STAGEVERSLAG

# Voorspellingsmodel voor reizigersverdeling in de trein

*Auteur:*
Kwongyen Mok

Nederlandse Spoorwegen
Business Intelligence & Analytics
Laan van Puntenburg 100
3500 ER Utrecht
*Begeleiders:*
Kees Jong
Karen Slijkhuis
Margot Peters

VU Universiteit Amsterdam
Faculteit der Exacte Wetenschappen
De Boelelaan 1081a
1081 HV Amsterdam
*Begeleiders:*
Mark Hoogendoorn
Rob van der Mei

31 augustus 2016

# Voorwoord

Voor u ligt mijn scriptie die is geschreven ter afsluiting van de master Business Analytics aan de Vrije Universiteit van Amsterdam. Het afstudeeronderzoek is uitgevoerd bij de Nederlandse Spoorwegen van maart tot en met augustus 2016.

Ten eerste wil ik mijn begeleider bij NS, Kees Jong, bedanken voor zijn goede begeleiding tijdens de gehele stageperiode. Bij problemen tijdens het onderzoek kon Ik altijd hem om hulp vragen. Daarnaast wil ik Karen Slijkhuis bedanken voor het bedenken van dit interessant onderzoek dat ik heb mogen uitvoeren. Ook wil ik Margot Peters van Nedtrain bedanken voor de wekelijkse feedback op mijn onderzoek. Zij was altijd bereid om tijd vrij te maken wanneer ik vragen had.

Tot slot wil ik mijn begeleider op de Vrije Universiteit, Mark Hoogendoorn, wie mij ook heeft begeleid bij het schrijven van mijn research paper, bedanken voor zijn hulp en advies die hij mij heeft gegeven tijdens mijn stage. Mijn tweede lezer, Rob van der Mei, wil ik graag bedanken voor het lezen van mijn scriptie.

Kwongyen Mok

Augustus 2016

# Samenvatting

Om de klanttevredenheid te bevorderen wordt in dit onderzoek gekeken hoe reizigers op het perron geïnformeerd kunnen worden over waar het druk en rustig is in de trein. We kunnen bijvoorbeeld de reizigersverdeling in de trein voor aankomst en de reizigersverdeling bij vertrek aan de reizigers op het perron tonen. De reizigersverdeling in de trein voor aankomst kan bepaald worden aan de hand van sensoren die het gewicht van de trein meten. De reizigersverdeling bij vertrek moet echter voorspeld worden.

Om de reizigersverdeling in de trein te bepalen is gekozen voor Gotcha (meetpunten op het spoor). Met Gotcha kan in tegenstelling tot beladingssensoren in de trein het gewicht van alle bakken gemeten worden, terwijl bij beladingsensoren alleen het gewicht van de motorbakken kan worden gemeten. Daarnaast hebben de metingen van het gewicht tussen de lege bakken een sterk positieve correlatie, wat van belang is om de reizigersverdeling te bepalen. Vervolgens is de reizigersverdeling uit Gotcha vergeleken met de reizigersverdeling bepaald uit reizigerstellingen in de trein, waaruit blijkt dat Gotcha in staat is om de globale reizigersverdeling weer te kunnen geven.

In dit onderzoek wordt alleen naar de reizigersverdeling in de VIRM-VI gekeken. Daarnaast zijn de analyses gebaseerd op de metingen in 2015 op het traject Utrecht Centraal – Weert met als tussenstations ’s Hertogenbosch en Eindhoven.

De volgende machine learning algoritmes zijn gebruikt bij het voorspellen van de reizigersverdeling in de trein bij vertrek: Multivariate Adaptive Regression Splines (MARS), Neural Networks (NN) en Conditional Inference Trees (CIT). Met 5-fold cross-validation is NN significant beter dan MARS en CIT. Daarnaast is als benchmark het gemiddeld reizigerspercentage per bak van de vorige twee weken genomen bij hetzelfde treinnummer en dag van de week, waarmee aangetoond is dat de machine learning modellen betere voorspellingen leveren dan het benchmark. Ten slotte zijn enkele reizigersverdelingen die bepaald waren met Gotcha vergeleken met de voorspellingen van de modellen en het benchmark. Hieruit blijkt dat NN de globale reizigersverdeling van Gotcha het best kan voorspellen.

Om het voorspellingsmodel op basis van Gotcha gangbaar te maken, zouden meer Gotcha meetpunten aangelegd kunnen worden zodat de reizigersverdeling op meerdere stations voorspeld kan worden.

# Inhoud

# 1 Inleiding

## 1.1 Aanleiding

Voor NS is het leveren van goede service aan reizigers heel belangrijk. Soms is het voor reizigers lastig om een zitplaats te vinden wanneer het druk is in de trein. Reizigers kunnen vanuit het perron moeilijk zien waar lege plekken beschikbaar zijn in de trein. Wanneer NS reizigers van te voren kan laten weten waar lege zitplaatsen beschikbaar zijn, zal het mogelijk de klanttevredenheid bevorderen. Met deze informatie zullen reizigers zich efficiënter op het perron opstellen. Dit zorgt voor een kortere instaptijd van reizigers en zullen meer treinen beter op tijd kunnen vertrekken.
Om deze informatie te kunnen leveren is het niet alleen belangrijk te weten hoe druk het is in de trein, maar ook hoe reizigers binnen de trein zijn verdeeld. Een mogelijke oplossing is om infrarood sensoren boven alle deuren van de trein te installeren die het aantal reizigers in elk rijtuig van de trein bepalen. Echter is deze aanpak duur. Andere potentiële manieren om de verdeling in de trein te bepalen zijn onder andere het gebruik van WiFi in de trein. Hiermee kan het aantal reizigers worden geteld dat connectie maakt met WiFi in de trein. Daarnaast kan met behulp van gewichtssensoren in de trein de verdeling van reizigers worden afgeleid.

## 1.2 Doelstelling

Het doel van NS is om het aantal lege zitplaatsen per rijtuig te voorspellen en deze informatie te tonen aan reizigers die wachten op het perron. Op het perron van station ‘s-Hertogenbosch hangen lichtbalken waarmee de drukte in elk rijtuig kan worden weergegeven. Dit onderzoek zal echter alleen focussen op het voorspellen van de reizigersverdeling (reizigerspercentage per bak) en dus niet het aantal reizigers in de trein.

## 1.3 Onderzoeksvragen

Voordat de onderzoeksvragen worden benoemd, wordt eerst onderscheid gemaakt tussen reizigersverdelingen op 3 verschillende momenten.

1. De reizigersverdeling in de trein voor aankomst op station.
2. De reizigers verdeling nadat arriverende reizigers zijn uitgestapt, maar voordat vertrekkende reizigers zijn ingestapt.
3. De reizigersverdeling in de trein bij vertrek uit station.

Voor reizigers op het perron is het interessant te weten hoe druk het in elk rijtuig zal zijn nadat alle reizigers zijn uitgestapt. Echter is hiervan geen data beschikbaar, omdat reizigers vaak al instappen voordat alle andere reizigers zijn uitgestapt. Dus dit moment bestaat in de praktijk niet.

Bij het tonen van de reizigersverdeling in de trein voor aankomst kan het gebeuren dat veel reizigers in een druk rijtuig uitstappen, waardoor er meer plek is dan aangegeven. Wanneer de reizigersverdeling bij vertrek wordt getoond, kan het voorkomen dat een weergegeven druk rijtuig minder druk is bij het instappen, omdat het grootste deel van de reizigers in dit rijtuig op het huidig station is ingestapt. De reizigersverdeling voor aankomst is real-time te bepalen. De reizigersverdeling bij vertrek moet echter worden voorspeld en daarover zal dit onderzoek voornamelijk over gaan.

In dit onderzoek staan de volgende vragen centraal:

**Hoofdvraag:**
*Is het mogelijk om met machine learning technieken de reizigersverdeling bij vertrek uit een station te voorspellen?*

**Deelvragen:**

- *Welke factoren hebben invloed op de reizigersverdeling in de trein?*
- *Welke databronnen zijn beschikbaar en wat zijn daarvan de mogelijkheden en beperkingen?*
- *Is het mogelijk om de verdeling van reizigers in de trein te bepalen op basis van het gewicht van de trein en WiFi in de trein?*
- *Hoe kunnen we een voorspellingsmodel evalueren?*
- *Welke technieken kunnen worden toegepast worden om een voorspellingsmodel te maken?*

De wetenschappelijke uitdaging van dit onderzoek zit voornamelijk in het maken van een model dat meerdere variabelen tegelijk kan voorspellen. Hierdoor is er meer werk vereist in het selecteren van significante attributen. Daarnaast moet onderzocht worden wat een geschikte methode is om een model te evalueren.

## 1.4 Opbouw verslag

Dit verslag begint in hoofdstuk 2 met een korte literatuurstudie over eerdere onderzoeken die zijn gedaan met betrekking tot de verdeling van reizigers. In hoofdstuk 3 wordt de context van het probleem besproken, waaronder verschillende factoren die mogelijk invloed hebben op de reizigersverdeling in de trein. Daarna begint hoofdstuk 4 met de scope van het onderzoek, gevolgd door een beschrijving van de verschillende databronnen die beschikbaar zijn. De belangrijkste twee databronnen worden hier ook uitgebreid met elkaar vergeleken. In hoofdstuk 5 is de uiteindelijke dataset voor het trainen van de modellen weergegeven. Verder in dit hoofdstuk worden de verschillende machine learning algoritmes besproken en hoe deze modellen worden geëvalueerd. Dit hoofdstuk eindigt met het selecteren van de relevante attributen voor het voorspellingsmodel. In hoofdstuk 6 zijn de resultaten weergeven, waarna de voorspellingen van de modellen worden geëvalueerd. Tot slot eindigt dit verslag met een conclusie en enkele voorstellen voor verder onderzoek in hoofdstuk 7.

# 2 Literatuurstudie

Er is veel onderzoek gedaan naar hoe reizigers zich verdelen op perrons, zodat er voor kan worden gezorgd dat reizigers sneller kunnen in- en uitstappen. Sommige onderzoeken hebben aangetoond dat de ligging van de roltrappen veel invloed heeft op de positionering van reizigers op het perron [1] [2]. Data over het aantal reizigers dat bij elk metro of trein deur in- en uitstappen wordt meestal niet bijgehouden [3]. Voor het analyseren van de verdeling van reizigers op het perron moet daarom het aantal reizigers die in- en uitstappen via camerabeelden worden gehaald of door handmatig het aantal reizigers te tellen. De meeste datasets uit de onderzoeken zijn daarom vrij beperkt.

Een onderzoek over het verband van de halteertijd van de trein en de verdeling van de reizigers op het perron was uitgevoerd door Mori van de universiteit van Toronto[4]. Het onderzoek resulteerde in een simulatie model dat suggereert wanneer reizigers zich uniform over de trein deuren verdelen, de halteertijd 15 seconde korter zal zijn.

In het onderzoek van Liu [5] is gekeken wat de belangrijkste factoren zijn voor waar reizigers gaan staan op het perron. Vervolgens is een model geïmplementeerd dat voor elke reiziger voorspeld waar hij/zij gaat staan m.b.v. een zogenaamde utility function. Deze functie is onder andere gebaseerd op de afstand tussen de ingang van het perron en waar de reizigers op het perron staan. Het model wijst voor elke reiziger een plek toe op het perron, waarbij de utility function gemaximaliseerd wordt. Om deze functie te maximaliseren moet dus voorkomen worden dat reizigers te ver van de ingang van het perron worden toegewezen. Voor de evaluatie van het model is drie keer het aantal reizigers op het perron geteld tijdens de spits en drie keer buiten de spits. Uit de resultaten blijkt dat het model beter presteert buiten de spits.

In het onderzoek van Krstanoski [3] wordt ook gekeken naar hoe reizigers zijn verdeeld bij het in- en uitstappen op het noordelijk perron van metrostation Bloor in Toronto. Het perron heeft aan één kant een grote ingang. Met behulp van camerabeelden zijn binnen een periode van twee uur van 44 treinen bepaald hoeveel reizigers via elke metrodeur zijn in- en uitgestapt. Uit het onderzoek blijkt dat de meeste reizigers bij de deuren instappen die dichtbij de ingang van het perron stoppen. Hetzelfde geldt voor de reizigers die uitstappen, hoewel het verschil tussen het aantal reizigers per deur minder groot is. Verder zijn de verdelingen op het perron redelijk stabiel over de tijd. Dit gaf aanleiding tot het gebruik van kansverdelingen om de verdeling van reizigers op het perron te modelleren. In het onderzoek van Krstanoski wordt gebruik gemaakt van een multinonimale verdeling. Szplett en Wirasinghe [6] modelleerde de reizigers op een perron met een negatief exponentiële verdeling waar er maar één ingang/uitgang aan een kant van het perron ligt.

Tenslotte is er een start-up OpenCapacity[7], gevestigd in Londen, dat real-time voorspellingen maakt van de drukte in de trein op compartiment niveau. OpenCapacity maakt zowel gebruik van historische data als real-time data van het trein gewicht, WiFi, camera's, reizigers tellingen en deursensoren in combinatie met machine learning technieken om accuraat het aantal reizigers in de trein te bepalen. Het voorspellingsmodel van OpenCapacity voorspelt het aantal reizigers dat in elke treinbak zullen uitstappen, zodat wachtende reizigers op het perron van te voren weten waar meer zitplaatsen beschikbaar zullen zijn.

# 3 Context

Voordat we beginnen met de analyses is het handig om achtergrond informatie van het probleem weer te geven. In dit hoofdstuk wordt kort ingegaan op NS. Vervolgens worden enkele vaak voorkomende begrippen in dit verslag toegelicht. Mogelijke invloeden op de reizigersverdeling in de trein worden in kaart gebracht met uitleg over hoe de trein is samengesteld.
Tenslotte worden enkele opties voor het bepalen van de reizigersverdeling in de trein beschreven.

## 3.1 Nederlandse Spoorwegen

Nederlandse Spoorwegen (NS) is een Nederlands spoorwegbedrijf dat dagelijks de treinreis voor haar klanten verzorgt. Met ongeveer 4800 treinritten per dag over een spoorwegnet van rond 2100 km lang is NS de grootste reizigersvervoerder in Nederland [8]. Het spoorwegnet is echter in het beheer van Prorail.
Bij NS wordt veel data bijgehouden van treinen en reizigersstromen. Deze data wordt beheerd en geanalyseerd op de Business Intelligence afdeling (BI&A) van NS Reizigers (NSR). Met data analyse kunnen verschillende processen in het bedrijf verbeterd worden.

## 3.2 Definities

Hieronder volgen enkele definities die vaak voorkomen in deze scriptie.

**Treinstel**
Een treinstel is een vaste combinatie van rijtuigen.

**Treinbak/rijtuig/treincoupé**
Een treinbak is een onderdeel van een treinstel. Een treinstel bestaat uit meerdere bakken.

**Treinserie**
De treinserie geeft een specifieke route aan van een trein. Bijvoorbeeld de trein die van Schagen naar Maastricht rijdt en andersom is de treinserie 800.

**Treinnummer**
Het treinnummer duidt een specifieke trein van de treinserie aan. Het bevat naast informatie over de treinserie ook informatie over het tijdstip waarop de trein rijdt. Bijvoorbeeld de trein die om 08:05 vertrekt vanuit Schagen en richting Maastricht rijdt, is treinnummer 831. De volgende trein die om 08:35 vertrekt is treinnummer 833. Treinen die in tegengestelde richting rijden worden van elkaar onderscheiden met even en oneven treinnummers.

**Materieelnummer/Treinstelnummer**
Het treinstelnummer is het nummer van een specifiek treinstel. Het gaat hierbij dus om een nummer die fysiek materieel identificeert, niet zoals treinnummer en treinserie om logische nummers uit de dienstregeling.

**Traject**
Een traject is het gedeelte van de spoorweg tussen twee stations dat door een trein wordt afgelegd.

**Rit**
Een rit is het gedeelte van de spoorweg tussen begin- en eindstation van een treinserie dat door een trein wordt afgelegd.

## 3.3 Invloeden op de reizigersverdeling in de trein

De manier waarop reizigers in de trein zijn verdeeld kan verschillende oorzaken hebben. Hieronder worden alle mogelijke oorzaken in kaart gebracht.

### 3.3.1 Inrichting van stations

De verdeling in de trein is afhankelijk van waar reizigers zijn in- en uitgestapt in voorgaande stations. Eén van de factoren die invloed heeft op de reizigersverdeling is de ligging van de (rol)trappen op het perron, wat al eerder is genoemd in hoofdstuk 2. De meeste reizigers hebben de neiging om dichtbij de trappen te gaan staan, omdat zij niet ver willen lopen. Met name wanneer de trappen aan één kant van het perron liggen, is de kans klein dat reizigers naar de andere kant van het perron lopen om op de trein te wachten. Andersom komt het ook vaak voor dat reizigers in een bepaalde treinbak instappen, zodat zij op hun eindbestemming dichterbij de trappen uitstappen. Dit zijn meestal ervaren reizigers die regelmatig hetzelfde traject reizen. Wanneer de trein ergens anders stopt dan verwacht, zullen reizigers ook anders in de trein stappen. Verder zijn er nog allerlei andere voorzieningen op het perron die invloed kunnen hebben op de reizigersverdeling, bijvoorbeeld de ligging van de kiosk, drank- en snoepautomaten, informatieborden, rookgebieden en aanwezigheid van beschutting op het perron.

### 3.3.2 Inrichting van de trein

Sommige reizigers baseren hun keuze waar zij instappen op de kenmerken van de verschillende treinbakken. Reizigers die met de eerste klas reizen zullen meestal in de treinbakken stappen waar eerste klas zitplaatsen aanwezig zijn. Aan de andere kant zijn er tweede klas reizigers die doelbewust kiezen voor treinbakken zonder eerste klas zitplaatsen, omdat daar meer tweede klas zitplaatsen beschikbaar zijn en het dus vaak makkelijker is om een plek te vinden. Daarnaast zijn er reizigers die in specifieke rijtuigen stappen met fiets/rolstoel ingang. Ten slotte is het mogelijk dat reizigers een voorkeur hebben voor een stiltecoupé.

## 3.4 Trein samenstelling

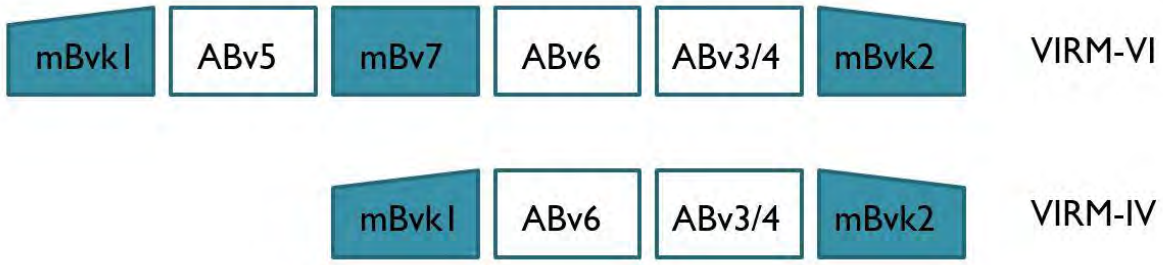
NS beschikt over verschillende type treinen. Deze kunnen we onderverdelen in sprinters gericht op korte trajecten en Intercity's voor lange trajecten [9]. In dit onderzoek wordt de reizigersverdeling bepaald voor een specifiek treintype: VIRM (Verlengd InterRegio Materieel), omdat dit treintype een dubbeldekstrein is dat voornamelijk als Intercity wordt ingezet en reizigers op het perron vaak moeilijker kunnen zien waar lege zitplaatsen zijn in dit type treinstel. Daarnaast is van dit treintype de benodigde data beschikbaar.

Een VIRM treinstel bestaat uit vier (VIRM-IV) of zes (VIRM-VI) bakken. NS heeft in totaal 178 VIRM treinstellen, waaronder 98 VIRM-IV en 80 VIRM-VI. Meerdere VIRM treinstellen kunnen aan elkaar gekoppeld worden. Een gekoppelde trein mag maximaal uit 12 bakken bestaan. Reizigers kunnen binnen een treinstel naar andere bakken lopen, maar niet naar een aansluitend treinstel.

De samenstelling van de bakken wordt beschreven met bakcodes. Vaak gebruikte afkortingen zijn hieronder toegelicht.

