoek de mediaan van de niet missing values ingevuld.

Let op: C5 refereert hier naar het gebruikte Decision Tree Algoritme.

Methods	**AUC**	**SE**
C5	0,892	0,004
ANN	0,874	0,004
SVM	0,876	0,004
APACHE III	0,871	0,003

Tabel 3.2: De 3 gebruikte datamining technieken ten opzichte van de APACHE III − score [20]

[1] deze score wordt normaliter gebruikt voor de mate hoe ziek een patiënt is

3.3 Missing Values

De gebruikte algoritmen weten niet hoe er met ontbrekende waarden omgegaan moet worden en deze moeten dus ingevuld worden. Dit kan met behulp van verschillende methoden. In dit onderzoek is er gekeken naar vier verschillende methoden, namelijk:

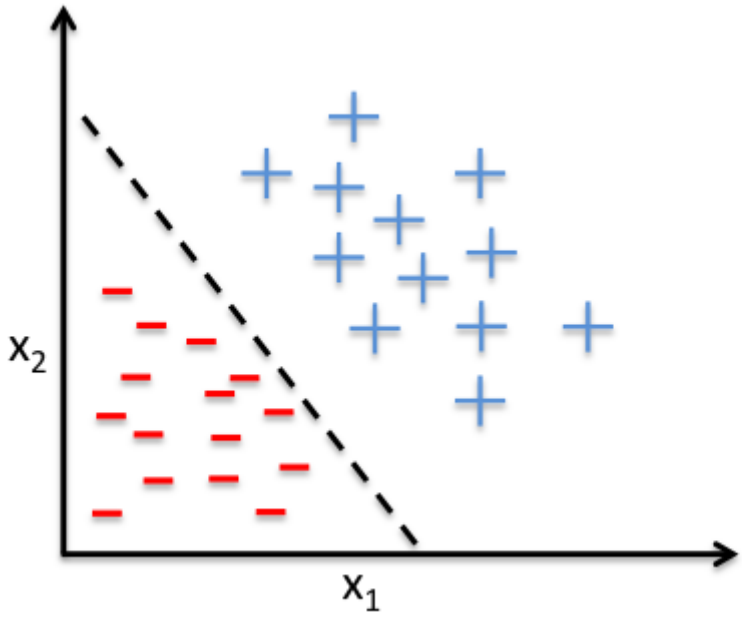
- $0,0$: Hierbij worden alle ontbrekende waarden ingevuld met $0,0$
- Skip: Hierbij worden alle voorbeelden met ontbrekende waarden overgeslagen, waardoor ze niet meedoen bij het bouwen van het model
- Mean: Hierbij wordt het gemiddelde berekend van de waarden in dezelfde kolom die wel beschikbaar zijn. Dit gemiddelde wordt ingevuld op de plek van ontbrekende waarden.
- Median: Hierbij wordt de mediaan berekend van de waarden in dezelfde kolom die wel beschikbaar zijn. Deze mediaan wordt ingevuld op de plek van ontbrekende waarden.

3.4 Datamining algoritmen

Voor deze scriptie stip ik kort alle gebruikte algoritmen aan, met een korte uitleg hoe elk algoritme werkt. Deze algoritmen zijn gebruikt in het programma om de dataset te analyseren. Alle algoritmen zijn supervised, wat inhoudt dat het algoritme leert aan de hand van labels die bij de dataset zitten. Het doel van de algoritmen is om een juiste voorspelling van de klasse van een nieuw datapunt te doen. In dit geval zijn de labels of de arts wel of geen bloedtransfusie heeft gegeven tijdens het verblijf op de IC. Aangezien de dataset al labels heeft aan de hand waarvan we classifiers kunnen leren is er gekozen voor supervised learning.

3.4.1 Perceptron

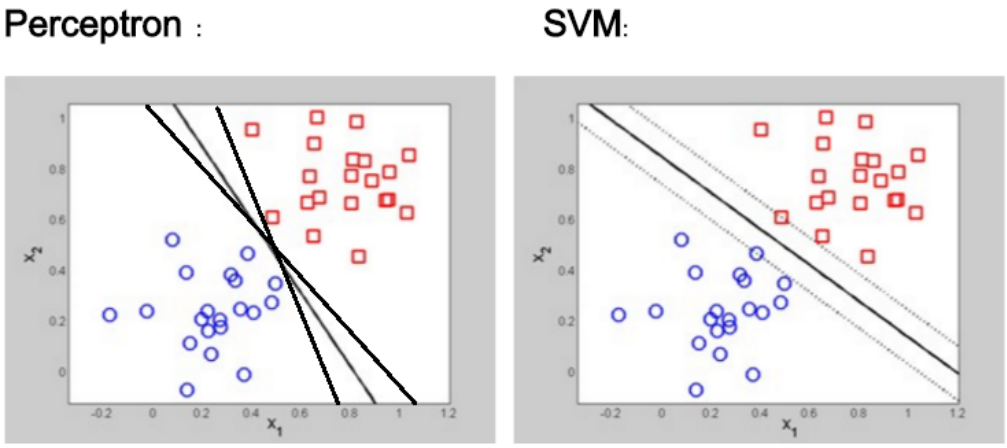
Een perceptron is een neuraal netwerk, dat een rechte lijn trekt (linear) door een dataset om verschillende cases te onderscheiden [13] [15]. Zoals in Figuur 3.1 te zien, onderscheidt deze lijn allebei de mogelijkheden (positief of negatief) perfect. Het is moeilijker om een perfecte scheidingslijn te behalen wanneer een dataset meer attributen heeft. Per extra attribuut komt er namelijk een extra dimensie bij. Het perceptron algoritme lijdt onder het "curse of dimensionality" probleem. Dit probleem wordt uitgelegd in sectie 3.4.8. Een perceptron minimaliseert het aantal fouten dat er gemaakt wordt wanneer een lijn getrokken is en verkleint de grootte van de fout die er bij een bepaalde lijn is. Afhankelijk van hoe een netwerk getraind wordt, zal het model de prioriteit tussen de hoeveelheid fouten en de grootte van de fouten aanpassen.



Figuur 3.1: Een Linear Perceptron in 2 dimensies [4]

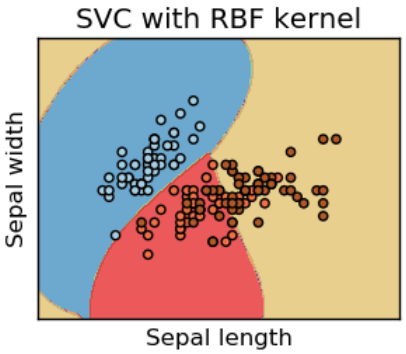
3.4.2 Support Vector Classifier - lineair & RBF

Een Support Vector Classifier (SVC) is een algoritme dat net als een perceptron door middel van een lijn verschillende klasse cases probeert te scheiden. Binnen een SVC kan er gebruik gemaakt worden van verschillende Kernels. Een Kernel bepaalt wat voor lijnen er getrokken mogen worden. Er is in dit onderzoek gebruik gemaakt van een Linear- en Radial Base Function (RBF). Het verschill tussen een Perceptron en een SVC is dat een perceptron zo klein mogelijke fouten maakt terwijl een SVC de marge vanaf de lijn tot datapunten zo groot mogelijk wil maken. Dit verschil is goed te zien in Figuur 3.2, waar beide vizualisaties naast elkaar gezet zijn. In het linker figuur is te zien dat de perceptron verschillende lijnen kan kiezen, die allemaal geen fouten maken en dus allemaal even goed zijn, terwijl er in het rechter figuur te zien is dat er maar één lijn mogelijk is, namelijk de lijn met de grootste marge tot de datapunten [11] [10].