- A: Afdeling eerste klas
- B: Afdeling tweede klas
- m: rijtuig met aandrijving / motorbak
- k: koprijtuig van een treinstel
- v: dubbeldeksrijtuig

VIRM-IV treinstellen hebben de volgende samenstelling: mBvk1 + ABv6 + ABv3/4 + mBvk2, waarbij de voorste en achterste bak aangedreven worden door motoren. De getallen in de bakcodes geven de positie van de bakken in het treinstel aan. De samenstelling van een VIRM-VI treinstel ziet er als volgt uit: mBvk1 + ABv5 + mBv7 + ABv6 + ABv3/4 + mBvk2. Een VIRM trein met zes bakken heeft dus een extra motorbak. Deze twee trein samenstellingen zijn nogmaals visueel weergegeven in Figuur 1.




Figuur 1: Treinsamenstelling VIRM-VI en VIRM-IV

## 3.5 WiFi in de trein

In de Intercitytreinen van NS is gratis WiFi beschikbaar voor reizigers. De VIRM heeft in elke treinbak een acces point waar de reiziger connectie mee kan maken. Er zijn gegevens beschikbaar over het aantal actieve WiFi-sessies (gebruiker heeft voorwaarden geaccepteerd) op elk traject. Helaas is deze data geaggregeerd over het hele treinstel en dus is het aantal actieve WiFi-sessies per treinbak onbekend. De WiFi data kan daarom op dit moment niet gebruikt worden bij het bepalen van de reizigersverdeling in de trein, behalve bij treinen die uit 2 treinstellen bestaan. In die gevallen zou een globale voor/achter verdeling kunnen worden bepaald.

**Verdere potentie WiFi (momenteel niet beschikbaar)**
Het WiFi protocol schrijft voor dat een WiFi apparaat periodiek een signaal verstuurt waarin het MQAC adres bekend wordt gemaakt. De frequentie kan per apparaat verschillen. Als tussen twee stations het aantal unieke MAC adressen wordt geteld (buiten de stations, dus geen mensen die wachten op de perrons), zou mogelijk een vollediger beeld van het aantal reizigers ontstaan, zonder grote privacy bezwaren. Daarbij moet wel opgemerkt worden dat sommige reizigers uiteraard geen WiFi apparaat hebben of WiFi uitgeschakeld hebben. Ook zouden er reizigers kunnen zijn met meerdere WiFi apparaten.

## 3.6 Gewicht van de trein

Om de verdeling van reizigers in de trein te bepalen, kunnen we gebruik maken van het gewicht van de afzonderlijke treinbakken inclusief reizigers. Het gewicht kan op twee manieren worden gemeten, namelijk met beladingssensoren in de trein en meetpunten langs het spoor (Gotcha). Beide worden in hoofdstuk 4 verder beschreven.

In beide gevallen moet goed rekening gehouden worden met het gewicht van de bakken dat per baktype verschilt. Er zijn bijvoorbeeld bakken met een motor (mBvk1, mBv7 en mBvk2) of een stroomafnemer (ABv6). Alleen de mBvk1 en mBvk2 zijn qua constructie hetzelfde. Andere factoren die ook invloed hebben op het gewicht van de bakken zijn wielslijtage en het waterreservoir van de toiletten. In tegenstelling tot het gewicht van de motor en stroomafnemer verandert het gewicht van de wielen en het waterreservoir. Een wiel kan door slijtage 100 kg afnemen voordat het wordt vervangen. Dus een treinbak kan hierdoor (8x100=) 800 kg minder wegen. De toiletten met waterreservoir bevinden zich in de ABv5 en ABv6. In elk waterreservoir kan 400 kg water opgeslagen worden en loopt gedurende dag leeg.

# 4 Vooronderzoek

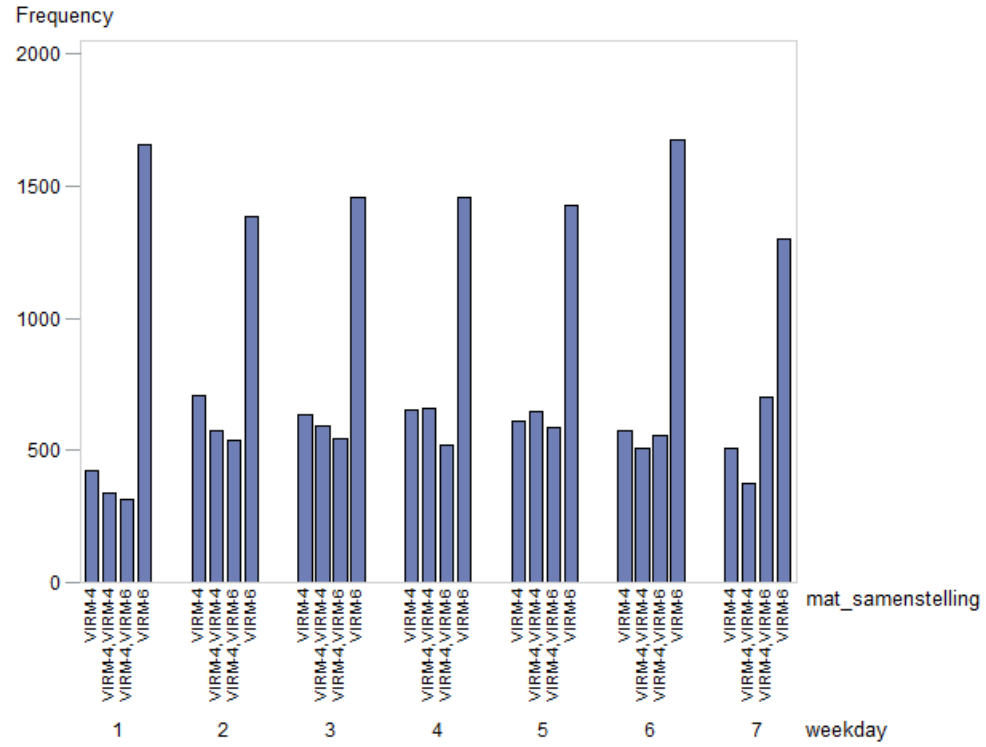
In dit hoofdstuk wordt eerst de scope van het onderzoek vastgesteld. Daarna wordt dieper ingegaan op de beschikbare databronnen. De beladingssensor data en Gotcha data worden geanalyseerd en met elkaar vergeleken om de meest geschikte databron voor het bepalen van de reizigersverdeling in de trein te selecteren.

## 4.1 Scope

In deze paragraaf wordt de setup van het onderzoek besproken. Om een voorspellingsmodel toe te kunnen passen moet de reizigersverdeling in de trein voor aankomst en bij vertrek bepaald kunnen worden. Hiervoor moeten er Gotcha meetpunten aanwezig zijn zowel voor als na het station waarop we de reizigersverdeling willen bepalen. Daarom is voor het onderzoek gekozen voor het traject Utrecht Centraal – Weert. Op dit traject rijden VIRM-treinen van treinserie 800 die ook in 's-Hertogenbosch en Eindhoven stoppen. Tussen alle stations op dit traject is een Gotcha meetpunt aanwezig.

We maken gebruik van de beladingssensor data en Gotcha data in 2015.

Daarna is er een keuze gemaakt voor een bepaalde treinsamenstelling om de reizigersverdeling ervan te bepalen. Hiervoor hebben we gekeken hoe vaak de verschillende treinsamenstellingen voorkwamen op het traject Utrecht Centraal - Weert. Dit is weergegeven in Figuur 2.




Figuur 2: Aantal uitgevoerde ritten per treinsamenstelling en dag van de week

Hier zijn voor 4 treinsamenstellingen het aantal uitgevoerde ritten in 2015 weergeven. De overige 2 treinsamenstellingen zijn weggelaten (3 VIRM-IV treinstellen en 2 VIRM-VI treinstellen), omdat die nauwelijks voorkwamen. Daarnaast zijn de aantallen per dag van de week gegroepeerd. Het valt meteen op dat de treinsamenstelling van een enkel VIRM-VI het vaakst voorkwam; bij vrijwel alle dagen van de week komt deze treinsamenstelling ongeveer 4 keer zo vaak als andere

treinsamenstellingen. Daarom wordt in dit onderzoek alleen data van enkele VIRM-VI treinen meegenomen.

## 4.2 Databronnen

NS beschikt over veel data. In deze paragraaf worden verschillende databronnen die nodig zijn voor dit onderzoek beschreven. Daarnaast wordt alvast voor de beladingssensor data en de Gotcha data een korte analyse uitgevoerd.

### 4.2.1 Treinactiviteiten

De dataset 'Treinactiviteiten' is nodig om de koppeling te kunnen maken tussen metingen en de dienstregeling. Deze dataset bevat informatie over vertrek- en aankomsttijden van treinen en bestaat uit 36 kolommen. Uit de dataset zijn alleen de benodigde kolommen geselecteerd en een gedeelte van de dataset is weergegeven in Tabel 1. Elke regel in de dataset geeft aan of het een vertrek uit of aankomst bij een station betreft. De naam van het station is aangegeven in de kolom TA_DRGLPT. Alleen de afkorting van de naam wordt hier weergegeven. In elke regel wordt aangegeven op wat tijdstip een bepaalde trein aankomt bij een station of wanneer deze vertrekt uit een station (TA_UITVTIJD_DT). Daarnaast is de geplande aankomst-/vertrektijd weergegeven (TA_APLAN_DT). Door het verschil te nemen met de werkelijke aankomst-/vertrektijd is ook het aantal minuten vertraging berekend (TA_APLAN_VERSCHILTIJD). Behalve aankomst- en vertrektijden worden ook doorkomsttijden bijgehouden. Dit zijn tijdstippen waarop treinen een bepaald station passeren en is in kolom TA_ACT_SRT met 'D' aangegeven. In de kolom TA_VOLGNR zijn de stations bij elk treinnummer op volgorde van passeren genummerd. Tenslotte is er ook informatie over de vervoerder (TA_VERVOERDER) en de richting waarop de trein rijdt (TA_RICHTING).

| | TA_VERKEERSDATUM_D | TA_TREINSERIE_N | TA_TREINNR_N | TA_RICHTING | TA_VOLGNR | TA_DRGLPT | TA_ACT_SRT | TA_APLAN_DT | TA_UITVTIJD_DT | TA_APLAN_VERSCHILTIJD | TA_VERVOERDER |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 13DEC2015 | 800 | 821 | O | 14 | Std | A | 13DEC15:08:47:00 | 13DEC15:08:47:11 | 0 | NSR |
| 15 | 13DEC2015 | 800 | 821 | O | 15 | Std | V | 13DEC15:08:50:00 | 13DEC15:08:50:17 | 0 | NSR |
| 16 | 13DEC2015 | 800 | 821 | O | 16 | Luta | D | 13DEC15:08:52:00 | 13DEC15:08:52:40 | 0 | NSR |
| 17 | 13DEC2015 | 800 | 821 | O | 17 | Lut | D | 13DEC15:08:53:00 | 13DEC15:08:53:33 | 0 | NSR |
| 18 | 13DEC2015 | 800 | 821 | O | 18 | Bk | D | 13DEC15:08:55:00 | 13DEC15:08:55:50 | 0 | NSR |
| 19 | 13DEC2015 | 800 | 821 | O | 19 | Bde | D | 13DEC15:08:59:00 | 13DEC15:09:00:31 | 1 | NSR |
| 20 | 13DEC2015 | 800 | 821 | O | 20 | Bha | D | 13DEC15:09:01:00 | 13DEC15:09:02:01 | 1 | NSR |
| 21 | 13DEC2015 | 800 | 821 | O | 21 | Mt | A | 13DEC15:09:04:00 | 13DEC15:09:04:44 | 0 | NSR |
| 22 | 13DEC2015 | 800 | 822 | E | 1 | Ehv | V | 13DEC15:07:31:00 | 13DEC15:07:31:22 | 0 | NSR |
| 23 | 13DEC2015 | 800 | 822 | E | 3 | At | D | 13DEC15:07:35:00 | 13DEC15:07:35:41 | 0 | NSR |
| 24 | 13DEC2015 | 800 | 822 | E | 4 | Bet | D | 13DEC15:07:37:00 | 13DEC15:07:37:36 | 0 | NSR |
| 25 | 13DEC2015 | 800 | 822 | E | 5 | Beto | D | 13DEC15:07:38:00 | 13DEC15:07:38:00 | 0 | NSR |
| 26 | 13DEC2015 | 800 | 822 | E | 6 | Lpe | D | 13DEC15:07:40:00 | 13DEC15:07:41:04 | 1 | NSR |
| 27 | 13DEC2015 | 800 | 822 | E | 7 | Btl | D | 13DEC15:07:42:00 | 13DEC15:07:42:14 | 0 | NSR |
| 28 | 13DEC2015 | 800 | 822 | E | 8 | Vg | D | 13DEC15:07:47:00 | 13DEC15:07:46:59 | 0 | NSR |
| 29 | 13DEC2015 | 800 | 822 | E | 9 | Vga | D | 13DEC15:07:47:00 | 13DEC15:07:48:25 | 1 | NSR |
| 30 | 13DEC2015 | 800 | 822 | E | 10 | Ht | A | 13DEC15:07:50:00 | 13DEC15:07:50:21 | 0 | NSR |
| 31 | 13DEC2015 | 800 | 822 | E | 11 | Ht | V | 13DEC15:07:53:00 | 13DEC15:07:54:02 | 1 | NSR |

**Tabel 1: Treinactiviteiten**

### 4.2.2 Materieelplanning

In de dataset 'Treinactiviteiten' staat niet weergegeven welke treinstellen zijn gebruikt in de dienstregeling. Dit is wel terug te vinden in de dataset 'Materieelplanning'. Ook deze dataset bestaat uit 36 kolommen (niet dezelfde attributen als Treinactiviteiten) en een deel hiervan met de benodigde attributen is weergegeven in Tabel 2. Hierin zijn treinnummers (MU_TREINNR_N) aan materieelnummers (MU_MATNR) gekoppeld. Door de treinnummers in 'Materieelplanning' te koppelen met de treinnummers in 'Treinactiviteiten', weten we op elk traject welk materieel heeft gereden. Voor VIRM-treinen weten we dan of er een VIRM-IV of een VIRM-VI heeft gereden

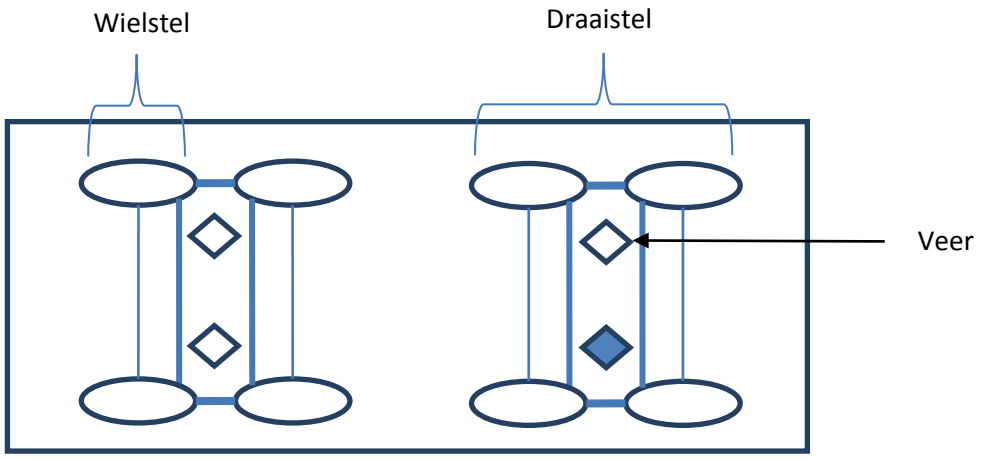
(MU_MATNR_SOORT en MU_MATNR_AANDUIDING) en of er sprake was van een gekoppelde trein. De treinen op regel 36 en 50 van Tabel 2 bijvoorbeeld zijn aan elkaar gekoppeld op het traject van Maastricht (Mt) naar Eindhoven (Ehv) en vormen samen treinnummer 876. De dataset 'Materieelplanning' wordt voornamelijk gebruikt om de dataset 'Treinactiviteiten' met de beladingssensor data te kunnen koppelen, omdat in de beladingssensor data geen treinnummers staan.

| | MU_DIENST DATUM_D | MU_TREIN NR_N | MU_B_DR GLPT | MU_E_DR GLPT | MU_MATNR | MU_MATNR _SOORT | MU_MATNR_ AANDUIDING |
|---|---|---|---|---|---|---|---|
| 36 | 13DEC2015 | 876 | Mt | Ehv | 9401 | VIRM | 4 |
| 37 | 13DEC2015 | 875 | Ehv | Mt | 9401 | VIRM | 4 |
| 38 | 13DEC2015 | 888 | Mt | Ehv | 9401 | VIRM | 4 |
| 39 | 13DEC2015 | 842 | Mt | Asd | 8736 | VIRM | 6 |
| 40 | 13DEC2015 | 851 | Asd | Mt | 8736 | VIRM | 6 |
| 41 | 13DEC2015 | 864 | Mt | Asd | 8736 | VIRM | 6 |
| 42 | 13DEC2015 | 873 | Asd | Ehv | 8736 | VIRM | 6 |
| 43 | 13DEC2015 | 873 | Ehv | Mt | 8736 | VIRM | 6 |
| 44 | 13DEC2015 | 886 | Mt | Ehv | 8736 | VIRM | 6 |
| 45 | 13DEC2015 | 886 | Ehv | Asd | 8736 | VIRM | 6 |
| 46 | 13DEC2015 | 832 | Mt | Asd | 8649 | VIRM | 6 |
| 47 | 13DEC2015 | 841 | Asd | Mt | 8649 | VIRM | 6 |
| 48 | 13DEC2015 | 854 | Mt | Asd | 8649 | VIRM | 6 |
| 49 | 13DEC2015 | 863 | Asd | Mt | 8649 | VIRM | 6 |
| 50 | 13DEC2015 | 876 | Mt | Ehv | 8649 | VIRM | 6 |

**Tabel 2: Materieelplanning**

### 4.2.3 Beladingsensor data

De VIRM trein beschikt over 114 sensoren die de toestand van de tractie-installatie monitoren. Deze metingen worden gemiddeld elke 3 seconden uitgevoerd. De sensoren die interessant zijn voor dit onderzoek zijn de beladingssensoren die het gewicht van een treinbak meet. Deze bevinden zich boven draaistellen waar de treinbak op rust. Op die plekken zitten veren die aan het gewicht van de treinbak aangepast kunnen worden. De beladingssensoren waarvan de data wordt opgehaald, zitten alleen op draaistellen van motorbakken. In Figuur 3 is een onderaanzicht van een treinbak weergegeven.




**Figuur 3: Onderaanzicht VIRM motorbak**

Elke treinbak heeft acht wielen, verdeeld over twee draaistellen. Op elk draaistel liggen twee grote veren naast elkaar. Bij een motorbak is één van de twee draaistellen aangedreven door een motor. De beladingssensor zit op één van de twee veren van een aangedreven wielstel. In Figuur 3 is de locatie van de beladingssensor met een blauw gevulde ruit aangegeven. Grofweg meet de beladingssensor dus een kwart van het totaal gewicht van de bak. Bij een motorbak is het gewicht op het aangedreven draaistel zwaarder dan het draaistel zonder motor. Dus de druk op de veren bij dit draaistel is groter dan op het ander draaistel, met als gevolg dat men niet de metingen simpelweg met 4 kan vermenigvuldigen om het totaal gewicht van de bak te krijgen. In Tabel 3 staan de theoretische veerlasten van de motorbakken bij een lege VIRM trein [10]. De mBvk1 en mBvk2 zijn qua constructie hetzelfde, dus deze wegen evenveel. Om het juiste totaal gewicht van een treinbak te bepalen moet het gewicht op de overige drie veren zo berekend worden dat de verhouding van de gewichten overeenkomen met de verhouding van de theoretische veerlasten.

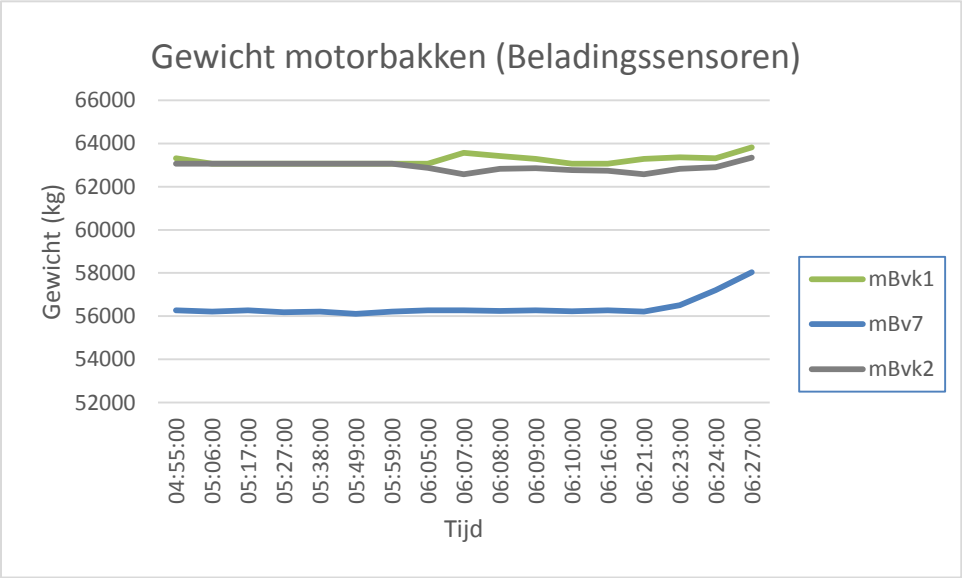
| Theoretische veerlasten (in tonnen) | | | | | |
|---|---|---|---|---|---|
| | Draaistel zonder motor | | Draaistel met motor | | Totaal gewicht |
| mBvk1/2 | 15,1 | 14,9 | 17 | 17 | 64 |
| mBv7 | 14,8 | 14,7 | 16,3 | 16,3 | 62,1 |

Tabel 3: Theoretische veerlasten per veer

Belangrijk om te noemen is dat bij de beladingssensor metingen die voor het onderzoek beschikbaar waren al het gewicht van de wielen is opgeteld. De beladingssensoren meten alleen het gewicht bovenop de wielen, maar in de beladingssensor data is bij de metingen een theoretisch gewicht van de wielen opgeteld. Tenslotte is de beladingssensor data alleen beschikbaar voor 56 VIRM treinstellen, waaronder 16 VIRM-VI en 40 VIRM-IV treinstellen.

**Korte analyse**

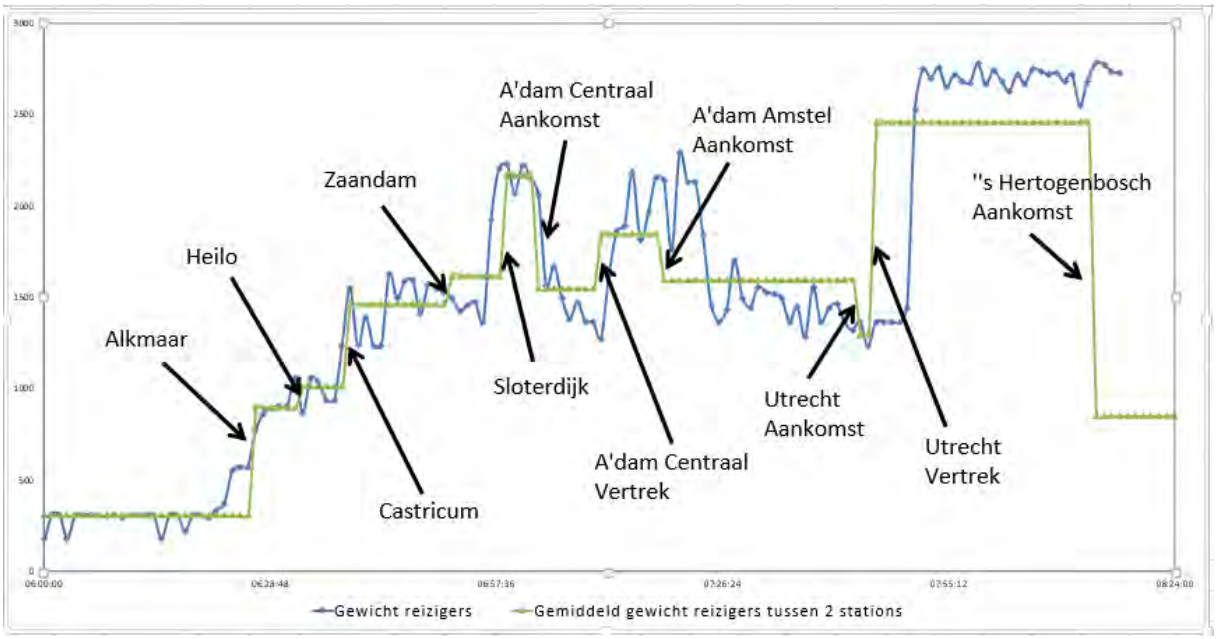
Om alvast enig inzicht te krijgen in deze data, bekijken we een kleine dataset van een VIRM-VI met treinstelnummer 8608 op 2 februari 2015. De metingen in deze dataset zijn naar minuten geaggregeerd.



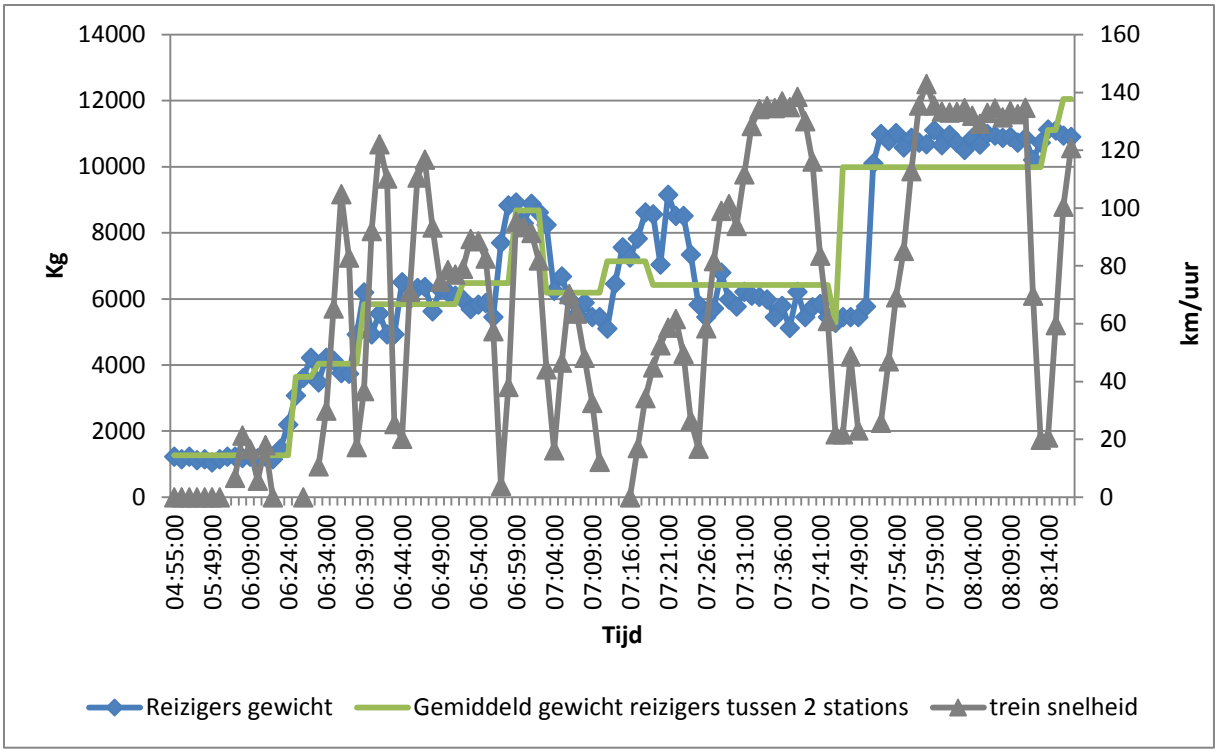

Figuur 4: Gewicht motorbakken van VIRM-VI (Beladingssensor)

In Figuur 4 zijn de metingen van de beladingssensoren in de drie motorbakken over de tijd geplot. Volgens de dataset 'Treinactiviteiten' begint de dienstregeling van deze trein om 06:27 in Alkmaar. We focussen ons nu eerst op de metingen vóór dit tijdstip om het gewicht van de bakken zonder reizigers te bepalen. De metingen tijdens de dienstregeling worden verderop weergegeven. Uit de grafiek zien we dat het gewicht van de bakken al toeneemt voor 06:27, met name de mBv7 bak. Daarnaast valt het op dat het gewicht van de mBv7 bak ongeveer 6000 kg lichter is dan het gewicht gegeven in Tabel 3. In paragraaf 4.3.1 wordt de oorzaak hiervan onderzocht. Naast de beladingssensoren zijn er ook sensoren die de snelheid van de trein meten. Deze data behoort tot dezelfde dataset als de beladingssensoren. Uit de data blijkt dat deze trein tussen 06:07 en 06:16 heeft gereden. Op dit tijdsinterval waren er nog geen reizigers in de trein. Toch is uit de grafiek te zien dat het gewicht van de trein varieert, wanneer hij aan het rijden is.

Vervolgens focussen we op de mBv7 bak en bekijken we hoe de metingen veranderen op een later tijdstip. In Figuur 5 is het gewicht van de reizigers in de mBv7 bak over de tijd weergegeven. Hierbij is het minimum gemeten gewicht op die dag afgetrokken. Daarnaast is ook het gemiddeld gewicht van de reizigers in de trein tussen twee stations gegeven. Dezelfde lijn geeft dus tegelijk de vertrek- en aankomsttijden op een station. Wat hier opvalt, is dat de beladingssensor data niet synchroon loopt met de vertrek- en aankomst tijden van de trein. Bijvoorbeeld, het gewicht van de reizigers in de treinbak neemt pas toe nadat de trein 4 minuten geleden was vertrokken vanuit Utrecht. Het berekend gemiddeld gewicht van de reizigers in de mBv7 bak van Utrecht naar 's-Hertogenbosch bevat dan ook enkele metingen van het vorig traject van Amsterdam Amstel naar Utrecht.
In Figuur 6 is ook de snelheid van de trein toegevoegd. Ook de snelheid van de trein komt niet overeen met de vertrek- en aankomst tijden uit 'Treinactiviteiten'. Bijvoorbeeld, volgens Treinactiviteiten staat de trein stil van 07:03 tot 07:11 op Amsterdam Centraal, terwijl de gemeten snelheid van de trein op dat tijdsinterval niet gelijk is aan nul, maar pas enkele minuten later.

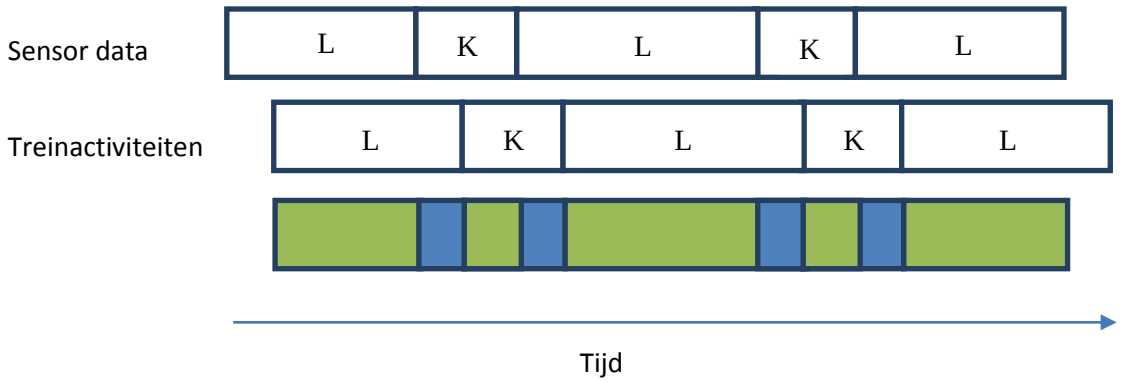



Figuur 5: Reizigersgewicht in mBv7 (Beladingssensor)




Figuur 6: Reizigersgewicht in mBv7 met treinsnelheid (Beladingssensor)

De reden dat de tijden uit de sensor data niet overeenkomen met de vertrek- en aankomsttijden uit 'Treinactiviteiten' is dat de sensoren gebruik maken van een interne klok in de trein. De tijden uit de sensor data verschillen dus ook per treinstel. Het verschil tussen de tijden uit de sensor data en de tijden uit 'Treinactiviteiten' is hooguit 15 minuten. Bij treinstelnummer 8608 loopt de tijd van de sensoren 6 minuten voor op de tijd uit 'Treinactiviteiten. Het probleem dat hieruit ontstaat, is dat metingen soms niet aan het juiste traject worden toegewezen. Bij korte trajecten is de kans groot dat er relatief meer metingen uit een andere traject hieraan worden toegewezen. Een voorbeeld is gegeven in Figuur 7 met drie lange en twee korte trajecten, welke met 'L' en 'K' respectievelijk zijn weergegeven.




Figuur 7: Tijdsverschil sensor data met treinactiviteiten

De tijdsintervallen waarin de metingen in het juiste traject vallen, zijn met groen weergegeven. De perioden waarin de metingen niet overeenkomen met het traject zijn blauw gekleurd. Bij lange trajecten is het percentage tijd met juiste metingen groter dan bij korte trajecten.
Met behulp van de snelheid sensoren kunnen tijdsintervallen worden gemaakt waarin de trein aan het rijden is. Een nieuw tijdsinterval begint dus wanneer de trein stil stond en vervolgens weer

vertrekt. Daarna worden deze tijdsintervallen gekoppeld aan de trajecten met de meeste overlap. Soms komt het voor dat verschillende tijdsintervallen aan hetzelfde traject worden gekoppeld. In dat geval wordt het tijdsinterval met de meeste overlap toegewezen aan dat traject. De sensor metingen in dat tijdsinterval worden tenslotte daaraan gekoppeld. Dit algoritme is hieronder weergegeven.

**Input:** Groep metingen $S$
Meting $s_i = \{z_i, v_i, m_i\} \in S$ $i \in [1, \dots, n]$,
$z =$ Tijdstip van meting,
$v =$ Snelheid in km/u,
$m =$ Gewicht in kg
Groep trajecten $T$
Traject $t_k = \{b_k, e_k\} \in T$ $k \in [1, \dots, m]$,
$b =$ Tijdstip van vertrek uit vorig station,
$e =$ Tijdstip van aankomst op huidig station
Tijdsinterval $x_j = \emptyset$ $j \in [1, \dots, \infty)$

**Output:** Koppeling metingen aan alle trajecten $R$
Koppeling metingen aan traject $r_l = \{t_l, x_{best}\} \in R$ $l \in [1, \dots, m]$

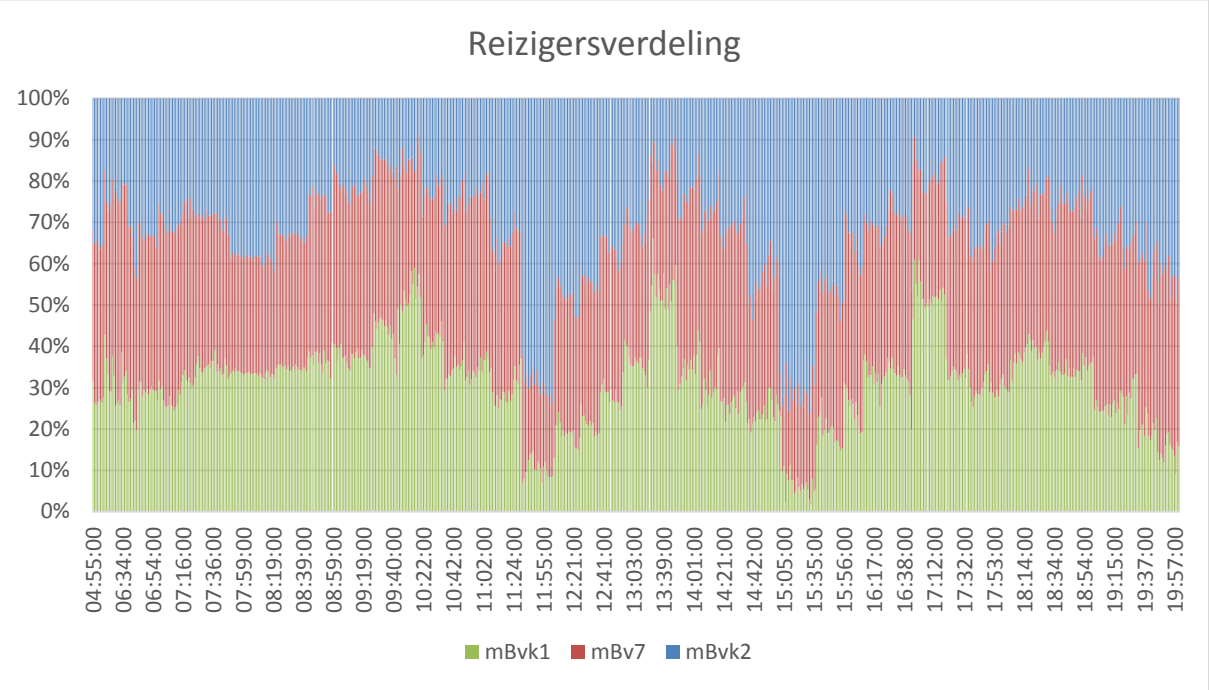
Tijdsintervallen creëren
$j = 1;$
**for each** $s_i \in S$ **do**
**if** $v_i > 0$ **then**
$x_j = \{x_j, s_i\};$
**if** $v_{i+1} = 0$ **then**
$j = j + 1;$
**end**
**end**
**end**

$X = \{x_1, \dots, x_j\};$

Tijdsintervallen toewijzen aan trajecten met grootste overlap in tijd
$l = 1;$
**for each** $t_k \in T$ **do**
$O = 0;$
$x_{best} = \emptyset;$
**for each** $x \in X$ **do**
**if** $\min(e_k, \max_x z) - \max(b_k, \min_x z) > O$ **then**
$O = \min(e_k, \max_x z) - \max(b_k, \min_x z);$
$x_{best} = x;$
**end**
**end**
$r_l = \{t_k, x_{best}\};$
$l = l + 1;$
**end**

In Figuur 8 is de verdeling van het reizigersgewicht over de drie motorbakken van hetzelfde treinstel weergegeven. Tijdens de spits is het gewicht van de reizigers gelijkmatiger verdeeld over de motorbakken, mogelijk omdat de trein dan meestal al redelijk vol zit. De mBvk1/2 en mBv7 hebben

ongeveer evenveel zitplaatsen. Als we kijken rond 11:48 en 15:25, zien we dat veel meer reizigers in de mBvk2 zitten. Rond 13:37 zitten de reizigers juist meer in de mBvk1. Uit deze grafiek zien we de mogelijkheid om met beladingssensoren de verdeling van de reizigers te bepalen.




**Figuur 8: Reizigersverdeling in de motorbakken**

### 4.2.4 Meetpunten op het spoor (Gotcha)

Een andere manier om het gewicht van de trein te achterhalen, is door het meten van de spoordoorbuiging veroorzaakt door de trein. Dit wordt gedaan met behulp van meetpunten op het spoor. Er zijn 44 meetpunten in het land waar op deze manier langs het spoor wordt gemeten. In Figuur 9 zijn de meetpunten rond het traject Utrecht Centraal – Weert weergegeven [11]. De meetpunten die op dit traject liggen zijn de meetpunten met nummer 110, 270 en 250. Bij een meetpunt wordt de spanning van elk wiel zes keer achter elkaar gemeten, waarbij het gemiddelde en maximum wordt gelogd. Deze worden daarna d.m.v. kalibratie omgezet naar het gewicht van elk wiel op het spoor. De gewichten zijn inclusief het gewicht van de wielen. De metingen staan vastgelegd in de Gotcha-data van Prorail. Dit meetsysteem wordt voornamelijk gebruikt om wieldefecten te signaleren en schade aan het spoor te voorkomen.


Figuur 9: Gotcha meetpunten rond traject Utrecht Centraal - Weert

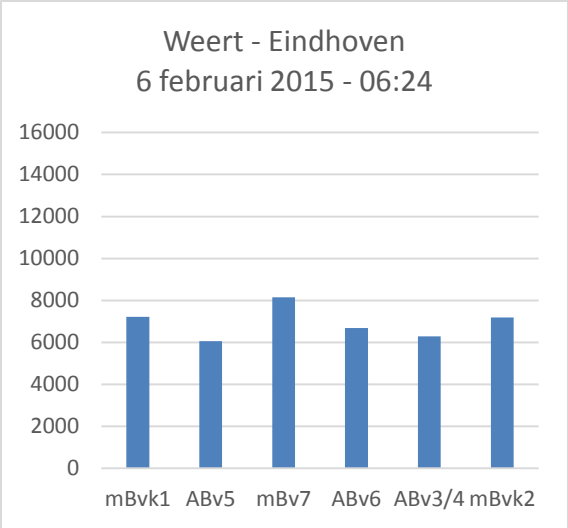
Elke trein heeft een RFID-tag waarmee het treinstel kan worden geïdentificeerd wanneer deze over een meetpunt rijdt. Een lijst wordt bijgehouden met de koppeling van de RFID-tagnummers aan de treinstelnummers. Deze lijst moet regelmatig worden bijgewerkt, wanneer RFID-tags vervangen worden. VIRM treinen worden op bak-niveau getagd. Met behulp van de tag-lijst kan het bijhorend treinstelnummer en baktype worden opgezocht. Vervolgens kan het baktype van de bakken zonder RFID-tag worden afgeleid met kennis van de VIRM treinsamenstelling. Het kan gebeuren dat oude tags niet meer voorkomen in de tag-lijst. In dat geval kan het treinstelnummer niet gevonden worden bij bepaalde treinpassages en kunnen deze niet voor analyse gebruikt worden.