Figuur 3.2: Vergelijking Perceptron en SVC - Linear Kernel, beide gebaseerd op een 2 dimensionale dataset [8]

In Figuur 3.3 is de RBF - Kernel gevisualiseerd. RBF gebruikt radial functies in plaats van lineaire functies, hiermee zijn er cirkelvormige lijnen te vormen. In de visualisatie is te zien dat deze lijnen een gebogen vorm hebben.



Figuur 3.3: Een Support Vector Machine met een Linear Kernel en Radial Basis Function (RBF) Kernel in 2 dimensies [5]

3.4.3 Decision Tree [21] [23] [24]

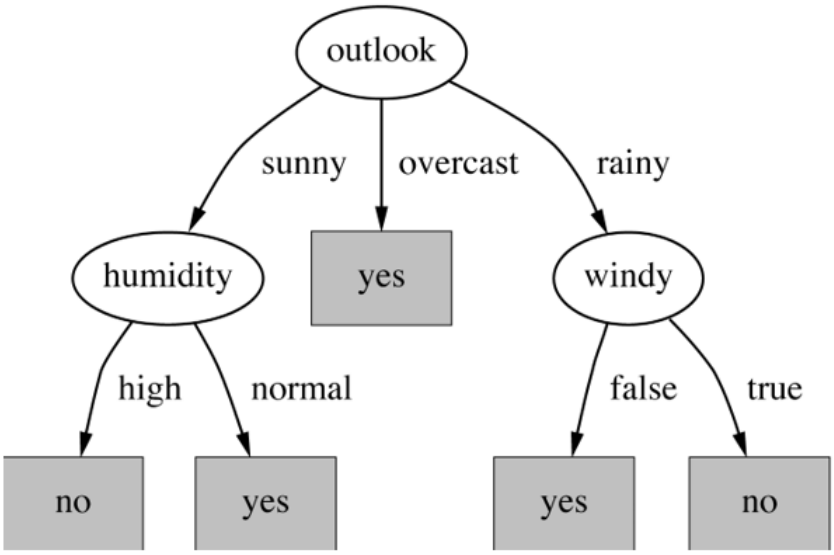
Een decision tree bestaat uit een root, internal nodes, branches en leaves. De root is de bovenste node van een decision tree. Hier begin je wanneer je een nieuwe casus wilt classificeren. Vervolgens worden branches gevolgd, die de uitkomst van een test in de internal nodes voorstellen. In Figuur 3.4 gaat dit als volgt:

- Root: Outlook
- Internal Nodes: Humidity, Windy
- Branches: Sunny, Overcast, Rainy, High, Normal, False, True
- Leaves: All Yes and No

In dit voorbeeld zijn er vanuit sommige internal nodes meer dan 2 paden die gekozen kunnen worden, zoals Outlook in dit geval. De Decision Tree die gebouwd is in het onderzoek is een zogenaamde binary tree, dit betekent dat er altijd precies twee kinderen van een internal node komen.

Via de tests in de internal nodes eindigt een nieuw datapunt bij een van de leaves. De leaf waar het nieuwe datapunt terecht komt, is het label dat er aan dit datapunt gehangen wordt, in het geval van het voorbeeld of je wel of niet gaat tennissen en in het geval van dit onderzoek of er wel of geen bloedtransfusie nodig is. Verschillende algoritmen voor het bouwen van deze decision trees zijn beschikbaar. De meest populaire zijn ID3 en C4.5 [13]. De gebruikte python library SKLearn gebruikt het CART-algoritme, wat staat voor Classification And Regression Tree. Dit houdt in dat deze voor zowel classificatie als regressie gebruikt kan worden. Classificatie is het hangen van een label (bijvoorbeeld wel of geen bloedtransfusie) aan een nieuw datapunt met behulp van een dataset. Regressie is de voorspelling van een continue feature (bijvoorbeeld het

temperatuurverloop van de dag) met behulp van een dataset. Bij dit onderzoek is classificatie van veel groter belang dan regressie. Het CART-algoritme zoekt de beste uitkomst van beide manieren en bouwt zo een boom. Om overfitting tegen te gaan hebben we verschillende maximale diepten gebruikt. Bij een maximale diepte van 3 was het percentage correcte voorspellingen maximaal.

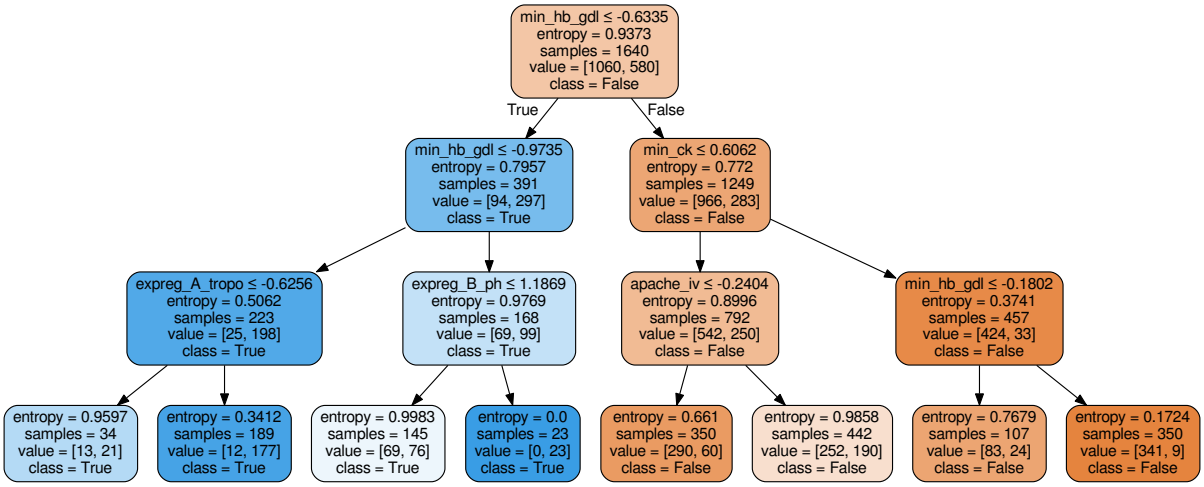


Figuur 3.4: Een decision tree, die beslist of je wel of geen tennis gaat spelen [6]