**Korte analyse**

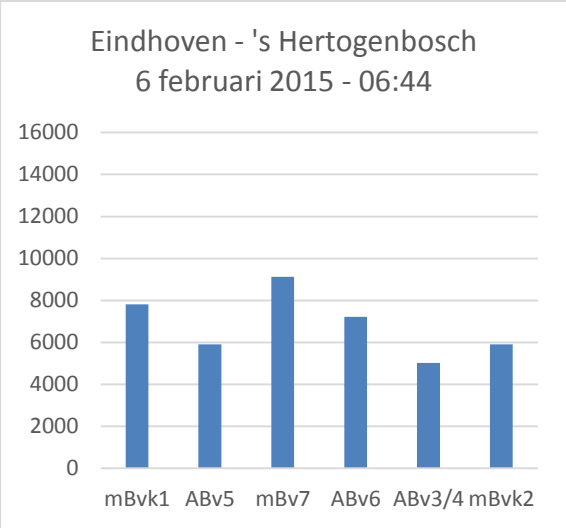
Om alvast enig inzicht te krijgen in deze data, bekijken we de metingen van treinstelnummer 8068 op 6 februari 2015. Deze heeft als treinnummer 818 op het traject van Weert naar Utrecht Centraal gereden. Tussen deze stations liggen drie Gotcha meetpunten en de trein stopt op dit traject ook in Eindhoven en ’s Hertogenbosch. In de onderstaande figuren is het gemeten reizigersgewicht in elke bak per Gotcha meetpunt weergegeven. Het reizigersgewicht is bepaald door het theoretisch gewicht van de bakken af te trekken van de Gotcha metingen. Het theoretisch gewicht van elke bak is in Tabel 4 weergegeven.

| mBvk1 | ABv5 | mBv7 | ABv6 | ABv3/4 | mBvk2 |
|---|---|---|---|---|---|
| 64.029 kg | 51.334 kg | 62.141 kg | 53.465 kg | 52.214 kg | 64.029 kg |

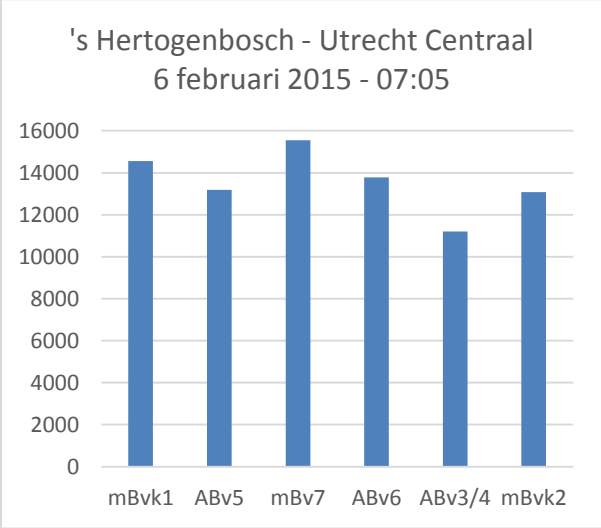
Tabel 4: Theoretisch gewicht VIRM-VI bakken




Figuur 10: Reizigersverdeling Weert - Eindhoven (Gotcha)




Figuur 11: Reizigersverdeling Eindhoven - 's Hertogenbosch (Gotcha)




Figuur 12: Reizigersverdeling 's Hertogenbosch - Utrecht Centraal (Gotcha)

Uit de figuren valt als eerste op dat het reizigersgewicht in elke bak op het traject ’s Hertogenbosch – Utrecht Centraal twee keer zo groot is als op het traject Weert – Eindhoven en Eindhoven – ’s Hertogenbosch. Verder zien we in Figuur 11 en Figuur 12 dat in de eerste drie bakken (mBvk1, ABv5 en mBv7) in totaal meer reizigers zitten dan in de bakken aan de andere kant van de trein (ABv6, ABv3/4 en mBvk2). Ook hier zien we dat het mogelijk zou zijn om de reizigersverdeling te bepalen met Gotcha.

## 4.3 Vergelijking Gotcha met beladingssensor

Voordat een voorspellingsmodel geïmplementeerd kan worden, moet een geschikte databron geselecteerd worden dat de reizigersverdeling in de trein kan bepalen. In paragraaf 4.2.3 en 4.2.4 zijn korte analyses uitgevoerd op de beladingssensor data en Gotcha data. In deze paragraaf wordt verder onderzocht hoe nauwkeurig de beladingssensoren zijn t.o.v. Gotcha in het bepalen van de reizigersverdeling in de trein.

Een groot nadeel van de beladingssensor data is dat alleen het gewicht van de reizigers in de motorbakken bepaald kan worden. In een VIRM-VI treinstel zijn dit de bakken: mBvk1, mBv7 en mBvk2. Het gewicht van de motorbakken moet bij beide datasets afgetrokken worden om het gewicht van de reizigers in elk motorbak te verkrijgen. De manier waarop het leeg gewicht van de motorbakken wordt bepaald is hieronder verder omschreven.

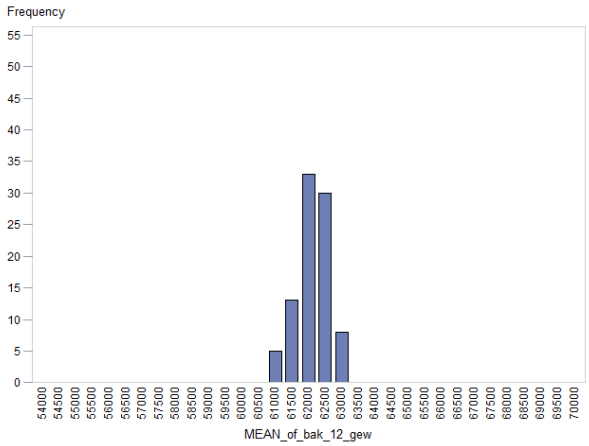
Door de lege ritten voor zowel de Gotcha als de beladingssensor data te analyseren, kunnen we de nauwkeurigheid van de metingen uit de Gotcha data en de beladingssensoren bepalen en met elkaar vergelijken. Daarna wordt onderzocht in hoeverre het gewicht van de reizigers dat bepaald is uit de beladingssensoren verschilt met die van Gotcha. Ook wordt gekeken of het berekend gewicht van de reizigers het aantal reizigers in de trein kan representeren. Het aantal reizigers in de trein is gebaseerd op een voorspelling van het SOFA model, waarover meer wordt verteld in paragraaf 4.6.3. Tenslotte worden de gemeten gewichten van de reizigers vergeleken met het aantal reizigers per bak, dat verkregen is door handmatig te tellen in de trein.

### 4.3.1 Bepalen leeg gewicht van de trein

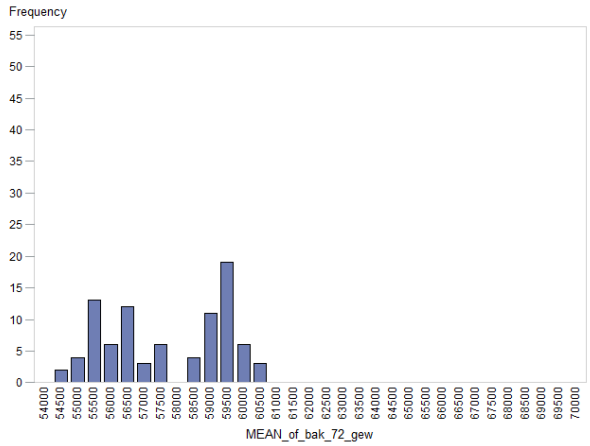
Om de nauwkeurigheid van de beladingssensoren te vergelijken met de Gotcha data worden de metingen van lege treinen geanalyseerd. Deze treinen zijn de treinnummers 70.000 t/m 90.000 in de dienstregeling. Deze rijden zonder reizigers. Ter referentie is ook het theoretisch gewicht van elk type treinbak gegeven. Het aantal lege ritten is beperkt op het traject Utrecht Centraal – Maastricht. Bij de analyse van lege treinen worden daarom alle lege ritten over heel Nederland genomen.

**Beladingssensoren**

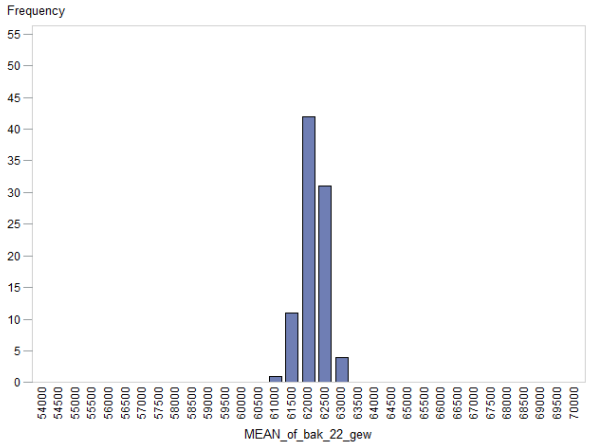
Het gewicht van de motorbakken kan alleen bepaald worden met beladingssensoren bij 16 VIRM-VI treinstellen. Uit de eerste helft van 2015 hielden we 89 lege ritten met beladingssensor data over. Zoals eerder beschreven moest er rekening gehouden worden met het tijdsverschil tussen de beladingssensor data en de daadwerkelijke vertrek- en aankomsttijden van de trein. Acht ritten die uitgevoerd werden door treinstelnummer 8610 zijn verwijderd, omdat het gemiddeld gewicht van de mBvk1 bij deze trein ongeveer 5000 kg lichter is dan de mBvk2, terwijl beide bakken even zwaar zouden moeten zijn. Per rit wordt het gemiddeld gewicht per bak berekend aan de hand van de uitgevoerde metingen tijdens de rit. In de onderstaande figuren zijn de verdelingen van de gemeten gewichten per motorbak weergegeven. Daarnaast is in Tabel 5 het gemiddeld gewicht per bak over alle ritten weergegeven inclusief de standaard deviatie.




Figuur 13: Verdeling gewicht mBvk1 in kg (Beladingssensor)




Figuur 14: Verdeling gewicht mBv7 in kg (Beladingssensor)




Figuur 15: Verdeling gewicht mBvk2 in kg (Beladingssensor)

| Gemiddeld gewicht mBvk1 in kg | Gemiddeld gewicht mBv7 in kg | Gemiddeld gewicht mBvk2 in kg | Standaard deviatie gewicht mBvk1 in kg | Standaard deviatie gewicht mBv7 in kg | Standaard deviatie gewicht mBvk2 in kg |
|---|---|---|---|---|---|
| 62.018 | 57.459 | 62.041 | 550 | 769 | 501 |

Tabel 5: Gemiddeld gewicht en standaard deviatie motorbakken (Beladingssensor)

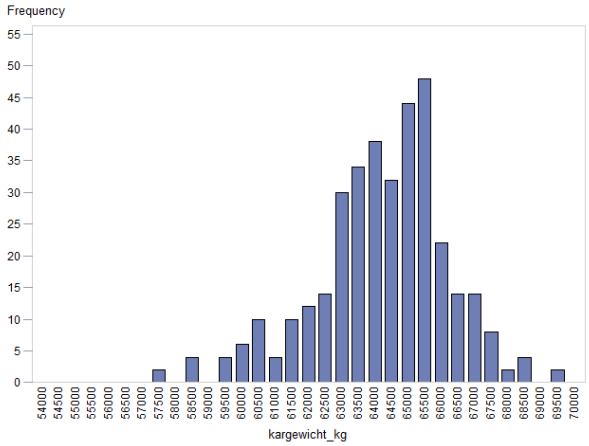
De gewichten van de mBvk1 en mBvk2 komen redelijk overeen. Ook is te zien dat de mBv7 inderdaad lichter is dan de mBvk1 en mBvk2 en dat de standaard deviatie van de gemeten mBv7 gewichten groter is dan bij mBvk1/2. Een Shapiro-Wilk test is uitgevoerd voor de metingen per motorbak. De p-waardes bij elke bak is weergegeven in Tabel 6. Met een significantie-waarde van 0,05 verwerpen we de hypothese dat de metingen bij mBv7 normaal verdeeld zijn. Het valt daarnaast ook op dat de verdeling in Figuur 14 gesplitst kan worden in metingen kleiner en groter dan 58.000 kg. Daarom hebben we onderzocht wat het verschil is tussen deze twee groepen metingen. Hieruit uit blijkt de metingen kleiner dan 58.000 kg bij de volgende treinstelnummers horen: 8608, 8638, 8642 en 8646. De metingen groter dan 58.0000 kg zijn afkomstig van de treinstelnummers: 8621, 8628, 8640, 8652 en 8671. Het verschil tussen deze twee groepen treinstellen is tot nu toe onduidelijk en vereist nog verder onderzoek.

| | mBvk1 | mBv7 | mBvk2 |
|---|---|---|---|
| p-waarde | 0,45 | 2,13e-06 | 0,47 |

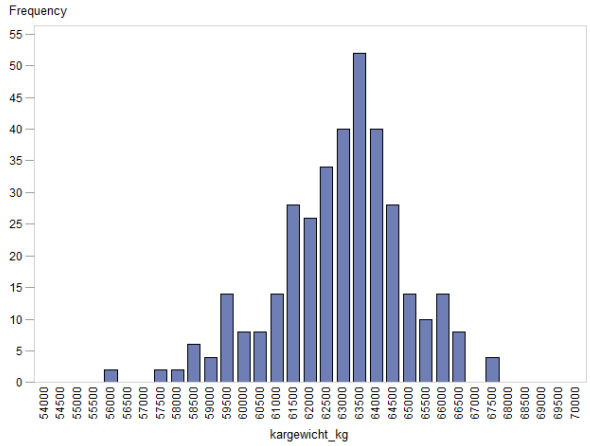
Tabel 6: Shapiro-Wilk test voor metingen van motorbakgewicht (Beladingssensor)

**Gotcha**

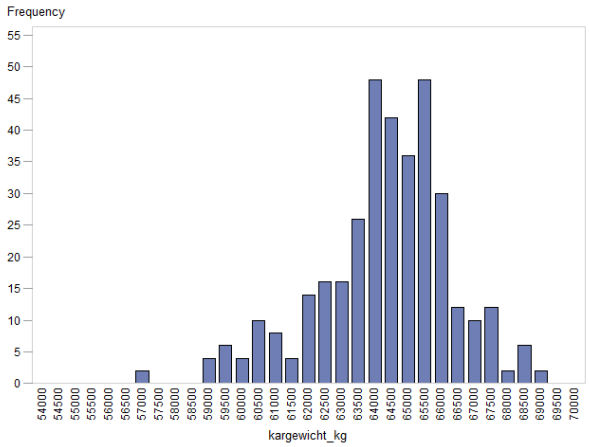
Het bepalen van het gewicht van de motorbakken met Gotcha gaat op dezelfde manier als bij de beladingssensoren, met uitzondering dat data van alle VIRM-VI treinstellen gebruikt kunnen worden. Voor dezelfde periode waren er 347 treinpassages langs meetpunten verkregen. Drie extreme uitschieters zijn verwijderd, waarvan de gewichten kleiner zijn dan 35.000 kg. De verdeling van de gemeten gewichten per motorbak zijn in de volgende grafieken weergegeven. Daarnaast zijn ook het gemiddeld gewicht en standaard deviatie per bak weergegeven in Tabel 7.




Figuur 16: Verdeling gewicht mBvk1 (Gotcha)




Figuur 17: Verdeling gewicht mBv7 (Gotcha)




Figuur 18: Verdeling gewicht mBvk2 (Gotcha)

| Gemiddeld gewicht mBvk1 in kg | Gemiddeld gewicht mBv7 in kg | Gemiddeld gewicht mBvk2 in kg | Standaard deviatie gewicht mBvk1 in kg | Standaard deviatie gewicht mBv7 in kg | Standaard deviatie gewicht mBvk2 in kg |
|---|---|---|---|---|---|
| 64.174 | 62.800 | 64.209 | 1903 | 2029 | 1895 |

Tabel 7: Gemiddeld gewicht en standaard deviatie motorbakken (Gotcha)

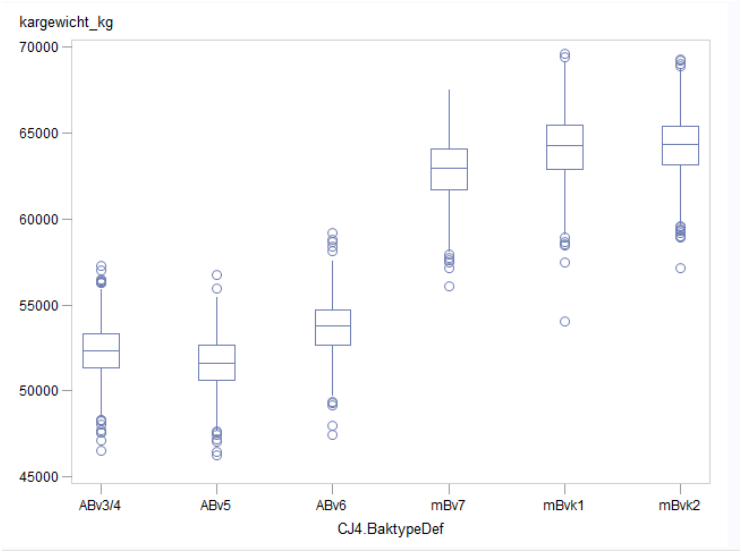
Het gewicht van de mBvk1 en mBvk2 zijn vrijwel gelijk aan elkaar en het gewicht van de mBv7 is net als bij de beladingssensoren kleiner dan mBvk1/2. Uit de figuren zien we dat de verdeling van mBv7 minder afwijkt van de verdeling van mBvk1/2 dan bij de beladingssensoren en ongeveer dezelfde verdeling vertoont. Echter is de standaard deviatie van de metingen relatief groter dan bij de beladingssensoren. Met een Shapiro-Wilk test is gekeken of de verdelingen van de motorbakgewicht

metingen normaal zijn verdeeld. De p-waarden voor elke motorbak zijn weergegeven in Tabel 8. Met een significantie-waarde van 0,05 wordt voor alle drie motorbakken de hypothese dat de metingen normaal verdeeld zijn verworpen.

| | mBvk1 | mBv7 | mBvk2 |
|---|---|---|---|
| p-waarde | 3,45e-03 | 1,11e-02 | 6,85e-04 |

Tabel 8: Shapiro-Wilk test voor metingen van motorbakgewicht (Gotcha)

In Figuur 19 is ook de verdeling van het gewicht van de ABv3/4, ABv5 en ABv6 in een box plot weergegeven. Hier zien we dat de motorbakken significant zwaarder zijn dan de bakken zonder motor.




Figuur 19: Verdeling gewicht per bak (Gotcha)

**Theorie**

De theoretische gewichten van de drie motorbakken zijn in Tabel 9 weergegeven [10].

| Theoretisch gewicht mBvk1 in kg | Theoretisch gewicht mBv7 in kg | Theoretisch gewicht mBvk2 in kg |
|---|---|---|
| 64.029 | 62.141 | 64.029 |

Tabel 9: Theoretisch gewicht motorbakken

De bakken mBvk1 en mBvk2 zijn zoals eerder vermeld even zwaar en de mBv7 bak is ongeveer 2000 kg lichter dan de andere twee motorbakken.

**Vergelijking en correlatie leeg gewicht tussen bakken**

De gewichten van de motorbakken die zijn bepaald met Gotcha, sluiten het best aan bij de theoretisch gewichten volgens Tabel 10. De standaard deviatie van de gewichten is echter ruim drie keer zo groot dan de gewichten die zijn bepaald met de beladingssensoren. Een mogelijke verklaring hiervoor is dat bij Gotcha slechts één keer het gewicht van de bakken wordt gemeten op een traject, terwijl de beladingssensoren gemiddeld elke drie seconden het gewicht van de bakken meten. Er zijn dus meer beladingssensor metingen dan Gotcha metingen op een traject.

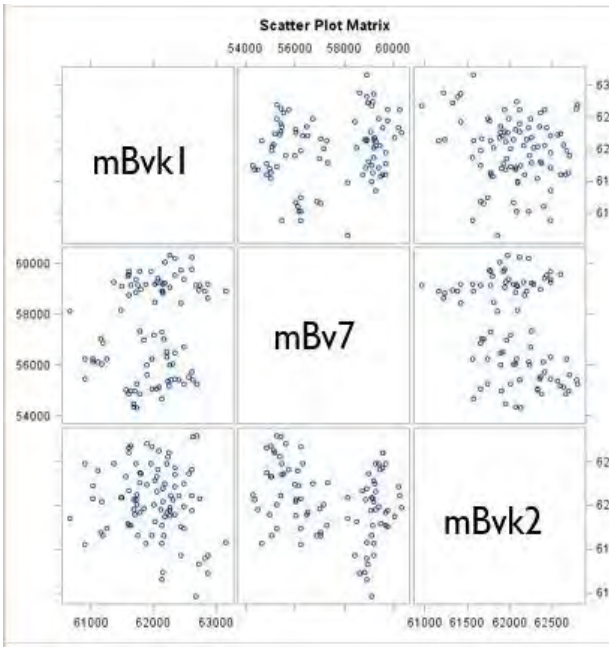
| | Gewicht mBvk1 in kg | Gewicht mBv7 in kg | Gewicht mBvk2 in kg | Standaard deviatie gewicht mBvk1 in kg | Standaard deviatie gewicht mBv7 in kg | Standaard deviatie gewicht mBvk2 in kg |
|---|---|---|---|---|---|---|
| Beladingssensor | 62.018 | 57.459 | 62.041 | 550 | 769 | 501 |
| Gotcha | 64.174 | 62.800 | 64.209 | 1903 | 2029 | 1895 |
| Theorie | 64.029 | 62.141 | 64.029 | - | - | - |

**Tabel 10: Vergelijking gewicht motorbakken (Beladingssensor en Gotcha) met theoretisch gewicht**

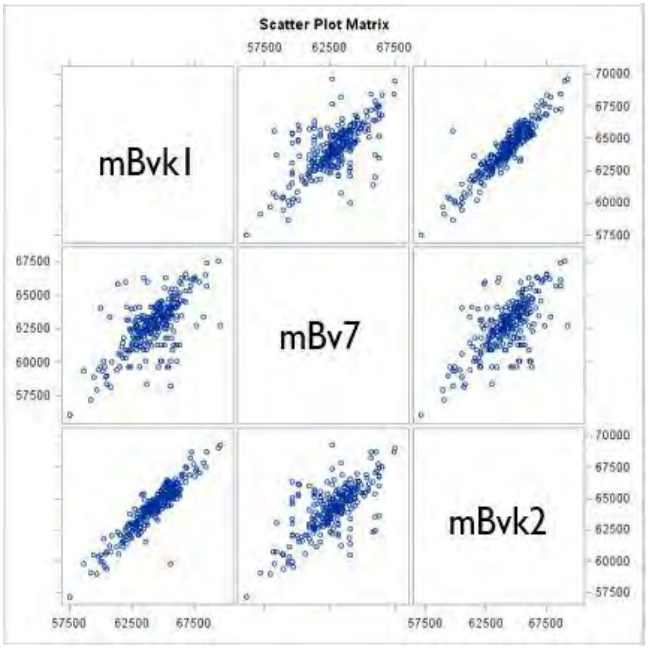
Variatie in het gewicht van de bakken is in eerste instantie geen groot probleem bij het bepalen van de reizigersverdeling. Zolang de meetfouten van de verschillende bakken positief gecorreleerd zijn, zal dit weinig invloed hebben op reizigersverdeling.

Daarom is in Figuur 20 en Figuur 21 de metingen van de bakken tegenover elkaar geplot. In Figuur 20 zien we dat er weinig verband is tussen de metingen van verschillende bakken. We zien bijvoorbeeld dat bij een gewicht van ongeveer 62.100 kg voor de mBvk2 de ene keer het gewicht van de mBvk1 gelijk is aan 61.250 kg en de andere keer weegt deze 62.500 kg. In Figuur 21 is een positieve correlatie tussen de metingen van de verschillende bakken duidelijk te zien. Neem bijvoorbeeld het punt waarbij 57.500 kg is gemeten voor de mBvk1 en 57.000 kg voor de mBvk2. Beide metingen zitten ongeveer evenveel (7000 kg) onder het theoretisch gewicht. In Tabel 11 en Tabel 12 zijn de correlatie coëfficiënten tussen het gewicht van de verschillende bakken weergegeven. Onder de correlatie coëfficiënten zijn ook de bijhorende p-waarden gegeven. Deze waarden geven aan hoe significant de correlatie coëfficiënten zijn. Bij p-waarden die kleiner zijn dan de significantiewaarde van 0,05 wordt de nulhypothese verworpen dat er geen correlatie is tussen de metingen. In Tabel 11 is dit het geval wanneer de beladingssensor metingen van mBvk2 met de metingen van mBv7 worden vergeleken. Er is tevens sprake van een negatieve correlatie. In Tabel 12 zijn alle correlatie coëfficiënten significant, voornamelijk omdat er meer Gotcha metingen van de lege ritten beschikbaar zijn dan beladingssensor metingen. De correlaties tussen de Gotcha metingen van de verschillende bakken zijn sterk positief. De correlatie tussen de metingen van mBvk1 en mBvk2 is zelfs groter dan 0,9. Dit betekent dat bij Gotcha de meetfouten tussen de motorbakken vrijwel overeenkomen, wat van belang is om de juiste verhouding van het reizigersgewicht over de bakken te kunnen bepalen.

Het verschil in correlatie tussen Gotcha en de beladingssensoren komt vermoedelijk doordat elke motorbak over een eigen beladingssensor bezit en dus de metingen van de beladingssensoren onafhankelijk van elkaar zijn. Bij Gotcha daarentegen wordt telkens bij het meten van het gewicht van de bakken dezelfde meetpunt gebruikt.




Figuur 20: Scatterplot gewicht motorbakken in kg (Beladingssensor)




Figuur 21: Scatterplot gewicht motorbakken in kg (Gotcha)

| Pearson Correlation Coefficients, N = 89<br>Prob > \|r\| under H0: Rho=0 | | | |
|---|---|---|---|
| | mBvk1 | mBv7 | mBvk2 |
| mBvk1 | 1.00 | 0.20 | -0.16 |
| | - | 0.07 | 0.13 |
| mBv7 | 0.20 | 1.00 | -0.32 |
| | 0.07 | - | <0.01 |
| mBvk2 | -0.16 | -0.32 | 1.00 |
| | 0.13 | <0.01 | - |

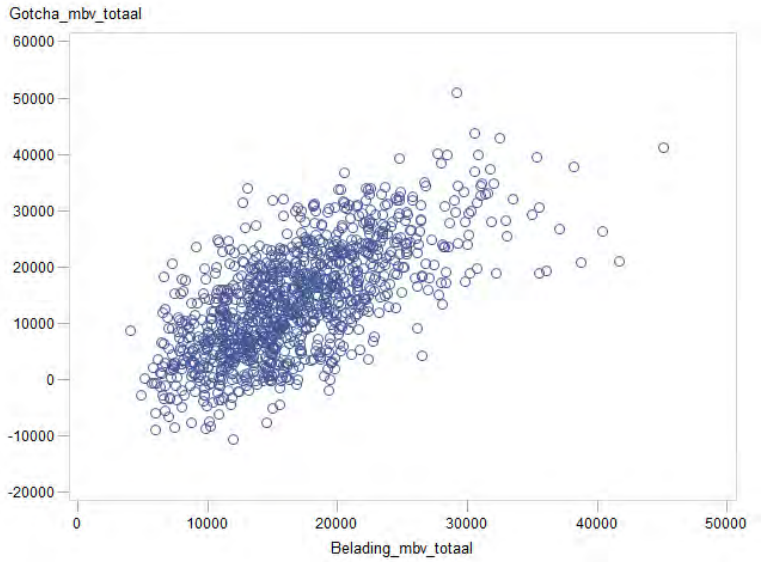
Tabel 11: Correlatie gewicht motorbakken (Beladingssensor)

| Pearson Correlation Coefficients, N = 347<br>Prob > \|r\| under H0: Rho=0 | | | |
|---|---|---|---|
| | mBvk1 | mBv7 | mBvk2 |
| mBvk1 | 1.00 | 0.66 | 0.93 |
| | - | <0.01 | <0.01 |
| mBv7 | 0.66 | 1.00 | 0.69 |
| | <0.01 | - | <0.01 |
| mBvk2 | 0.93 | 0.69 | 1.00 |
| | <0.01 | <0.01 | - |

Tabel 12: Correlatie gewicht motorbakken (Gotcha)

### 4.3.2 Correlatie Gotcha met beladingssensor

Vervolgens wordt onderzocht in hoeverre het gewicht van de reizigers in de trein verschilt per databron. Hierbij worden de metingen op de trajecten tussen Utrecht Centraal en Maastricht gebruikt waar zowel beladingssensor data als Gotcha data beschikbaar is. Bij Gotcha wordt het theoretisch gewicht van de bakken afgetrokken van de gemeten waarden om het gewicht van de reizigers te krijgen, omdat in paragraaf 4.3.1 de Gotcha metingen van de lege bakken vrijwel overeenkomen met het theoretisch gewicht. De beladingssensor metingen weken af van het theoretisch gewicht, dus hiervoor wordt het minimum gewicht per dag, motorbak en treinstelnummer afgetrokken van de beladingssensor metingen.



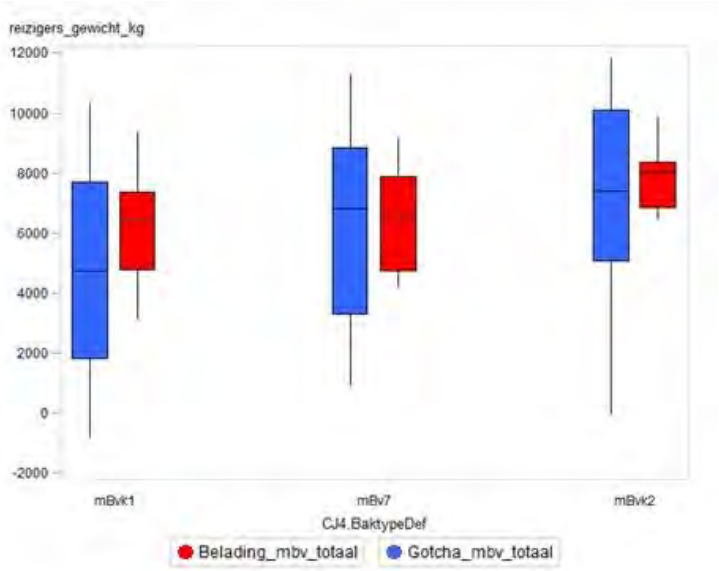

Figuur 22: Scatterplot reizigersgewicht in kg van Gotcha en beladingssensor

| Pearson Correlation Coefficients, N = 1102<br>Prob > \|r\| under H0: Rho=0 | |
|---|---|
| | Belading_mbv_totaal |
| Gotcha_mbv_totaal | 0.67<br><0.01 |

Tabel 13: Correlatie reizigersgewicht in kg van Gotcha met beladingssensor

De correlatie tussen het totaal gewicht van de reizigers in de drie motorbakken bij Gotcha en de beladingssensoren is in Figuur 22 weergegeven. Volgens Tabel 13 is de correlatie gelijk aan 0,67, wat een redelijk sterke correlatie aanduidt. Uit de grafiek is te zien dat er ook negatieve gewichten voorkomen bij Gotcha. De oorzaak hiervan is dat soms de metingen kleiner zijn dan het gewicht dat wordt afgetrokken. Daarnaast zagen we hiervoor dat de variatie in de metingen van Gotcha groter is dan de variatie in de metingen van de beladingssensoren.

Om een beeld te hebben van de verdeling van het reizigersgewicht over de drie motorbakken wordt als voorbeeld treinnummer 823 genomen die van Utrecht Centraal naar ‘s-Hertogenbosch rijdt. In de eerste helft van 2015 waren er 11 dagen waarvan we zowel beladingssensor data als Gotcha data hebben. In Figuur 23 is de verdeling van het reizigersgewicht per motorbak weergegeven voor beide databronnen in de vorm van een boxplot.

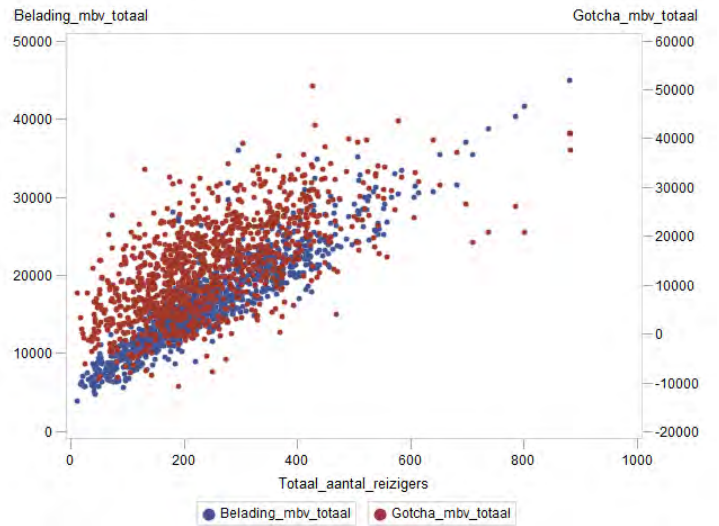



Figuur 23: Verdeling reizigersgewicht van Gotcha en beladingssensor

Opnieuw zien we dat de variatie in de Gotcha metingen groter is dan de variatie bij de metingen van de beladingssensoren. Daarnaast valt het op dat het reizigersgewicht in de mBvk2 groter is dan bij mBvk1.

### 4.3.3 Correlatie Gotcha en beladingssensor met SOFA

Het is nuttig om te onderzoeken of het berekend gewicht van de reizigers wel overeenkomt met het aantal reizigers in de trein, want het is mogelijk dat de beladingssensoren en Gotcha niet nauwkeurig genoeg zijn om het aantal reizigers in de trein te bepalen. Het aantal reizigers wordt bepaald met een applicatie dat de zitplaatskans in de trein berekend, genaamd SOFA. Deze kansen worden niet real-time bepaald, maar zijn achteraf berekend. SOFA maakt hiervoor gebruik van OV-chipkaart gegevens. Een tussenresultaat van SOFA is het aantal reizigers in de trein. Deze wordt hier gebruikt.
In Figuur 24 en Tabel 14 is te zien dat het reizigersgewicht over de motorbakken bij de beladingssensoren het meest correleert met het aantal voorspelde reizigers uit SOFA. Er moet wel rekening gehouden worden met dat dit het aantal reizigers is over de hele trein en het reizigersgewicht alleen het gewicht van de reizigers in de motorbakken is.




Figuur 24: Scatterplot aantal reizigers met reizigersgewicht Gotcha en beladingssensor

| Pearson Correlation Coefficients, N = 1102<br>Prob > \|r\| under H0: Rho=0 | | |
|---|---|---|
| | Belading_mbv_totaal | Gotcha_mbv_totaal |
| Totaal_aantal_reizigers | 0.92<br><0.01 | 0.65<br><0.01 |

Tabel 14: Correlatie aantal reizigers met reizigersgewicht Gotcha en beladingssensor

### 4.3.4 Vergelijking reizigerstellingen met Gotcha

In de voorgaande analyses waren er zowel redenen om de beladingssensor data als de Gotcha data te gebruiken bij het bepalen van de verdeling van de reizigers over de bakken. Daarom is het van belang om beide metingen te vergelijken met de daadwerkelijke verdeling van reizigers in de trein. Om de daadwerkelijke verdeling van de reizigers in de trein te kunnen bepalen, moet het aantal reizigers in elke bak gegeven zijn. NS heeft hiervan geen data, dus de data moet verkregen worden door handmatig te tellen in de trein. Op twee dagen (26 mei 2016 & 30 mei 2016) werd op een paar trajecten het aantal reizigers in elke bak geteld. In totaal is de reizigersverdeling verkregen van 10 trajecten. Helaas kan deze verdeling alleen met de Gotcha data vergeleken worden, omdat de beladingssensor data niet beschikbaar was op de dagen dat de reizigerstellingen werden uitgevoerd. Het blijkt dat sinds eind 2015 er een software fout heeft plaatsgevonden bij het ophalen van de beladingssensor data. In Tabel 15 en Tabel 16 is de daadwerkelijke reizigersverdeling naast de reizigersverdeling uit Gotcha gezet voor twee trajecten. Daarnaast is ook het verschil per bak weergegeven. De overige acht reizigerstellingen zijn te vinden in Appendix B. De reizigersverdeling uit Gotcha is bepaald door het theoretisch gewicht van de bakken af te halen van de Gotcha metingen, omdat het gewicht van de bakken dat wordt bepaald met Gotcha weinig verschilt met het theoretisch gewicht.

| Datum: | 26-mei | Traject: | Ht-Ehv |
|---|---|---|---|
| Treinnummer: | 833 | Aantal reizigers: | 270 |
| Materieelnummer: | 8642 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 15,2% | 18,3% | 3,1% |
| ABv5 | 5,9% | 7,7% | 1,8% |
| mBv7 | 16,3% | 17,5% | 1,2% |
| ABv6 | 19,3% | 18,3% | 1,0% |
| ABv3/4 | 19,6% | 15,9% | 3,7% |
| mBvk2 | 23,7% | 22,3% | 1,4% |
| | | Gemiddelde | 2,1% |
| | | Maximum | 3,7% |

Tabel 15: Reizigersverdeling op basis van treintellingen en Gotcha (1)

In Tabel 15 zien we dat de verdeling uit Gotcha redelijk overeenkomt met de daadwerkelijke verdeling van de reizigers. Volgens de daadwerkelijke verdeling zitten de minste mensen in de ABv5-bak en de meeste mensen in de mBvk2-bak. Hetzelfde is terug te zien bij de verdeling uit Gotcha.

| Datum | 26-mei | Traject | Ht-Ut |
|---|---|---|---|
| Treinnummer | 300836 | Aantal reizigers | 249 |
| Materieelnummer | 8642 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 3,2% | 11,6% | 8,4% |
| ABv5 | 8,4% | 10,1% | 1,7% |
| mBv7 | 17,7% | 16,7% | 1,0% |
| ABv6 | 18,5% | 17,4% | 1,1% |
| ABv3/4 | 20,1% | 18,4% | 1,7% |
| mBvk2 | 32,1% | 25,7% | 6,4% |
| | | Gemiddelde | 3,4% |
| | | Maximum | 8,4% |

**Tabel 16: Reizigersverdeling op basis van treintellingen en Gotcha (2)**

Op het traject in Tabel 16 zien we dat het percentage reizigers vanaf de mBvk1 richting de mBvk2 geleidelijk toeneemt. De Gotcha metingen waren in staat om deze trend weer te geven met uitzondering van de mBvk1 bak waar meer gewicht is gemeten dan dat er aan reizigers in de mBvk1 bak zitten.

Vervolgens is een Wilcoxon signed-rank test toegepast om te testen of de reizigerspercentages uit de 10 tellingen significant verschillen van de reizigerspercentages uit Gotcha. Uit deze test is een p-waarde van 0,98 verkregen. Dit betekent dat er geen significant verschil is tussen de reizigerspercentages van de tellingen en Gotcha.

Het bepalen van de reizigersverdeling met Gotcha vormt een probleem wanneer negatieve reizigersgewichten ontstaan, nadat het theoretisch gewicht van de bakken is afgetrokken van de Gotcha metingen. Dit is niet voorgekomen bij de 10 tellingen die zijn uitgevoerd. De negatieve gewichten ontstaan voornamelijk door de grote standaard deviatie in de Gotcha metingen. Met negatieve reizigersgewichten kan het percentage reizigers in elke bak niet rechtstreeks worden bepaald.