Decision trees hebben ook een grote toegevoegde waarde bij uitgebreide data met veel features, waardoor het bij ons probleem een interressante keuze is. Ook zijn Decision Trees te visualiseren, zoals te zien is in Figuur 3.4. In elke node zijn een aantal waardes te lezen. De bovenste is de Decision Rule, in de eerste node wordt er dus een splitsing gemaakt op wat de voorspelling zegt, als dit zonnig (Sunny) is dan wordt het pad naar het linker kind gevolgd, als het bewolkt (overcast) is dan wordt het middelste pad gevolgd en wanneer het gaat regenen (rainy) dan wordt het pad naar het rechterkind gevolgd. Zo volg je een pad totdat je in een van de Leaves onderaan het model komt. In deze leaves ontbreekt de Decision Rule aangezien je hier niet meer verder gaat. De onderste waarde (class = yes/no) betekent of iemand wel of niet gaat tennissen. In de decision tree die in dit onderzoek is gebouwd (zie Figuur 3.5), wordt gebruik gemaakt van de volgende metrieken:

- Entropie is hoeveel "informatie" er aan de hand van deze Decision Rule gevonden kan worden. Deze waarde loopt van 0 tot 1, waarbij 0 staat voor het volledig splitsen van True en False, wat te zien is in de vierde Leaf van links in Figuur 3.5. Wanneer er precies evenveel True als False zijn, is de entropie 1. Het doel is om de entropie in de Leaves zo laag mogelijk te krijgen, wat in dit Figuur op sommige paden heel goed gaat en sommige paden totaal niet.
- Samples is het aantal cases dat via dit pad loopt bij het trainen van het algoritme en het creeëren van deze boom.

- Value bestaat uit twee waarden. De eerste waarde is het aantal False-Cases en de tweede is het aantal True-Cases. Value hangt direct samen met Entropie. Wanneer het verschil tussen deze twee waarden groter wordt, wordt de entropie lager.



Figuur 3.5: De Decision Tree, gemaakt aan de hand van de Oefen-Dataset van het LUMC met een maximale diepte van 3

3.4.4 Random Forest

Een Decision Tree algoritme heeft zonder restricties de eigenschap om voor elke casus een apart pad naar een blad van de boom te maken. Dit verschijnsel noemt men Overfitting. Dit valt handmatig aan te pakken door een maximale diepte van de decision trees of een maximum aantal voorbeelden in een leaf mee te geven als parameter. Het Random Forest algoritme is een algoritme dat door het gebruik van verschillende decision trees het overfitten tegen gaat en zo een beter model opstelt. Dit doet het Random Forest algoritme door een aantal decision trees te maken die elk een aantal features van het totale aantal features tot zijn beschikking heeft. De features die beschikbaar zijn voor de verschillende bomen worden random bepaald (dit is het random gedeelte van het RandomForest algoritme). Door dit willekeurige element in dit algoritme, wordt het overfitten tegen gegaan en wordt dit algoritme erg interessant om te testen op de dataset [18] [19] [26].

3.4.5 ExtraTrees

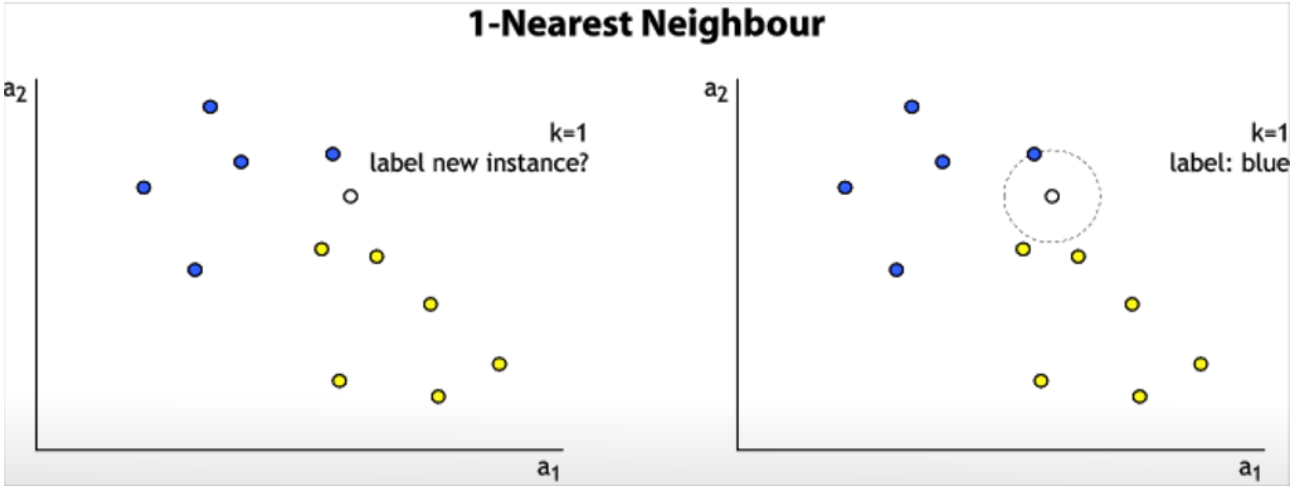
Het ExtraTrees algoritme, of Extremely Randomized Trees algoritme, werkt ongeveer hetzelfde als het Random Forest algoritme. Behalve dat de features die gebruikt mogen worden willekeurig geselecteerd worden (zoals bij het RandomForest algoritme) wordt er ook bij de internal nodes op willekeurige waarden gesplitst [22].

3.4.6 AdaBoostClassifier

Het AdaBoost algoritme, wat een afkorting is voor "Adaptive Boosting" algoritme, traint zichzelf door meerdere keren een nieuw simpel model te trainen. Deze modellen zijn gebaseerd op de fouten van het vorige model. Zo begint het algoritme met een standaard model en worden er gewichten aan de fout voorspelde patiënten gehangen. Zo worden de moeilijk te onderscheiden patiënten gevonden en kan de volgende sequentie van het algoritme meer focussen op deze moeilijkere patiënten. In elke iteratie van het algoritme behalve bij de eerste iteratie, wordt er gebouwd met als doel een zo laag mogelijke weighted error [2]. Aan het einde van het algoritme wordt de combinatie van de hele set gebruikt met een combinatie van gewichten en wordt het uiteindelijke model gebouwd. Het AdaBoost algoritme maakt gebruik van vele "zwakke" leer algoritmen, die op zichzelf geen moeilijke voorspelling kunnen doen. Echter door een combinatie van modellen uit dit algoritme te gebruiken, is het toch mogelijk om complexere vraagstukken te beantwoorden. [25] [14] [28].

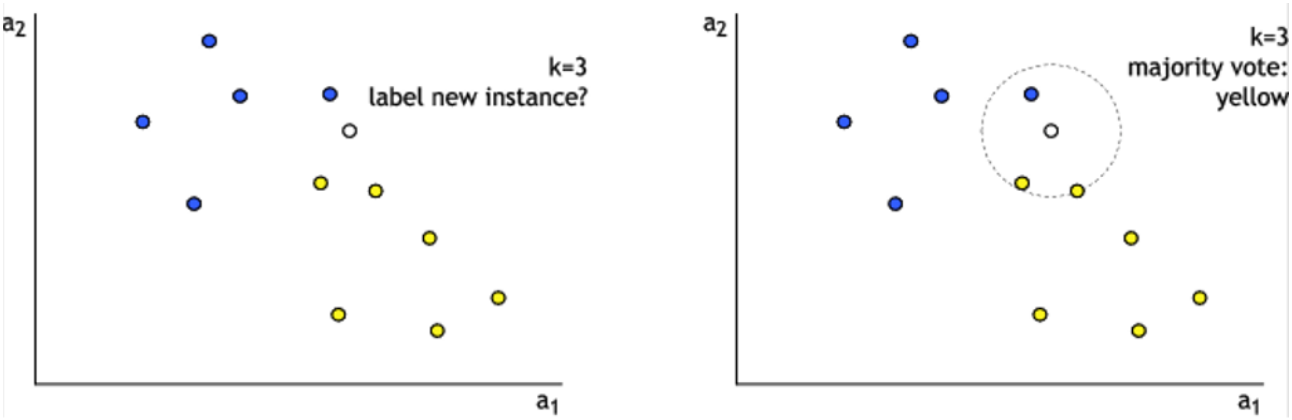
3.4.7 K-Nearest Neighbour

Het k-Nearest Neighbour algoritme kijkt naar de k-dichtstbijzijnde (meest gelijkende) datapunten in de dataset en neemt van deze datapunten de meest voorkomende klasse over. k is in dit geval een positief natuurlijk getal, die het liefst oneven is om een gelijkspel te voorkomen. In Figuur 3.6 is te zien dat wanneer k = 1 het nieuwe datapunt de blauwe klasse aanneemt. Wanneer k = 3, zoals te zien in Figuur 3.7, dan neemt het nieuwe datapunt de klasse geel aan. Het is bij dit algoritme dus heel erg belangrijk welke waarde van k je neemt. Stel k = 2, dan zou er een gele en een blauwe buurman geweest zijn en is het niet duidelijk welke klasse het nieuwe datapunt moet aannemen [13].



Figuur 3.6: Wanneer k = 1 in deze dataset, dan wordt het nieuwe punt blauw

[2] in de voorbeelden met hogere gewichten mogen minder fouten gemaakt worden dan met lagere gewichten



Figuur 3.7: Wanneer k = 3 in deze dataset, dan wordt het nieuwe punt geel

Een reden waarom dit algoritme minder goed presteert, is de curse of dimensionality besproken in 3.4.8. Wanneer er een grotere dataset ter beschikking wordt gesteld kan dit algoritme opnieuw geëvalueerd worden. Met een grotere dataset wordt het algorimte preciezer, maar het doorrekenen van het algoritme duurt ook steeds langer.

3.4.8 "Curse of Dimensionality"

Dit houdt in dat een algoritme per extra feature met een extra dimensie te kampen heeft. Hierdoor wordt met elke dimensie de afstand tussen datapunten groter, waardoor het steeds moeilijker wordt om een goed model op te stellen. Een oplossing voor dit probleem is een grotere dataset op te stellen, deze moet per dimensie exponentieel groeien. Dit zorgt er op zijn beurt voor dat de tijd om het algoritme uit te voeren alsmaar groter wordt [9].

3.5 Evaluatie statistieken

De hierna beschreven metingen zijn gebruikt om te bekijken hoe goed de voorspelling van een programma is. Bij het doen van een voorspelling zijn er een aantal mogelijke uitkomsten:

- TP: True Positive (Een correcte voorspelling dat er een bloedtransfusie gedaan zal worden)
- TN: True Negative (Een correcte voorspelling dat er geen bloedtransfusie gedaan zal worden)
- FP: False Positive (Een foute voorspelling dat er een bloedtransfusie gedaan wordt terwijl deze niet gedaan wordt)
- FN: False Negative (Een foute voorspelling dat er geen bloedtransfusie gedaan wordt terwijl deze wel gedaan wordt)

3.5.1 Accuracy

De accuracy van het programma is het precentage correct geclassificeerde datapunten bij een 10-fold cross-validation, dit wordt hieronder uitgelegd. De accuracy laat niet zien hoeveel het programma ernaast zit omdat het niet duidelijk is of het om False Positives of False Negatives gaat.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

10-Fold Cross-Validation

10-Fold Cross-Validation houdt in dat de dataset wordt opgedeeld in 10 gelijke delen. Hiervan wordt op 9 delen getraind en op 1 deel getest, dit gebeurt 10 keer met elke keer een ander deel waarop getest wordt, todat elk deel een keer is gebruikt om op te testen.

3.5.2 Precision

De precision of precisie van het programma is het percentage correcte voorspellingen waarbij wel een bloedtransfusie wordt voorspeld ten opzichte van het totaal aantal correcte voorspellingen.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3.5.3 Recall

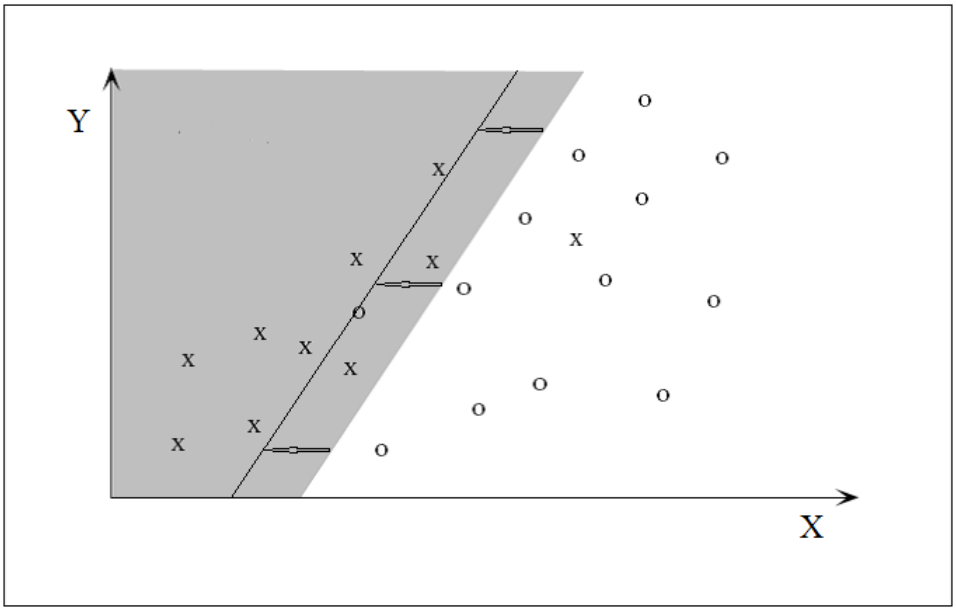
De recall is het percentage correcte voorspellingen waarbij een bloedtransfusie wordt voorspeld ten opzichte van het totaal aantal keren dat een bloedtransfusie voorspeld zou moeten worden.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.5.4 Threshold (θ)