Aangezien het onderzoek alleen focust op het percentage reizigers in elke bak en niet de absolute aantallen, wordt het probleem opgelost door bij elke bak een kleiner gewicht dan het theoretisch gewicht af te trekken van de Gotcha metingen. Hoeveel gewicht er wordt afgetrokken, hangt af van het minimum gemeten gewicht per bak (inclusief reizigers) uit de data in paragraaf 4.3.2. De nieuwe gewichten die worden afgetrokken zijn in de laatste kolom van Tabel 17 weergegeven. Merk op dat deze gewichten kleiner zijn dan de minimum gemeten gewichten en dat de nieuwe gewichten voor de bakken dezelfde verhouding hebben als de theoretische gewichten. Wanneer deze nieuwe gewichten worden afgetrokken van de Gotcha metingen zal de reizigersverdeling uit Gotcha het best behouden blijven.

| Baktype | Theoretisch leeg gewicht in kg | Minimum gemeten gewicht in kg (inclusief reizigers) | Nieuw leeg gewicht in kg |
|---|---|---|---|
| mBvk1 | 64.029 | 58.309 | 58.029 |
| ABv5 | 51.334 | 47.767 | 46.524 |
| mBv7 | 62.141 | 59.293 | 56.318 |
| ABv6 | 53.465 | 49.787 | 48.455 |
| ABv3/4 | 52.214 | 48.543 | 47.321 |
| mBvk2 | 64.029 | 59.012 | 58.029 |

**Tabel 17: Theoretische en nieuwe bakgewichten voor Gotcha**

Belangrijk om op te merken is dat deze nieuwe berekende gewichten niet de werkelijke gewichten van de bakken zijn, maar alleen gebruikt worden om ongeveer het percentage reizigers in de bakken te kunnen bepalen.

In Tabel 18 is voor één traject waarop het aantal reizigers per bak is geteld de daadwerkelijke reizigersverdeling vergeleken met de verdeling die is bepaald door het theoretisch gewicht af te trekken van de Gotcha metingen en de verdeling waarbij het nieuw leeg gewicht is afgetrokken.

| | | | | | |
|---|---|---|---|---|---|
| Datum: | | 26-mei | Traject: | | Ut-Ht |
| Treinnummer: | | 833 | Aantal reizigers: | | 252 |
| Materieelnummer: | | 8642 | | | |
| Reizigersverdeling | | | | | |
| | Tellingen | Gotcha - theoretisch gewicht | Verschil | Gotcha - nieuw gewicht | Verschil |
| mBvk1 | 17,9% | 16,4% | 1,5% | 17,3% | 0,6% |
| ABv5 | 13,1% | 8,7% | 4,4% | 11,5% | 1,6% |
| mBv7 | 24,2% | 22,9% | 1,3% | 20,8% | 3,4% |
| ABv6 | 11,1% | 15,6% | 4,5% | 15,5% | 4,3% |
| ABv3/4 | 13,5% | 14,1% | 0,6% | 14,5% | 1,0% |
| mBvk2 | 20,2% | 22,2% | 2,0% | 20,5% | 0,3% |
| | | Gemiddeld | 2,4% | | 1,9% |
| | | Maximum | 4,5% | | 4,3% |

**Tabel 18: Vergelijking reizigersverdeling bij aftrekken van theoretisch en nieuw gewicht (één telling)**

Hier zien we dat de reizigersverdeling niet veel is veranderd wanneer een ander gewicht van de Gotcha metingen wordt afgetrokken. Integendeel, de nieuwe verdeling komt beter overeen met de werkelijke reizigersverdeling in dit specifiek voorbeeld.

Tenslotte is over de 10 reizigerstellingen het gemiddeld verschil per bak tussen het werkelijk percentage reizigers en het percentage reizigers uit Gotcha in Tabel 19 weergegeven.

| Gemiddeld verschil met reizigerstellingen | | |
|---|---|---|
| | Gotcha - theoretisch gewicht | Gotcha - nieuw gewicht |
| mBvk1 | 4,8% | 5,3% |
| ABv5 | 2,5% | 3,3% |
| mBv7 | 2,1% | 2,4% |
| ABv6 | 3,3% | 3,0% |
| ABv3/4 | 2,6% | 3,6% |
| mBvk2 | 3,2% | 3,5% |
| Gemiddelde | 3,1% | 3,5% |
| Maximum | 4,8% | 5,3% |

**Tabel 19: Vergelijking reizigersverdeling bij aftrekken van theoretisch en nieuw gewicht (alle tellingen)**

Het verschil is het grootst bij de mBvk1, ongeacht of het theoretisch gewicht of het nieuw gewicht van de Gotcha metingen is afgetrokken. Met een Wilcoxon signed-rank test is gekeken of de verschillen van de Gotcha metingen met de 10 reizigerstellingen tussen beide aanpakken significant verschillen. Dit levert een p-waarde op van 4,56e-02 dat net kleiner is dan de significantie waarde van 0,05. In principe betekent dit dat er een significant verschil is tussen beide aanpakken.

Daarom is opnieuw een Wilcoxon signed-rank test toegepast om te testen of de reizigerspercentages uit de 10 tellingen significant verschillen van de reizigerspercentages uit Gotcha waarbij nieuwe gewichten zijn afgetrokken. Uit deze test is een p-waarde van 0,99 verkregen. Dus voor beide aanpakken kunnen we vrij zeker van zijn dat er geen significant verschil is tussen de Gotcha metingen en de reizigerstellingen.

## 4.4 Conclusie

Om de juiste reizigersverdeling te kunnen bepalen, is onderzocht hoe nauwkeurig de metingen van de beladingssensoren zijn vergeleken met de metingen van Gotcha. Uit het onderzoek van de lege ritten blijkt dat de gemeten gewichten van de lege bakken uit Gotcha meer overeenkomen met de theoretische gewichten, maar de standaard deviatie van deze metingen groter is dan bij de beladingssensor metingen. Daarnaast is de correlatie van het voorspeld aantal reizigers uit SOFA met het totaal reizigersgewicht in de motorbakken dat is bepaald uit de beladingssensor data sterker dan met Gotcha. Toch gaat de voorkeur naar Gotcha, omdat we de Gotcha metingen konden vergelijken met de daadwerkelijke reizigersverdeling en deze de reizigersverdeling redelijk kon weergeven. Een andere reden is dat bij Gotcha de gewichten van de bakken met elkaar correleren, wat heel belangrijk is om de reizigersverdeling in de trein te kunnen bepalen. Aangezien het reizigersgewicht uit de beladingssensoren sterk correleert met SOFA, zouden de beladingssensoren wel gebruikt kunnen worden voor het bepalen van het totaal aantal reizigers in de trein. Verder heeft Gotcha als voordeel dat het gewicht van alle bakken wordt gemeten in plaats van alleen de motorbakken bij de beladingssensoren. Bovendien kan met Gotcha ook het gewicht van andere treintypen gemeten worden en is het mogelijk eenvoudiger meer Gotcha meetpunten aan te leggen dan alle treinen te voorzien van beladingssensoren.

# 5 Model

In dit hoofdstuk wordt de uiteindelijke dataset voor het voorspellingsmodel geanalyseerd om een idee te krijgen welke attributen in de dataset een belangrijke rol spelen bij het voorspellen van de reizigersverdeling in de trein. Daarna volgen de gebruikte algoritmes voor het voorspellingsmodel en een methode om het model te evalueren. Tot slot worden de meest significante attributen geselecteerd m.b.v. twee attributen selectie methoden. Deze attributen worden dan gebruikt om de modellen te trainen.

## 5.1 Dataset

In het vorig hoofdstuk is de keuze gemaakt om de reizigersverdeling weer te geven met Gotcha. De Gotcha data wordt met de datasets 'Treinactiviteiten' en 'Materieelplanning' gekoppeld, zodat op elke regel de reizigersverdeling in de trein voor aankomst en de reizigersverdeling bij vertrek uit een station zijn gegeven. In totaal zijn er 5152 observaties verkregen. Deze nieuwe dataset wordt gebruikt om een model te trainen dat de reizigersverdeling bij vertrek kan voorspellen. De attributen in deze dataset worden hierna toegelicht.

### 5.1.1 Attributen

De meeste attributen in onze dataset zijn rechtstreeks uit 'Treinactiviteiten', 'Materieelplanning' en Gotcha gehaald. Attributen die afkomstig zijn van andere datasets worden hieronder aangegeven. Verder is ook aangegeven welke attributen afgeleid zijn en waarom deze attributen zijn opgenomen in de dataset.

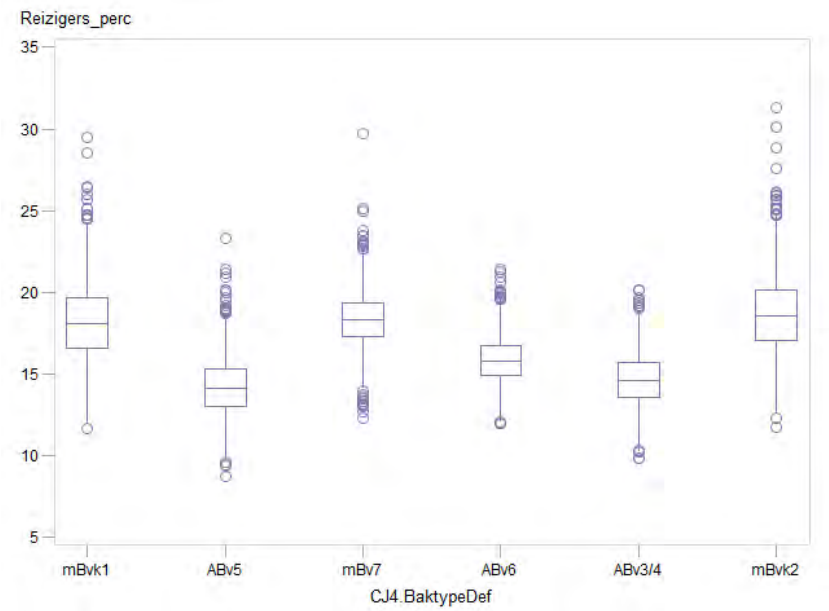
| Attributen | Omschrijving |
|---|---|
| *Uur* | Het uur waarin de trein op het station aankomt. Vermoedelijk heeft het tijdstip invloed op de reizigersverdeling. Vooral tijdens de spits is het aantal reizigers meestal groter en zal de verdeling ook anders zijn dan buiten de spits. Deze attribuut is aangemaakt in de plaats van de exacte aankomsttijd op het station, omdat het groeperen van de observaties in uren eenvoudiger is. |
| *Dag van de week* | Met deze attribuut kan bijvoorbeeld gekeken worden of reizigers zich anders verdelen op een doordeweekse dag dan in het weekend. |
| *Maand* | De reizigers zouden zich anders in de trein kunnen verdelen in de zomermaanden, omdat veel mensen op vakantie gaan en er bijvoorbeeld meer toeristen in de trein zitten. |
| *Treinnummer* | Dit attribuut heeft ongeveer dezelfde functie als het attribuut *uur*.<br>Het treinnummer verwijst naar een specifiek trein binnen de treinserie die elke dag rond een |

| | |
|---|---|
| | bepaald tijdstip rijdt. De richting waar de trein op rijdt wordt met even en oneven treinnummers onderscheiden. |
| *Materieelnummer* | Het specifiek treinstel dat heeft gereden. |
| *Bijzonderheden* | Dit geeft bijvoorbeeld aan of er een ander treinstel heeft gereden dan is gepland. Om dit aan te geven wordt “300” voor het treinnummer toegevoegd. |
| *Huidig station* | Het station waar de verandering van de reizigersverdeling plaatsvindt. Dit is in ons onderzoek het station ‘s-Hertogenbosch of station Eindhoven. |
| *Richting traject* | Een binaire variabele die de rijrichting van de trein aangeeft. |
| *Richting trein* | Een binaire variabele dat aangeeft of de mBvk1 voorop rijdt of de mBvk2. |
| *Vertraging* | Het aantal seconden vertraging op het vorig station. |
| *Aantal reizigers in de trein bij aankomst* | Dit aantal is gebaseerd op de voorspellingen van SOFA. Het aantal reizigers zou invloed kunnen hebben op de reizigersverdeling, bijvoorbeeld wanneer de trein overvol zit, zijn de reizigers meestal gelijkmatig over de bakken verdeeld. |
| *Aantal reizigers in de trein bij vertrek* | Hier wordt voorlopig de voorspellingen van SOFA gebruikt. Deze voorspellingen zijn pas achteraf gemaakt, maar binnenkort is het mogelijk om real-time het aantal reizigers te voorspellen. |
| *Gemiddeld aantal reizigers in de trein in het verleden (2x)* | Het gemiddeld aantal reizigers in de trein wordt berekend van de vorige twee metingen van hetzelfde treinnummer en bij dezelfde dag van de week. Het gemiddeld aantal reizigers is zowel voor aankomst als bij vertrek berekend. |
| *Gewicht (6x) en percentage (6x) van de reizigers per bak bij aankomst* | Bij een VIRM-VI hebben we zes attributen die het gewicht in elke bak en zes attributen die het percentage reizigers in elke bak aangeven. Deze attributen zijn waarschijnlijk één van de belangrijkste attributen voor het voorspellen van de reizigersverdeling bij vertrek, omdat we willen onderzoeken of het mogelijk is om de reizigersverdeling bij vertrek te voorspellen met |

| | |
|---|---|
| | de reizigersverdeling bij aankomst. |
| *Gemiddeld gewicht (12x) en percentage (12x) van de reizigers per bak in het verleden* | Per bak wordt het gemiddeld gewicht van de reizigers genomen van de vorige twee metingen van hetzelfde treinnummer en bij dezelfde dag van de week. Daarna is per bak ook het percentage van deze gewichten over het totaal gewicht berekend. De gewichten en percentages worden zowel voor aankomst als bij vertrek berekend. |
| *Weer (9x)*<br>- Windsnelheid met een nauwkeurigheid van 0.1 m/s<br>- Temperatuur met een nauwkeurigheid van 0.1 graden Celsius<br>- Duur van zonneschijn en neerslag met een nauwkeurigheid van 0.1 uur<br>- Het wel of niet voorkomen van de volgende weerverschijnselen: Mist, regen, sneeuw, onweer en ijsvorming | De attributen die betrekking hebben tot het weer zijn afkomstig van een extern databron[12]. |

### 5.1.2 Analyse dataset

Vóór het toepassen van een model op de dataset wordt de reizigersverdeling bestudeerd op verschillende trajecten. In Figuur 25 is het gemiddeld reizigerspercentage per bak geplot voor het traject ‘s-Hertogenbosch – Eindhoven over alle ritten.




**Figuur 25: Verdeling reizigersgewicht op traject ‘s-Hertogenbosch - Eindhoven**

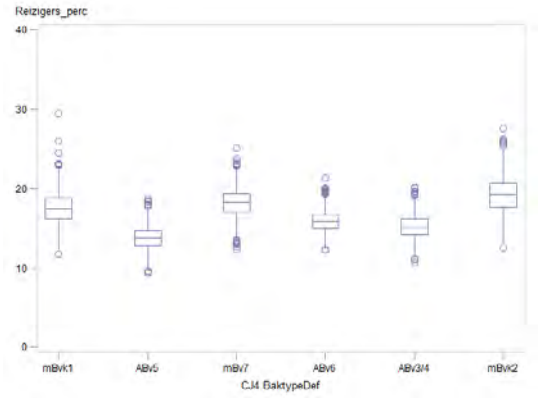
Hieruit zien we dat de meeste reizigers in de motorbakken zitten. Om te onderzoeken wat de oorzaak hiervan is, wordt naar het aantal zitplaatsen per bak gekeken. Deze is weergegeven in Tabel 20. Hier zien we dat het totaal aantal zitplaatsen per baktype ongeveer gelijk is. Dus in de motorbakken zitten daadwerkelijk meer reizigers, terwijl de capaciteit van alle bakken ongeveer gelijk is. Verder zien we dat in de motorbakken ongeveer twee keer zoveel tweede klas zitplaatsen zijn dan in de bakken waar ook eerste klas zitplaatsen zijn. Een mogelijke verklaring voor de drukte in de motorbakken is dat er relatief meer tweede klas reizigers zijn dan eerste klas reizigers en er meer tweede klas zitplaatsen beschikbaar zijn in de motorbakken dan in de ABv bakken.

| | mBvk1 | ABv5 | mBv7 | ABv6 | ABv3/4 | mBvk2 | Totaal |
|---|---|---|---|---|---|---|---|
| 1e klas | 0 | 45 | 0 | 45 | 39 | 0 | 129 |
| 2e klas | 93 | 42 | 92 | 42 | 50 | 93 | 412 |
| Totaal | 93 | 87 | 92 | 87 | 89 | 93 | 541 |

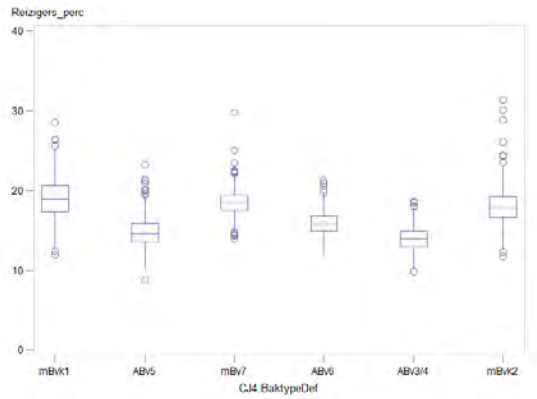
Tabel 20: Aantal zitplaatsen per VIRM bak

**Reizigersverdeling met mBvk1 of mBvk2 voorop**

Voor het traject 's-Hertogenbosch – Eindhoven wordt vervolgens onderscheid gemaakt tussen ritten waar mBvk1 voorop rijdt en ritten waar mBvk2 vooraan zit. Dit is gedaan om te onderzoeken of reizigers bijvoorbeeld vaker voorin of achterin de trein zitten ongeacht het baktype. In Figuur 26 en Figuur 27 zien we opnieuw dat het merendeel van de reizigers in de motorbakken zitten. Op het eerste gezicht lijkt er nauwelijks verschil te zijn tussen de twee verdelingen, maar als we goed kijken zien we dat in Figuur 26 meer reizigers in de mBvk2 zitten dan in de mBvk1 en meer reizigers in ABv3/4 t.o.v. ABv5. In Figuur 27 is dit omgekeerd. Om deze waarneming te bevestigen wordt het verschil tussen deze bakken op significantie getest. Allereerst wordt gekeken of het percentage reizigers per bak normaal verdeeld is.

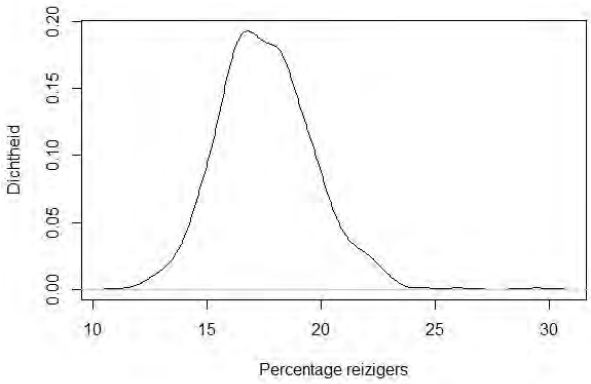



Figuur 26: Reizigersverdeling op traject 's-Hertogenbosch - Eindhoven met mBvk1 voorop

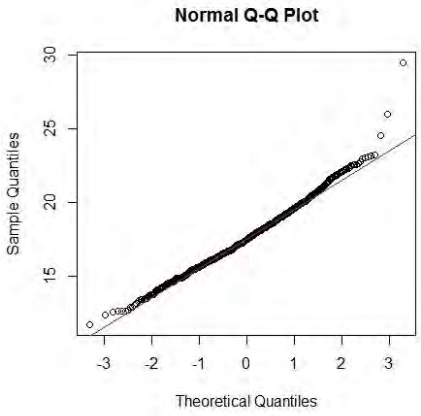



Figuur 27: Reizigersverdeling op traject 's-Hertogenbosch - Eindhoven met mBvk2 voorop

Dit is als voorbeeld gedaan voor de reizigerspercentages in mBvk1 uit Figuur 26. In Figuur 28 is de verdeling hiervan geplot en in Figuur 29 is een QQ-plot weergegeven. Uit beide figuren lijkt het erop dat de percentages niet helemaal normaal verdeeld zijn. Tenslotte wordt een Shapiro-Wilk test uitgevoerd waarbij een p-waarde van 2.185e-07 is verkregen. Met een significantie-waarde van 0.05 wordt dus de hypothese verworpen dat de percentages normaal verdeeld zijn.




Figuur 28: Verdeling reizigerspercentages in mBvk1 op traject 's-Hertogenbosch - Eindhoven met mBvk1 voorop



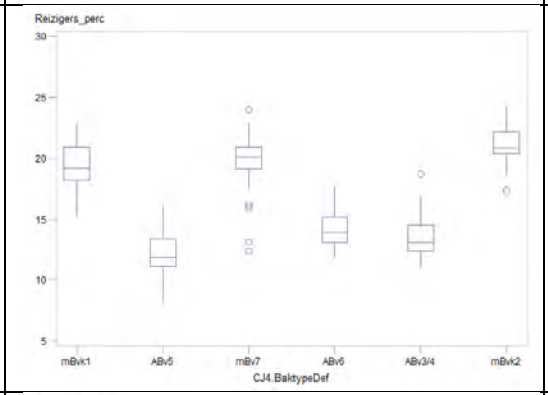
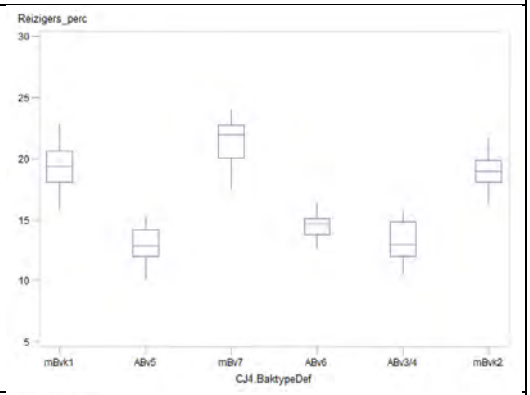
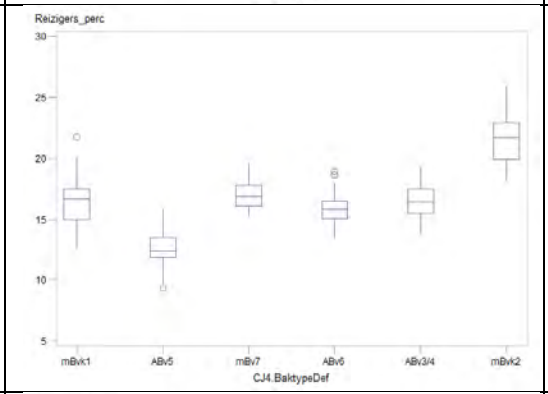
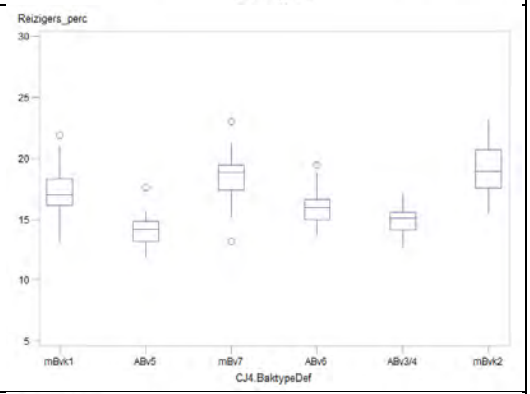
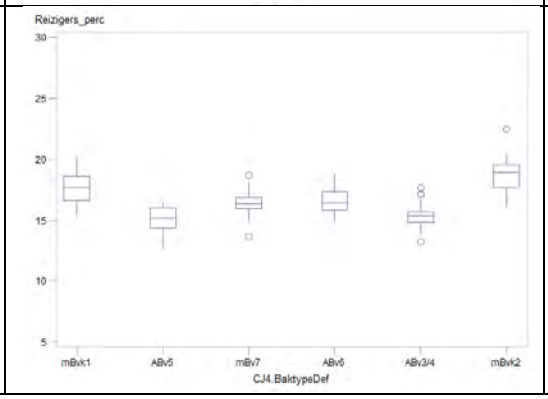
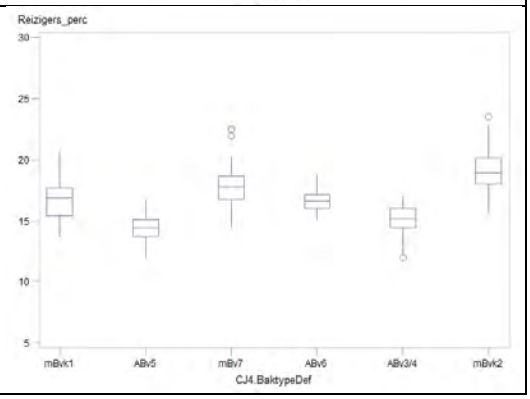

Figuur 29: QQ-plot van reizigerspercentages in mBvk1 op traject 's-Hertogenbosch - Eindhoven met mBvk1 voorop

Om te testen of het percentage reizigers in de mBvk1 significant verschillen van het percentage in de mBvk2 wordt een Wilcoxon signed-rank test toegepast, omdat deze test geschikt is voor niet normale verdelingen.

Een p-waarde kleiner dan 2.2e-16 is verkregen bij het vergelijken van de percentages van de mBvk1 en de mBvk2 uit Figuur 26. Dat betekent dat hier inderdaad gemiddeld vaker reizigers in de mBvk2 zitten dan in de mBvk1. Met de Wilcoxon signed-rank test kan ook worden aangetoond dat in Figuur 26 meer mensen in de ABv3/4 zitten dan in de ABv5. Dus gemiddeld zitten er meer mensen in de achterkant van de trein wanneer de mBvk1 voorop rijdt. Uit de Wilcoxon signed-rank test volgt juist dat er meer reizigers in de mBvk1 en mBv5 zitten dan in de mBvk2 en ABv3/4 wanneer de mBvk2 voorop zit. Dus er zitten gemiddeld meer mensen achter in de trein op het traject van 's-Hertogenbosch naar Eindhoven ongeacht welke bak voorop staat. Een verklaring hiervoor is dat de achterkant van de trein dichterbij de trappen op het perron van 's-Hertogenbosch stopt. Reizigers zouden dan minder geneigd zijn om aan de andere kant van de trein in te stappen. Vooral wanneer reizigers die laat aankomen nog de trein willen halen, zullen eerder in de dichtstbijzijnde bak stappen.

**Reizigersverdeling tijdens en buiten de spits**

Daarna wordt gekeken hoe de verdeling van de reizigers in de trein verandert op de stations 's-Hertogenbosch en Eindhoven. In Tabel 21 zijn de verdelingen geplot op de trajecten Utrecht Centraal – 's-Hertogenbosch (Ut – Ht), 's-Hertogenbosch – Eindhoven (Ht – Ehv) en Eindhoven – Weert (Ehv – Wt) van treinnummer 821 en 843. Treinnummer 821 rijdt tijdens de ochtendspits rond 07:00 en 843 rijdt 's middags rond 12:00. Alleen treinen waarbij de mBvk1 voorop rijdt worden meegenomen.

| + | 821 | 843 |
|---|---|---|
| Ut - Ht |  |  |
| Ht- Ehv |  |  |
| Ehv - Wt |  |  |

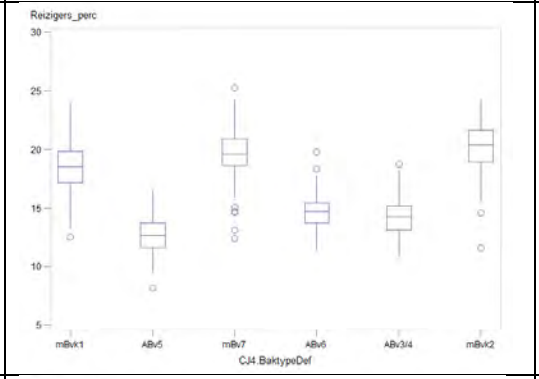
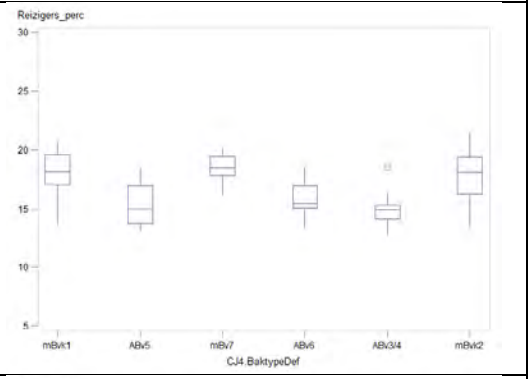
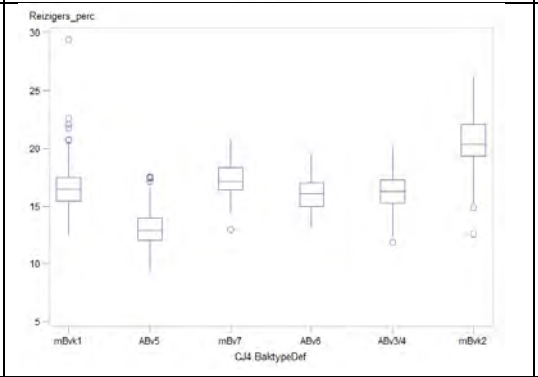
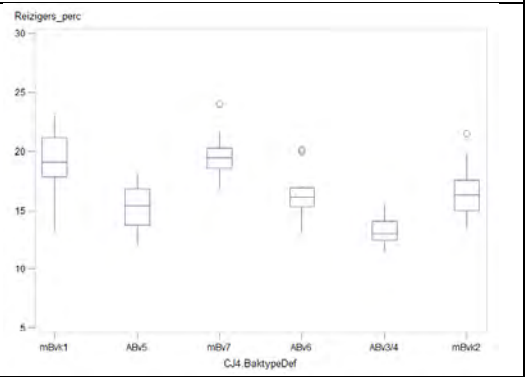
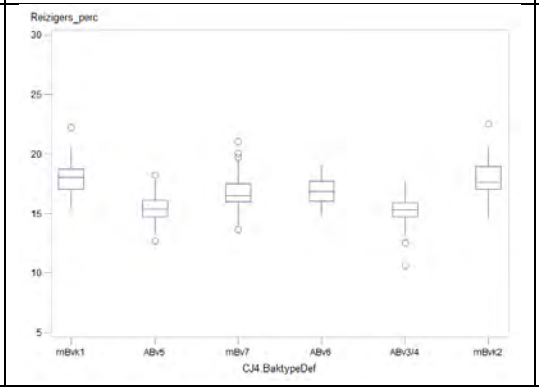
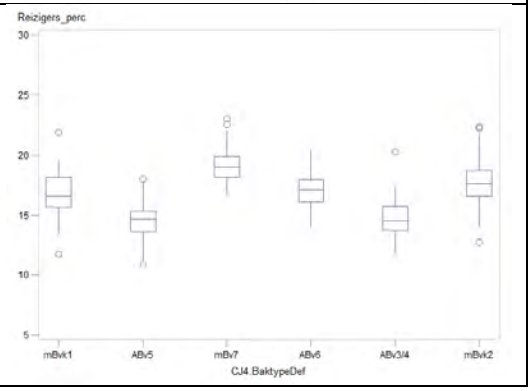


**Tabel 21: Reizigersverdeling van treinnummer 821 en 843 op meerdere trajecten**

Hier zien we opnieuw dat op het traject van ‘s-Hertogenbosch naar Eindhoven meer reizigers achter in de trein zitten. Vooral tijdens de spits is dit verschil beter te zien dan buiten de spits. Verder zijn de reizigers op het traject van Eindhoven naar Weert gelijkmatiger over de trein verdeeld dan op het traject van Utrecht Centraal naar ‘s-Hertogenbosch. Een verklaring hiervoor is dat het drukker is op het traject Utrecht Centraal – ‘s-Hertogenbosch en er een beperkt aantal 2$^{e}$ klas zitplaatsen is in de ABv-bakken.

### Reizigersverdeling op heen/terug weg

Tenslotte wordt in Tabel 22 ook gekeken hoe de verdeling is op de terugweg. Hierbij focussen we op de spits en nemen we alleen treinen waarbij de mBvk1 voorop rijden.

| + | Ut -> Wt | Wt -> Ut |
|---|---|---|
| (Ut, Ht) |  |  |
| (Ht, Ehv) |  |  |
| (Ehv, Wt) |  |  |

**Tabel 22: Reizigersverdeling op enkele trajecten (heen- en terugweg) tijdens de spits**

Nogmaals zien we dat op het traject van ‘s-Hertogenbosch naar Eindhoven meer reizigers achter in de trein zitten, met als mogelijke verklaring dat de achterste bakken van de trein het meest dichtbij de trappen liggen. Deze verklaring is meer aannemelijker wanneer we zien dat op de terugweg juist meer reizigers voorin de trein zitten. Op de terugweg van Eindhoven naar ‘s-Hertogenbosch stopt namelijk de voorste bak het meest dichtbij de trappen. Verder zien we op het traject Utrecht Centraal – ‘s-Hertogenbosch relatief meer reizigers in de motorbakken zitten dan op het traject ‘s-Hertogenbosch – Utrecht Centraal, waar men meer evenredig over de trein is verdeeld.

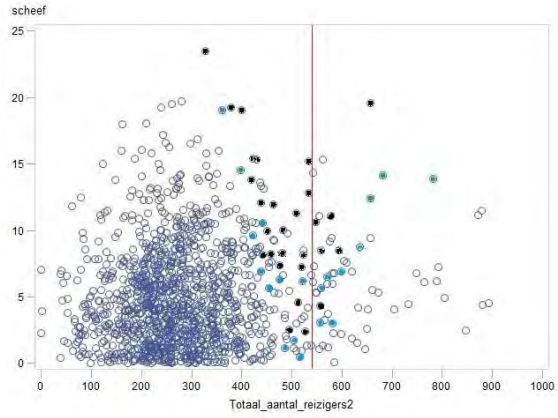
### Scheve reizigersverdeling

Uit het voorgaande zagen we dat het verschil tussen het reizigerspercentage in mBvk1 en mBvk2 het grootst is op het traject van ’s Hertogenbosch naar Eindhoven en andersom. Zo’n scheve reizigersverdeling in de trein is voornamelijk een probleem wanneer aan één kant van de trein reizigers moeten staan vanwege de drukte, terwijl aan de andere kant nog lege zitplaatsen beschikbaar zijn. Daarom wordt gekeken hoe vaak dit in de dataset voorkomt. De mate van scheefheid van een reizigersverdeling wordt bepaald door het verschil van het percentage reizigers in

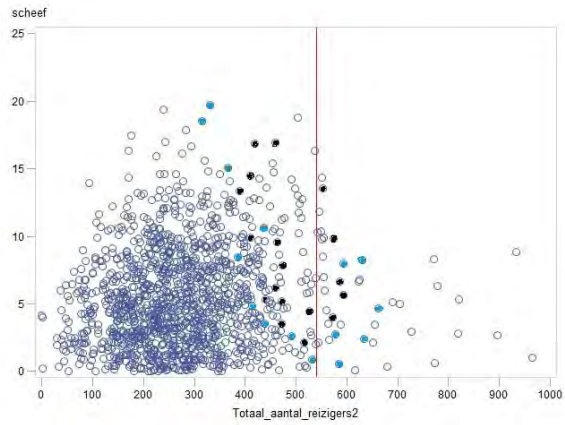
de mBvk1 met mBvk2 te nemen en deze op te tellen bij het verschil tussen het percentage reizigers in de ABv5 en de ABv3/4. In formule ziet het er als volgt uit:

$$scheef = \left|(percentage_{mBvk1} - percentage_{mBvk2}) + (percentage_{ABv5} - percentage_{ABv3/4})\right|$$

De mBvk1 wordt met mBvk2 vergeleken, omdat beide bakken hetzelfde aantal 1e en 2e klas zitplaatsen hebben. Hetzelfde geldt bij het vergelijken van de ABv5 met ABv3/4, omdat beide ongeveer hetzelfde aantal 1e en 2 klas zitplaatsen hebben. In Figuur 30 en Figuur 31 zijn voor de trajecten 's-Hertogenbosch – Eindhoven en Eindhoven – 's-Hertogenbosch de scheefheid van de reizigersverdelingen geplot t.o.v. het totaal aantal reizigers in de trein. De rode verticale lijn geeft de capaciteit van de trein aan. Er zijn in totaal 541 zitplaatsen in de VIRM-VI. Uit beide grafieken is te zien dat er veel ritten zijn met een scheve verdeling, met name wanneer het aantal reizigers kleiner is dan de capaciteit van de trein. Enkele ritten zijn gemarkeerd met zwart waarbij het aan één kant van de trein overvol is terwijl aan de andere kant nog plaatsen beschikbaar zijn. Ritten waarbij dat niet het geval is, is met licht blauw weergegeven. We zien dat de gevallen waarbij de ene bak overvol is terwijl in een andere bak nog plekken beschikbaar zijn, het aantal reizigers in de hele trein vrijwel tegen de capaciteit van de trein ligt. Het aantal reizigers per bak is berekend door het percentage van het reizigersgewicht uit Gotcha te vermenigvuldigen met de voorspelling van het aantal reizigers in de trein uit SOFA.




**Figuur 30: Scatterplot scheefheid van reizigersverdeling met het aantal reizigers op traject 's-Hertogenbosch - Eindhoven**




**Figuur 31: Scatterplot scheefheid van reizigersverdeling met het aantal reizigers op traject Eindhoven - 's-Hertogenbosch**

### 5.1.3 Conclusie

In paragraaf 5.1.2 is voor enkele attributen onderzocht wat voor invloed deze hebben op de reizigersverdeling in de trein. De reizigersverdeling blijkt onder andere afhankelijk te zijn van het traject waarop de trein rijdt en of de trein tijdens of buiten de spits rijdt. Daarnaast is het mogelijk dat de ligging van de trein ook een belangrijke rol speelt, dus of de mBvk1 of mBvk2 voorop rijdt. Tenslotte is ook gekeken hoe vaak het in de dataset voorkomt dat reizigers scheef zijn verdeeld in een drukke trein.

## 5.2 Methoden

Het doel van het onderzoek is om een model te implementeren dat de reizigersverdeling in de trein bij vertrek kan voorspellen. Er is echter voor gekozen om het reizigersgewicht in elk bak te voorspellen in plaats van het percentage reizigers in elke bak, omdat bij het voorspellen van percentages deze niet altijd tot 100% zullen optellen. Bij het voorspellen van de reizigersgewichten wordt deze restrictie omzeild en kunnen de gewichten per bak achteraf omgezet worden naar percentages.

### 5.2.1 Machine learning

De voorspellingsmethoden die in dit onderzoek worden toegepast, zijn afkomstig van een tak binnen de computerwetenschappen, genaamd machine learning. Met machine learning kunnen patronen in grote datasets ontdekt worden. Op basis daarvan kunnen algoritmes zelf leren en voorspellingen doen in de toekomst. Binnen machine learning wordt er onderscheid gemaakt tussen supervised learning en unsupervised learning [13]. Bij supervised learning zijn in de training set de input waarden waarop getraind kan worden, gekoppeld aan de output waarden. Met een model dat is toegepast op deze training set, kunnen voorspellingen van de output worden gedaan voor nieuwe input waarden. Bij unsupervised learning zijn er geen output waarden. Hier is de bedoeling om structuur in de data te ontdekken. De dataset in ons onderzoek is gelabeld. De output variabelen zijn namelijk de reizigersgewichten in elke bak. Daarom worden supervised learning algoritmes gebruikt in ons onderzoek. Supervised learning is verder onderverdeeld in classificatieproblemen en regressieproblemen. Bij classificatieproblemen zijn de output variabelen discreet, bijvoorbeeld het geslacht (man of vrouw). Bij regressieproblemen zijn deze variabelen continu, bijvoorbeeld het gewicht. Het voorspellen van het reizigersgewicht behoort dus tot een regressieprobleem. Verschillende supervised learning algoritmes zijn beschikbaar binnen machine learning.
Voor ons onderzoek moeten deze algoritmes geschikt zijn voor regressieproblemen, daarnaast heeft het de voorkeur als deze algoritmes in staat zijn om meerdere variabelen te voorspellen, waarbij rekening wordt gehouden met de interactie van de variabelen die worden voorspeld. In ons geval wordt het reizigersgewicht in zes bakken voorspeld.