De threshold is het getal (θ in functie 3.4) waarmee wordt vergeleken of er wel of geen bloedtransfusie voorspeld wordt. De threshold wordt vergeleken met de Functie 3.4 die door middel van alle features een waarde berekent. In Figuur 3.8 is een voorbeeld te zien van een lineaire functie van twee features. Hierin is te zien dat wanneer de lijn naar links wordt verschoven, het laatste rondje ook goed voorspeld wordt. Dit gaat echter wel ten koste van twee kruisjes die verkeerd voorspeld worden. Het verplaatsen kan leiden tot een hogere recall score of een hogere precision score. Een hogere recall score betekent dat er meer correcte 'wel een bloedtransfusie' wordt voorspeld. Een hogere precision score betekent dat er meer correcte 'geen bloedtransfusie' wordt voorspeld. In dit onderzoek is het belangrijker dat de recall score hoger is, omdat het om mensenlevens gaat.

$$ax_1 + bx_2 + ... + nx_n \geq \theta \tag{3.4}$$



Figuur 3.8: Wanneer de lijn (threshold) zich verplaatst, wordt het laatste rondje ook goed benoemd, ten koste van 2 kruisjes

3.5.5 feature_importances_

Een manier om het belang van features te bepalen is met de Feature_importances_ functie van sklear n. Bij deze functie wordt er gebruik gemaakt van Gini Importance op basis van de gini impurit (zie sectie 3.5.6). Deze metriek wordt berekend op basis van de dataset, voor- en nadat deze in een node van één van de random forest trees gesplitst is. Wanneer er alleen maar False of alleen maar True waarden in een node overblijven is de entropy 0. Wanneer de False en True waarden precies gelijk aan elkaar zijn dan is de entropy 1. Hoe dichter deze waarde bij de 0 is, hoe belangrijker een feature is. De feature_importances_ functie maakt vervolgens van de verschillende gini impurity waarden, verzameld in verschillende bomen van het Random forest, een waarde waarop de features gerangschikt worden. De feature_importances_ functie berekent de Gini Importances van features op basis van het model die gebouwd is. Dit verschilt dus elke keer dat er een random forest wordt gebouwd en er zijn dus ook telkens andere belangrijke features.

3.5.6 Gini Impurity

Gini impurity is een manier om te meten hoe belangrijk een bepaalde node in een van de bomen in een Random Forest is. De feature die in deze node aanwezig is, is daarmee een belangrijke feature. Gini impurity is gebaseerd op een node in een decision tree. Er is een bepaalde verhouding van wel of geen bloedtransfusie vóór de node en een verhouding na de splitsing op deze node. Eerst moet de initiële Gini impurity voor de node berekend worden. Dit gebeurt aan de hand van de volgende formule:

$$I(n) = 1 - P(n_+)^2 - P(n_-)^2 \tag{3.5}$$

In deze formule wordt de Gini impurity van node n berekend $I(n)$ met behulp van $P(n_+)$ en $P(n_-)$. Vervolgens wordt deze formule ook gebruikt na de splitsing van deze node n. Voor zowel het rechter als het linkerkind. Het verschil tussen de Gini index vóór en na de splitsing maakt hoe belangrijk een feature is voor een decision tree. Hierbij wordt er ook rekening gehouden met de proporties $+$ en $-$ datapunten:

$$GiniWinst(n) = I(n) - P(I_{links}(n)) - P(I_{rechts}(n)) \tag{3.6}$$

$P(I_{links}(n))$ en $P(I_{rechts}(n))$ zijn respectievelijk de notatie voor de berekende Gini impurity van het linkerkind en van het rechterkind.

Hoofdstuk 4

Contributies

In dit hoofdstuk worden de bevindingen die uit dit onderzoek naar voren zijn gekomen besproken. Deze bevindingen zijn gevonden aan de hand van de dataset in hoofdstuk 2 beschreven en de SKLearn library in Python. De code waarmee deze onderzoeken zijn gedaan is geschreven in Python.

4.1 Problemen in de data

Het model kan problemen hebben wanneer de dataset het volgende bevat: 2 patiënten met dezelfde features. Stel dat de volgende 2 patiënten in de dataset staan:

1. Man, 54 jaar, komt beademd binnen, geen bloedtransfusie
2. Man, 54 jaar, komt beademd binnen, wel bloedtransfusie

Bij het opstellen van het model is het niet duidelijk wat er met mannen van 54 jaar die beademd worden moet gebeuren. In onze dataset zijn er maar 5 patiënten die meer dan 52% exact overeenkomende features hebben. Op deze cijfers gebaseerd zal het voorgaand besproken probleem uitgesloten worden in deze dataset.

Bij het handmatig controleren van de data waren er een aantal patiënten die opvielen in de dataset. Het gaat hier bijvoorbeeld om patiënten met een lengte van 0,00187 cm die daarmee een berekend BMI hebben van meer dan 300.000. Deze data is verkeerd ingevuld door een fout in het omrekenen of het verschil tussen een komma en een punt. Ondanks deze grote verschillen heeft dit niet voor problemen in het model gezorgd.

Bij het veranderen van de foute waarden in de dataset of het uitsluiten van de patiënten die erg veel op elkaar leken, zijn er geen verbeteringen in de data gevonden.

4.2 Resultaten

Voor het onderzoek is er eerst met behulp van de in hoofdstuk 3.4 beschreven classficatie-algoritmen bepaald welke algoritmen er verder uitgezocht moesten worden voor deze dataset. Aan de hand van de data in tabel 4.1, waarbij voor elk algoritme de accuracy bij deze dataset is berekend, is er vervolgens voor gekozen om nader onderzoek te doen bij de bovenste 5 algorimen. Alle resultaten in dit hoofdstuk besproken zijn met een 10-fold-cross validation gegenereerd, behalve de figuren 4.1, 4.2, 4.3, 4.4 en 4.5 en de daarbij horende waarden in tabel 4.5

SVM RBF	0,82 +- 0,05
Adaboost	0,80 +- 0,02
Random Forest	0,78 +- 0,07
ExtraTrees	0,78 +- 0,03
Decision Tree	0,77 +- 0,04
Perceptron	0,77 +- 0,02
SVM Linear	0,77 +- 0,05
k-NearestNeighbour	0,76 +- 0,01

Tabel 4.1: De Accuracy waarden, bij de gebruikte algoritmen

Nadat deze keuze gemaakt is, is er bij deze algoritmen gekeken of er significante verschillen zijn, als je de eerste 6, 12, 24 of 48 uur aan datapunten in de dataset neemt. Hierbij is opgevallen dat de eerste 6 en 12 uur de accuracy dusdanig veel lager wordt dat het de verkorte tijd niet waard is. Het verschil tussen 24 uur en 48 uur resulteert in minimale verbetering, vandaar dat 48 uur niet afgewacht wordt. De exacte resultaten staan in tabel 4.2. Verder in deze scriptie wordt er alleen gebruik gemaakt van de datapunten van tijdreeks features die in de eerste 24 uur na opname zijn vastgelegd.

Naast bovengenoemde tijdreeks features is er gekeken naar de verschillende manieren van het invullen van missing values. Deze missing values kunnen ontstaan zijn doordat deze simpelweg niet door de verpleegkundige of arts zijn ingevuld of dat bijvoorbeeld het systeem niet goed werkte.

De resultaten van deze methoden zijn met bijbehorende classifier scores in tabel 4.3 te zien. Uit deze tabel valt op te maken dat wanneer de missing values met het gemiddelde van de overige datapunten wordt ingevuld, de uitslag van het algoritme het beste is. Voor het vervolg van dit onderzoek is deze manier van omgaan met missing values gebruikt.

Omdat in dit onderzoek Accuracy alleen geen niet is waarop we willen vergelijken, is er ook gekeken naar

Aantal uren dataset	**SVM/RBF**	**DecisionTree**	**Random Forest**	**AdaBoost**	**ExtraTrees**
6	0,77 +- 0,06	0,72 +- 0,03	0,74 +- 0,04	0,77 +- 0,03	0,76 +- 0,03
12	0,81 +- 0,05	0,77 +- 0,02	0,75 +- 0,06	0,77 +- 0,03	0,77 +- 0,03
24	0,82 +- 0,05	0,77 +- 0,04	0,78 +- 0,07	0,80 +- 0,02	0,78 +- 0,03
48	0,83 +- 0,05	0,79 +- 0,03	0,81 +- 0,05	0,81 +- 0,04	0,81 +- 0,03

Tabel 4.2: De classifier waarden voor data binnen een bepaalde tijd

Methode	SVM/RBF	DecisionTree	Random Forest	AdaBoost	ExtraTrees
Missing value = 0, 0	0,81 +- 0,04	0,77 +- 0,05	0,78 +- 0,04	0,80 +- 0,03	0,78 +- 0,04
Missing value = skip	0,81 +- 0,04	0,77 +- 0,05	0,78 +- 0,03	0,80 +- 0,03	0,78 +- 0,05
Missing value = mean	0,82 +- 0,04	0,77 +- 0,04	0,78 +- 0,07	0,80 +- 0,02	0,78 +- 0,03
Missing value = median	0,82 +- 0,05	0,77 +- 0,07	0,78 +- 0,05	0,79 +- 0,03	0,78 +- 0,07

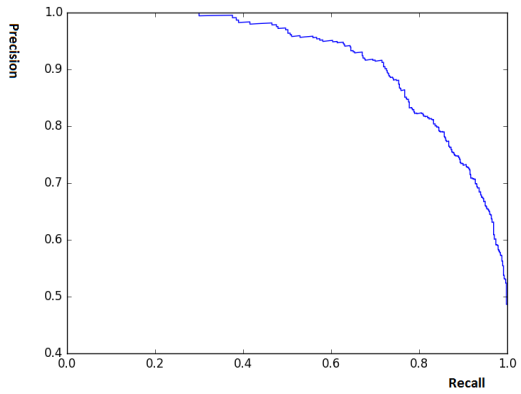
Tabel 4.3: Verschillende manieren om met missing values om te gaan

de precision en recall (uitgelegd in respectievelijk hoofdstuk 3.5.2, hoofdstuk 3.5.3). Deze waarden zagen er voor een aantal algoritmen een stuk veelbelovender uit dan de accuracy alleen. Dit is te zien in tabel 4.4. Voor dit specifieke onderzoek is het van belang om de recall zo hoog mogelijk te krijgen, het liefst 100%. Dit mag eventueel ten koste van de precision. Het is beter een bloedtransfusie teveel voorspellen dan dat er een mensenleven verloren gaat. In tabel 4.4 is te zien dat het ExtraTrees algoritme ondanks de redelijke accuracy, niet goed scoort op de Recall en Precision statistieken. Verder is in de tabel te zien dat de overige 4 algoritmen goed naar voren komen, met recall scores van meer dan 90%. De Precision is voor SVM met RBF kernel en Random Forest beter dan ExtraTrees, DecisionTree en Adaboost.

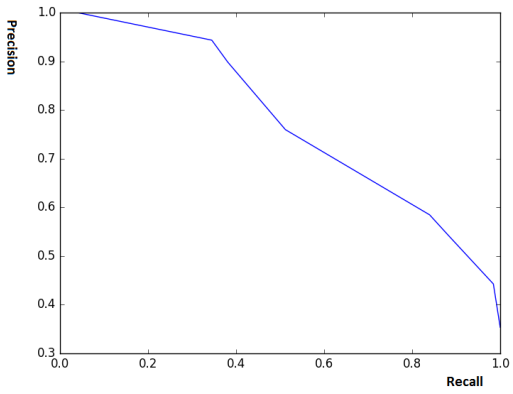
Statistiek	SVM/RBF	DecisionTree	Random Forest	AdaBoost	ExtraTrees
Accuracy	0,82 +- 0,04	0,77 +- 0,04	0,78 +- 0,07	0,80 +- 0,02	0,78 +- 0,03
Recall	94,24%	97,04%	93,15%	97,04%	<60%
Precision	95,11%	77,60%	96,13%	87,52%	77,30%

Tabel 4.4: Accuracy, Recall en Precision voor de gebruikte algoritmen. 24 uur na opname.

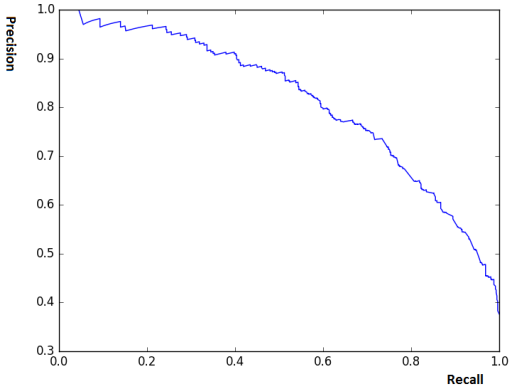
Voor elk van deze algoritmen is een Precision−Recall Curve gemaakt, waarbij er verschillende waarden voor θ zijn gebruikt. De recall scores waar we naar zoeken moeten zo hoog mogelijk liggen zonder de bijbehorende precision scores teveel te schaden. Elke bloedtransfusie teveel is ook niet goed. Deze curves zijn te zien in de volgende Figuren: 4.1, 4.2, 4.3, 4.4 en 4.5.



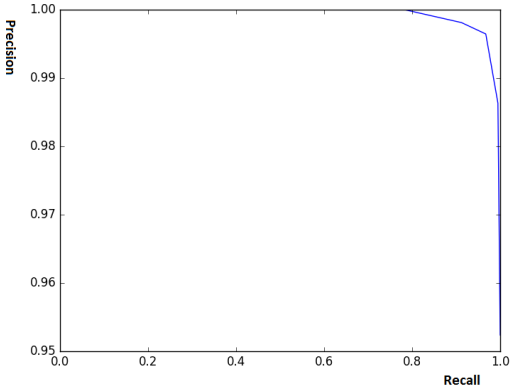
Figuur 4.1: Precision−Recall curve, AdaBoost



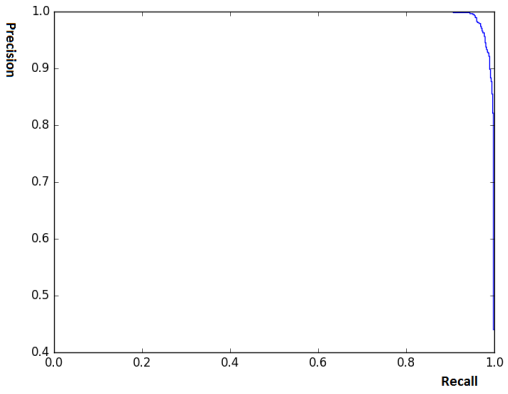
Figuur 4.2: Precision−Recall curve, DecisionTree



Figuur 4.3: Precision−Recall curve, ExtraTrees



Figuur 4.4: Precision−Recall curve, RandomForest



Figuur 4.5: Precision−Recall curve, SVM met RBF kernel

De waarden in tabel 4.5 zijn de maximaal mogelijke precision bij een recall van 100%, per getest algoritme. Hierbij staat de bijbehorende threshold om deze waarden te bereiken. Het enige algoritme dat redelijk goed scoort op de precision statistiek, is het RandomForest algoritme.

Statistiek	SVM/RBF	DecisionTree	Random Forest	AdaBoost	ExtraTrees
Maximale Precisie	44,04%	35,36%	95,23%	48,66%	37,64%
Threshold	-1.1252	0.0257	0.4	-0.0680	0.0731

Tabel 4.5: De maximale precision per algoritme waar de recall 100% is en de bijbehorende Threshold

4.3 Interpreteren en visualiseren

In de hiervoor geschreven paragraaf is het RandomForest algoritme als beste algoritme naar voren gekomen. De volgende stap is het interpreteren en eventueel visualiseren van het algoritme. Dit moet de medewerkers van het LUMC méér dan alleen een voorspelling geven, maar ook een idee geven waar de voorspelling vandaan komt. Het probleem van het algoritme is dat het dusdanig ingewikkeld en uitgebreid is, dat het voor een mens niet goed te interpreteren of visualiseren is. De hierna gepresenteerde resultaten zijn gebaseerd op verschillende modellen van het Random Forest Algoritme aangezien een Random Forest Algoritme elke keer een ander model opstelt.

4.3.1 Handmatig bekijken van de hoogste nodes

Een eerste manier van interpreteren en visualiseren is het visualiseren van de trees die in het model worden gegenereerd. Hierin worden de features in de bovenste nodes van de dataset als het belangrijkst bestempeld. De reden om de bovenste nodes handmatig te controleren is, omdat deze nodes de grootste split in de dataset maken en ze hierdoor het belangrijkst zijn. Tijdens het onderzoek zijn er 4 verschillende combinaties van parameters gebruikt. Deze combinaties zijn in de onderstaande opsomming weergegeven. De genoemde parameters zijn meegegeven aan het SciKit Learn Random Forest algoritme om het interpreteren van het algoritme mogelijk te maken. Dit zijn de volgende 4 combinaties:

- 10 bomen zonder maximale diepte
- 10 bomen met een maximale diepte van 4
- 4 bomen zonder maximale diepte
- 4 bomen met een maximale diepte van 4

De belangrijkste features bij deze combinaties zijn in tabel 4.6 gegeven. In deze tabel staan de features gepresenteerd die in de bovenste nodes van de trees staan.

10/∞	10/4	4/∞	4/4
min_iabp	hb_gdl	linreg_alkali_reserve	max_ck
hb_cat	linreg_yint_alkali_reserve	linreg_slope_ecmo	min_hb_gdl
ht	linreg_yint_alkali_reserve	hb_gdl	apache_iv
min_dopba	stdev_ck	ht	hb_gdl
hb_cat	stdev_alkali_reserve		
stdev_ecmo	median_hb_gdl		
min_lactate	min_dopa		
mean_ph	expreg_A_vent		
min_dopa	expreg_B_pao2		
mean_hb_change	max_ck		

Tabel 4.6: Het eerste getal in de bovenste rij is de hoeveelheid gegenereerde bomen en het tweede getal is de diepte

4.3.2 feature_importances_ functie SKlearn

Een tweede manier van interpreteren en visualiseren is met het gebruik van de feature_importances_ functie van scikit learn. Deze geeft op basis van Gini impurity de meest belangrijke features terug.

De resultaten van bovengenoemde berekeningen zijn te zien in de tabel 4.7. Het probleem van deze methode is dat er aan de geselecteerde features geen richting zit. Met deze richting wordt bedoeld of bijvoorbeeld de leeftijd hoger of juist lager moet zijn om een bloedtransfusie te voorspellen.

Feature	Importance
min_hb_gdl	0,042895
median_hb_gdl	0,035641
stdev_alkali_reserve	0,01512533
linreg_slope_fio2	0,014618
min_bili	0,0144714
max_ck	0,0137409
hb_gdl	0,0135373
stdev_ck	0,0127206
stdev_pco2	0,0122307
expreg_B_dobu	0,0110935
max_albumin	0,0111104
expreg_A_fio2	0,0114542
linreg_slope_leuco	0,010936
linreg_yint_alkali_reserve	0,0101978

Tabel 4.7: Resultaten feature importances, op basis van door scikit learn geselecteerde features

4.3.3 Tree Interpreter Library

De derde manier van interpreteren en visualiseren is via de Tree Interpreter Library van Python. Hierin wordt voor elke feature een contributie bepaald. Met deze contributie per feature en een vaste bias wordt voor beide labels (bloedtransfusie of niet) een voorspelling berekend door middel van formule 4.1

$$prediction = bias + feature\_1\_contribution + ... + feature\_n\_contribution \tag{4.1}$$

Attribute	Not Transfused	Transfused
age	-0,0857	0,0857
expreg_A_apb_systole	0,0666	-0,0666
expreg_B_hvad	0,0967	-0,0967
expreg_B_resp_rate	-0,0500	0,0500
gender	-0,0545	0,0545
ht	-0,1101	0,1101
linreg_slope_apb_systole	-0,0744	0,0744
m_alkali_reserveax	0,0555	-0,0555
mean_baseexcess	-0,0547	0,0547
mean_lactate	-0,0916	0,0916
median_lactate	-0,0931	0,0931
median_ph	-0,0604	0,0604
median_ph	-0,0850	0,0850
median_thrombo	-0,0953	0,0953
min_bili	-0,0635	0,0635
min_bili	-0,0833	0,0833
stdev_alkali_reserve	-0,0620	0,0620

Tabel 4.8: De impact van verschillende features op de wel of geen bloedtransfusie klasses

De feature contributions worden als volgt berekend. Er wordt in SciKit Learn bijgehouden welke paden er worden gebruikt voor voorspellingen, waarna deze paden worden gebruikt door de Tree Interpreter Library. Van het pad dat voor de voorspellingen afgelegd is, wordt er berekend hoe vaak bepaalde features gebruikt worden. Van alle paden, van deze voorspellingen, krijgen de features die vaker voorkomen in paden, een hogere waarde.

Deze feature contributions, vertellen of een feature een positieve of negatieve impact heeft op de klasse. Zo is er te zien dat leeftijd invloed heeft in de richting van een transfusie, dit houdt logischerwijs in dat leeftijd een omgekeerde invloed heeft richting geen transfusie. Een overzicht van de hoogste invloeden die uit deze methode naar voren zijn gekomen, zijn te vinden in tabel 4.8. Dit zijn de waarden die gevonden zijn bij 5 Random Forest Modellen.

Hoofdstuk 5

Conclusies

Uit het voorgaand onderzoek is gebleken dat wanneer er datamining toegepast wordt op data van de Intensive Care van het Leidsch Universitair Medisch Centrum, er belangrijke resultaten geboekt kunnen worden. Tijdens het onderzoek is gebleken dat verschillende algoritmen gebruikt kunnen worden. Elk algoritme heeft specifieke voor- en nadelen. Bij dit specifieke onderzoek is gebleken dat het RandomForest algoritme het beste naar voren komt.

In een eerdere studie is een accuracy gehaald van 79% waar een RandomForest algoritme in dit onderzoek maar een accuracy van 78% haalt. Het RandomForest algoritme is echter wel sterker als het aankomt op recall en precision. Hier haalt het algoritme een maximale precisie van 95,23% bij een recall van 100%. Omdat dit algoritme te maken heeft met de voorspelling van een bloedtransfusie, die op zijn beurt weer invloed heeft op het overleven van de patiënt, is het belangrijk dat de recall zo hoog mogelijk is. De overige geteste algoritmen halen hier maximaal een precision van 35-50%.

Wanneer het aankomt op het interpreteren van het algoritme, is het RandomForest algoritme minder geschikt. Het RandomForest algoritme bestaat uit vele bomen met grote maximale diepten, dit maakt het interpreteren van dit algoritme erg lastig. Doordat er in het RandomForest algoritme ook een willekeurig element zit, is de uitkomst van de berekening van belangrijke features elke keer verschillend. Één conclusie kunnen we in ieder geval trekken: "De combinatie van veel van de beschikbare features zorgt voor de sterkste voorspelling". Wanneer het algoritme versimpeld wordt, om het interpreteren makkelijker te maken, dalen de precision en recall zodanig dat de voorspellingen onbetrouwbaar zijn. Ondanks dat er veel features gebruikt worden in het model van het RandomForest algoritme, is overfitting geen probleem, door de manier waarop het algoritme werkt.

5.1 Future Work

De berekeneningen die gedaan worden over de tijdseries features kunnen uitgebreid worden. Op dit moment kijken we naar een aantal vaste waarden voor de berekeningen. Hier bij zijn nog extra metrieken te bedenken. Er zou echter ook nog gekeken kunnen worden naar voorspellingen van hoe tijdseries lopen. Aan de hand daarvan kunnen voorspellingen gemaakt kunnen worden of een patiënt een bloedtransfusie nodig heeft.

In dit onderzoek is er ook niet volledig onderzocht wat er te bereiken valt op het gebied van missing values. Er kan bijvoorbeeld gekeken worden naar regressie. Hiermee kunnen de missing values van tijdserie variabelen ingevuld worden met waarden die dichter bij de realiteit liggen dan tot nu toe is gedaan.

Bibliografie

[1] https://www.labuitslag.nl/bloedgas/pco2/.

[2] https://nl.wikipedia.org/wiki/Lijst_van_laboratoriumbepalingen_in_bloed.

[3] https://nl.wikipedia.org/wiki/.

[4] http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html.

[5] http://scikit-learn.org/stable/modules/svm.html.

[6] http://www.astro.caltech.edu/ george/aybi199/Djorgovski_DMintro.pdf.

[7] https://www.lumc.nl/org/intensive-care/.

[8] Based on lecture: 1 cse 881: Data mining lecture 10: Support vector machines. http://slideplayer.com/slide/5237968/.

[9] R. Bellman. *Dynamic Programming*. Dover Books on Computer Science. Dover Publications, reprint edition edition, March 2003.

[10] C. Chang C. Hsu and C. Lin. A practical guide to support vector classification. April 2010.

[11] C. Chang and C. Lin. Libsvm: A library for support vector machines, March 2013.

[12] J.M.C. Goncalves et al. *Advances in Information Systems and Technologies*, volume 206 of *Advances in Intelligent Systems and Computing*, chapter Predict Sepsis Level in Intensive Medicine - Data Mining Approach, pages 201–211. Springer Berlin Heidelberg, 2013.

[13] I. H. W. E. Frank and M. A. Hall. *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, third edition edition, January 2011.

[14] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[15] Y. Freund and R.E. Schapire. *Machine Learning*, volume 37, pages 277–296. Springer (USA), 1999.

[16] K. K. Giuliano. Physiological monitoring for critically ill patients: testing a predictive model for the early detection of sepsis. *American Journal of Critical Care*, 16(2):122–130, 2007.

[17] J. C. Ho, C. H. Lee, and J. Ghosh. Septic shock prediction for patients with missing data. *ACM Transactions on Management Information Systems (TMIS)*, 5(1):1, 2014.

[18] T. K. Ho. Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, August 1995.

[19] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[20] S. Kim, W. Kim, and R. W. Park. A comparison of intensive care unit mortality prediction models through the use of data mining techniques. *Healthcare informatics research*, 17(4):232–243, 2011.

[21] R.A. Olshen L. Breiman, J.H. Friedman and C.J. Stone. *Classification and regression trees*. Wadsworth Statistics/Probability. Chapman and Hall/CRC, first edition edition, January 1984.

[22] D. Ernst P. Geurts and L. Wehenkel. Extremely randomized trees. Springer Science + Business media, Inc., March 2006.

[23] J.R. Quinlan. *Induction of Decision Trees*, pages 81 – 106. Kluwer Academic Publishers, 1986.

[24] L. Rokach and O. Maimom. *Datamining with Decision Trees, Theory and Applications*. Series in Machine Perception and Artificial intelligence. World Scientific Publishing Company, April 2008.

[25] R. E. Schapire. Explaining adaboost. In *Empirical Inference*, pages 37–52. Springer, 2013.

[26] R. Tibshirani T. Hastie and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer-Verlag New York, second edition edition, 2009.

[27] S. Wang, F. Wu, and B. Wang. Prediction of severe sepsis using svm model. In *Advances in computational biology*, pages 75–81. Springer, 2010.

[28] J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.