**Multivariate Adaptive Regression Splines**

Multivariate adaptive regression splines (MARS) is een vorm van lineaire regressie dat voor het eerst werd geïntroduceerd door Jerome Friedman in 1991 [14]. MARS is een uitgebreider variant van een lineair regressie model dat in staat is om non-lineaire verbanden tussen afhankelijke en onafhankelijke variabelen te modelleren. Dit wordt gedaan door meerdere lineaire segmenten (ook wel basis functies genoemd) met verschillende richtingscoëfficiënten aan elkaar te koppelen. De overgang van de ene basis functie naar een andere basis functie wordt een knoop genoemd.
Het MARS model heeft de volgende vorm:

$$\hat{f}(x) = \sum_{i=1}^{k} c_i B_i(x)$$

Het model is een gewogen som van de basis functies $B_i(x)$ die vaak uit de volgende twee functies bestaan:

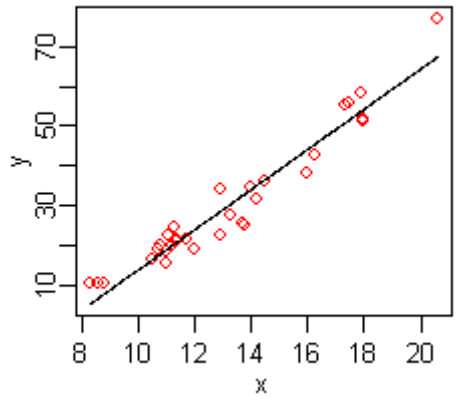
$$\max(0, k - x) \qquad \text{of de gespiegelde vorm} \qquad \max(0, x - k)$$

waar $k$ een knooppunt aanduidt. In Figuur 32 is een voorbeeld van een lineair model weergegeven waarbij variabele y wordt voorspeld aan de hand van variabele x [15]. Het model ziet er als volgt uit:

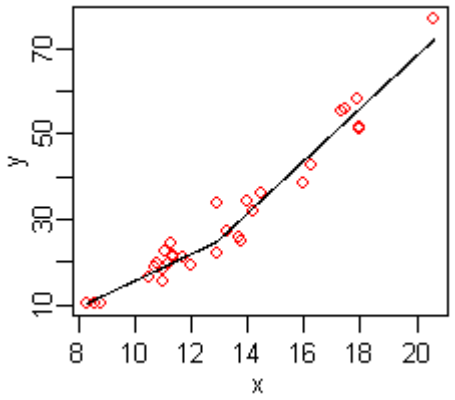
$y = -37 + 5.1x$. In Figuur 33 is met dezelfde data een MARS model toegepast met de volgende formule: $y = 25 + 6.1\max(0, x - 13) - 3.1\max(0, 13 - x)$ [15]. Het MARS model maakt gebruik van twee basisfuncties die van elkaar gespiegeld zijn om het non-lineair verband tussen x en y vast te stellen. Het model heeft automatisch het knooppunt in x = 13 aangemaakt.
Het MARS algoritme bestaat uit twee fases [16]. In de eerste fase voegt het algoritme basis functies in paren toe, die van elkaar zijn gespiegeld. Bij elke iteratie wordt de plaats van de knoop aangepast om de maximale reductie van de kwadratische fout te krijgen. Dit proces stopt wanneer het maximaal aantal termen in het model is bereikt of er te weinig verbetering in het model is. De eerste fase leidt vaak tot overfitting. Daarom worden in de tweede fase van het algoritme basis functies verwijderd die het minst toevoegen aan het model. Het verwijderen van de basis functies is gebaseerd op de Generalized Cross-Validation (GCV):

$$\mathrm{GCV} = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2}{[1 - \frac{M + d \times (M-1)/2}{N}]^2}$$

met $N$ het aantal observaties, $d$ de penalty voor elke basis functie en $M$ het aantal basis functies in het model. Deze maatstaf houdt dus ook rekening met de complexiteit van het model. Om het reizigersgewicht in alle bakken van VIRM-VI te voorspellen, worden zes MARS-modellen getraind worden die uit dezelfde verzameling van basis functies bestaan. Het algoritme bepaalt de optimale verzameling van basis functies dat de som van de GCV's over de modellen minimaliseert.




Figuur 32: Voorbeeld lineair model

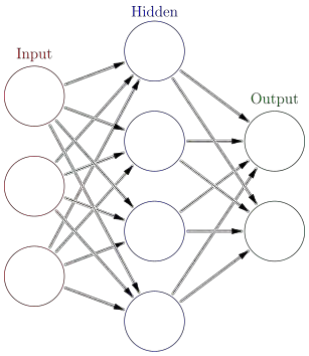



Figuur 33: Voorbeeld MARS model

Het voordeel van MARS is dat het geen specifiek verband vereist tussen afhankelijke en onafhankelijke variabelen. Daarnaast selecteert het MARS algoritme automatisch attributen die significant zijn en is het trainen van het algoritme redelijk snel.
Het MARS algoritme is toegepast in R met de package 'earth' [17].

**Neurale netwerken**

Neurale netwerken (NN) zijn gebaseerd op de werking van het zenuwstelsel, waar neuronen een belangrijke rol spelen. Deze techniek wordt vaak toegepast bij problemen, zoals gezichts- en spraakherkenning. Een veelgebruikte vorm van een neuraal netwerk is de multilayer perceptron (MLP) [18]. MLP bestaat uit een laag input neuronen, een laag output neuronen en één of meerdere verborgen lagen die met synapsen aan elkaar verbonden zijn. In Figuur 34 is een voorbeeld van een neuraal netwerk weergegeven [19].




Figuur 34: Voorbeeld neuraal netwerk

Hoe meer verborgen lagen worden gebruikt, des te complexer het model wordt. Het aantal input neuronen is gelijk aan het aantal onafhankelijke variabelen in het model en het aantal output neuronen is gelijk aan het aantal afhankelijke variabelen. Aan elk synaps is een gewicht toegewezen dat de sterkte van een bijhorend neuron aangeeft. Berekeningen worden uitgevoerd met behulp van activerende functies in de verborgen lagen en de output laag. De volgende functie wordt berekend bij een MLP met één output neuron, $n$ input neuronen en één verborgen laag van $J$ neuronen [20]:

$$o(\boldsymbol{x}) = f(w_0 + \sum_{j=1}^{J} w_j \cdot f(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i))$$

waar $w_j$ het gewicht van de synaps is die neuron $J$ met de output neuron verbindt en $w_{ij}$ de gewichten van de synapsen zijn die richting neuron $J$ bijeenkomen. Voorbeelden van activerende functies voor $f$ zijn [20]:

$$f(\boldsymbol{x}) = \tanh(\boldsymbol{x}) \quad \text{en} \quad f(\boldsymbol{x}) = (1 + e^{-\boldsymbol{x}})^{-1}$$

Het leerproces in een MLP netwerk gebeurt door middel van het continu aanpassen van de gewichten van de neuronen, zodat het verschil tussen de output en de werkelijke waarden minimaal is. Dit wordt ook wel backpropagation genoemd. Om het minimum van de kost functie te bepalen wordt de gradiënt van de partieel afgeleide van de kost functie t.o.v. de gewichten berekend. Wanneer er sprake is van een negatief gradiënt wordt het gewicht verhoogd. Bij een positief gradiënt wordt deze verlaagd. Op deze manier kan een lokaal minimum van de kost functie bereikt worden. De gewichten veranderen met de volgende formule bij een standaard backpropagation algoritme [20]:

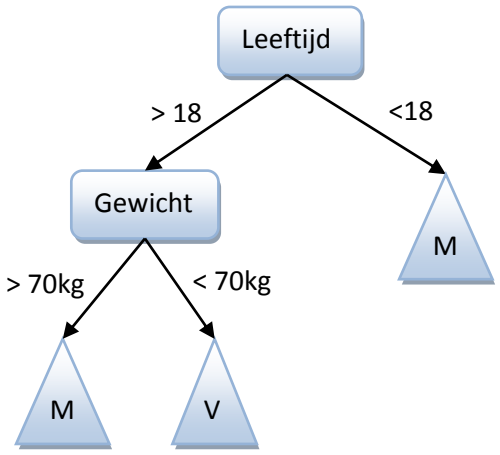
$$w_k^{(t+1)} = w_k^{(t)} - \eta \cdot \frac{\delta E^{(t)}}{\delta w_k^{(t)}},$$

waar $t$ de iteratie stap aangeeft en $k$ de gewichten. $E$ staat voor de kost functie en $\eta$ is de learning rate. Om het algoritme sneller te laten convergeren kan gekozen worden voor een hoge learning rate. Het risico hiervan is dat het lokaal minimum gemist kan worden en er geen convergentie plaats vindt.

Het trainen van een neuraal netwerk is in het algemeen langzamer dan andere algoritmes en wordt vaak als een 'black box' model beschouwd, maar is wel in staat om complexe verbanden tussen afhankelijke en onafhankelijke variabelen te leren.
MLP is toegepast in R met de package 'nnet' [21].

**Conditional Inference Trees**

Conditional inference trees (CIT) hebben veel gemeen met de normale beslissingsbomen, waar de uitkomst van de onafhankelijke variabele afhangt van de keuzes die worden gemaakt in de beslissingsboom. In Figuur 35 is een voorbeeld van een beslissingsboom weergegeven, waarin het geslacht van iemand wordt voorspeld aan de hand van leeftijd en gewicht. In de knopen die met rechthoeken zijn weergegeven worden de attributen getest. Op basis daarvan wordt de boom doorlopen tot een blad is bereikt waar een bepaalde voorspelling wordt gedaan. De voorspellingen zijn met driehoeken weergegeven. Dit voorbeeld is een classificatie probleem. Het is ook mogelijk om een regressie probleem te modelleren met een beslissingsboom, waarin de uitkomsten continue waarden zijn.




Figuur 35: Voorbeeld beslissingsboom

Afhankelijk van het algoritme kan de beslissingsboom op veel manieren gesplitst worden. Een knoop kan bijvoorbeeld opsplitsen in meer dan twee knopen. Daarnaast bepaalt het algoritme welk attribuut wordt gebruikt bij het splitsen en hoe groot de boom kan groeien.

Bij CIT worden de knopen binair gesplitst en het splitsen is gebaseerd op een permutatie toets. Dit houdt in dat onder de nulhypothese de paren van de gegroepeerde data $(X_i, Y_i)$ omgewisseld kunnen worden. Voor alle permutaties van $X$ en $Y$ wordt de volgende toetsingsgrootheid berekent voor variabele $j$ [22]:

$$T_j = \sum_{i=1}^{n} w_i g_j(X_{ij}) h\big(Y_i, (Y_1, \dots, Y_n)\big),$$

met $n$ het aantal observaties. $w = 1$ als de observatie in de huidige partitie van de boom zit, anders $w = 0$. $g$ en $h$ zijn transformatiefuncties. Het attribuut met de laagste p-waarde wordt gebruikt om te splitsen. Wanneer de nulhypothese voor geen enkele attribuut verworpen kan worden, wordt de knoop niet verder gesplitst. CIT selecteert in elke knoop het attribuut met het grootste effect op de onafhankelijke variabele om te splitsen in tegenstelling tot bijvoorbeeld CART [23], waar attributen die op meerdere manieren kan worden gesplitst eerder geselecteerd worden.

Beslissingsbomen zijn één van de meest inzichtelijke modellen binnen machine learning. Daarnaast is het trainen van bomen redelijk snel. Conditional inference trees wordt in R toegepast met de package ‘partykit’ [24].

### 5.2.2 Model evaluatie

**Kost functie**

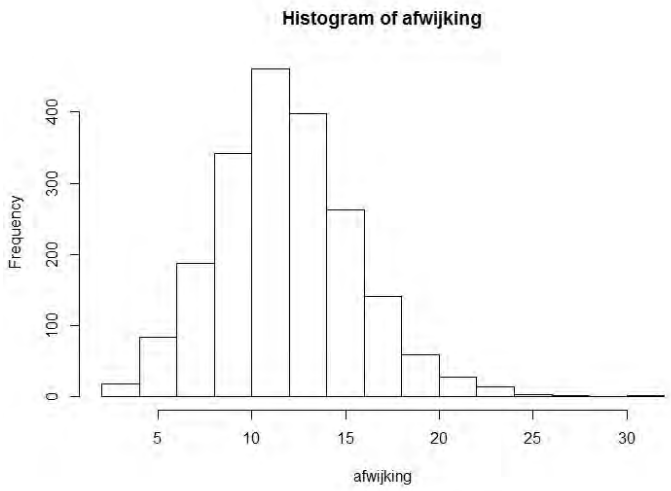
Een belangrijke vraag is hoe de voorspellingen van een model moeten worden beoordeeld. Zoals eerder vermeld, wordt het reizigersgewicht voorspeld in elke bak in plaats van het percentage reizigers. De algoritmes uit paragraaf 5.2.1 zullen hierdoor het verschil tussen het Gotcha reizigersgewicht en het voorspeld reizigersgewicht in een bak minimaliseren. Dit betekent dus niet automatisch dat het verschil in reizigerspercentage per bak tussen Gotcha en de voorspelling geminimaliseerd wordt. Bij het evalueren van de modellen moet echter gekeken worden of het bepaald percentage reizigers in elke bak overeenkomt met de reizigersverdeling uit Gotcha.
Door het reizigersgewicht in alle bakken tegelijk te voorspellen zal mogelijk het verschil in reizigerspercentage per bak tussen Gotcha en de voorspelling ook geminimaliseerd worden.

Een simpele kost functie kan verkregen worden door voor elk observatie de absolute verschillen van de voorspelde percentages met de percentages uit Gotcha te sommeren en hiervan het gemiddelde te nemen. Deze functie is hieronder als formule weergegeven.

$$Kost\ functie = \frac{\sum_{i=1}^{n}\sum_{j=1}^{6}|voorspelde\ verdeling[i,j] - Gotcha\ verdeling[i,j]|}{n}$$

waar $voorspelde\ verdeling[i,j]$ het voorspelde percentage is van de reizigers dat in bak $j$ zit van observatie $i$.

Er kan overwogen worden om een extra penalty te geven aan reizigersverdelingen met veel variatie tussen de bakken, omdat het belangrijker is grote verschillen tussen de bakken accuraat te kunnen voorspellen. Als maatstaf voor de variatie van het percentage reizigers tussen de bakken, wordt gekeken in hoeverre deze reizigersverdeling afwijkt van de evenwichtsverdeling waarbij in elke bak 100/6 ≈ 16,7% van de reizigers in elke bak zitten. Het aantal zitplaatsen in elke bak is ongeveer gelijk en we maken hierbij geen onderscheid tussen 1$^{e}$ en 2$^{e}$ klas reizigers. De afwijking van elk observatie wordt bepaald door de absolute verschillen van het percentage reizigers in elke bak met 16,7% te sommeren. De afwijking van alle observaties in de dataset is een histogram weergegeven in Figuur 36.




Figuur 36: Verdeling van de afwijking van de evenwichtsverdeling in percentages

In Tabel 23 is de verdeling met de kleinste, grootste en de gemiddelde afwijking van het evenwicht weergegeven.

| | Afwijking | mBvk1 | ABv5 | mBv7 | ABv6 | ABv3/4 | mBvk2 |
|---|---|---|---|---|---|---|---|
| Minimum | 1,2% | 16,5% | 16.3% | 16.8% | 17.0% | 16.7% | 16.7% |
| Gemiddeld | 11,4% | 18.5% | 16.1% | 19.7% | 14.9% | 13.3% | 17.5% |
| Maximum | 30,9% | 14.1% | 9.3% | 22.3% | 13.9% | 13.9% | 26.5% |

**Tabel 23: Reizigersverdelingen met minimale, gemiddelde en maximale afwijking van evenwichtsverdeling**

De reizigersverdeling met een gemiddelde afwijking verschilt niet veel met de evenwichtsverdeling. Gemiddeld wijkt het percentage reizigers per bak ongeveer 2% af van het evenwicht. In de meeste gevallen valt het niet op dat in de ene bak relatief veel meer reizigers zitten dan in een ander bak. De verdelingen met de grootste afwijkingen zijn het meest interessant, omdat daar de verschillen in percentages duidelijker zijn. Dus in principe is het belangrijker dat het model verdelingen met grote afwijking goed kan voorspellen. Daarom is de volgende kost functie gebruikt voor het evalueren van het model:

$$Kost\ functie = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{6}|Gotcha\ verdeling[i,j] - \frac{100}{6}|) * (\sum_{j=1}^{6}(voorspelde\ verdeling[i,j] - Gotcha\ verdeling[i,j])^{2})}{n}$$

Het verschil met de vorige kost functie is dat de verschillen tussen de voorspelde en Gotcha verdeling in het kwadraat wordt genomen zodat grote verschillen zwaarder worden gewogen. Dit lijkt veel op de zogenaamde mean squared error. Daarna wordt dit vermenigvuldigd met hoeveel de Gotcha verdeling afwijkt van de evenwichtsverdeling en als laatst wordt het gemiddelde over alle observaties genomen.

**Cross-validation**

Cross-validation is een methode om modellen te valideren. Het doel van cross-validation is om overfitting te beperken, waarbij een model slechts goed presteert op de training set, maar niet op nieuwe data. Bij het toepassen van de modellen op de dataset van 5152 observaties wordt gebruik gemaakt van 5-fold cross-validation. Dit houdt in dat de dataset wordt gesplitst in vijf groepen observaties, waar bij elke iteratie een andere groep gebruikt wordt als test set en de overige groepen als training set.

## 5.3 Attributen selectie

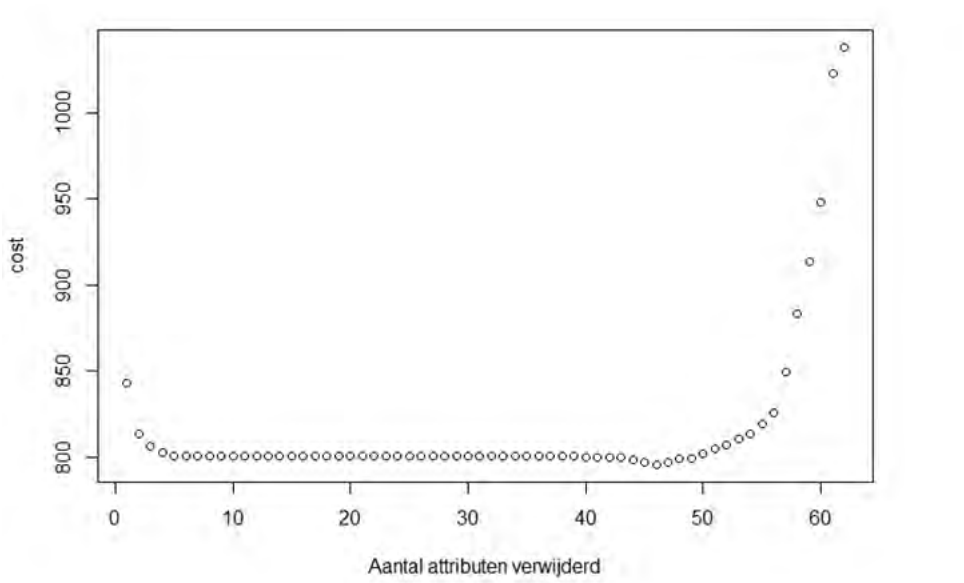
In totaal zijn er 62 attributen beschikbaar voor het implementeren van de modellen. Het gebruik van alle attributen kan echter leiden tot onnodig complexe modellen. Daarom is het van belang om attributen te verwijderen die geen toegevoegde waarde hebben voor de modellen of zelfs leiden tot slechtere voorspellingen. Twee manieren zijn toegepast voor het selecteren van attributen, die hieronder worden besproken.

### 5.3.1 Backward elimination op basis van laagste kosten van model

Eerst wordt multivariate adaptive regression splines toegepast op de dataset met 5-fold cross-validation. Het model wordt getraind met alle attributen uit de dataset. Vervolgens wordt opnieuw cross-validation toegepast met telkens één attribuut weggelaten. Het model met de laagste gemiddelde kosten uit de cross-validation wordt geselecteerd. Dit proces wordt herhaald tot er slechts één attribuut overblijft. Daarna wordt het model met de laagste kosten geselecteerd uit de iteraties. In Figuur 37 is te zien dat de beste score is bereikt na het verwijderen van 46 attributen. De overgebleven attributen die gebruikt waren bij het model worden dan geselecteerd en deze zijn hieronder weergegeven.

- *Huidig station*
- *Richting traject*
- *Aantal reizigers bij vertrek*
- *Temperatuur*
- *Het reizigersgewicht per bak voor aankomst (behalve mBv7)*
- *Het reizigerspercentage per bak voor aankomst (behalve mBvk1)*
- *Het gemiddelde percentage reizigers per bak bij vertrek in het verleden (behalve mBvk1)*

In de laatste drie groepen attributen is steeds het attribuut van één bak niet geselecteerd. Aangezien de attributen voor vijf van de zes bakken geselecteerd zijn, hebben we het ontbrekend attribuut ook meegenomen voor consistentie.




**Figuur 37: Kosten van model bij backwards elimination op basis van laagste kosten**

### 5.3.2 Backward elimination op basis van correlatie met reizigersverdeling

Met deze methode wordt eerst de correlatie van alle attributen met de reizigersverdeling bij vertrek bepaald. Voordat de correlatie kan worden berekend, worden enkele categorische variabelen omgezet naar binaire variabelen. De transformaties worden hieronder beschreven:

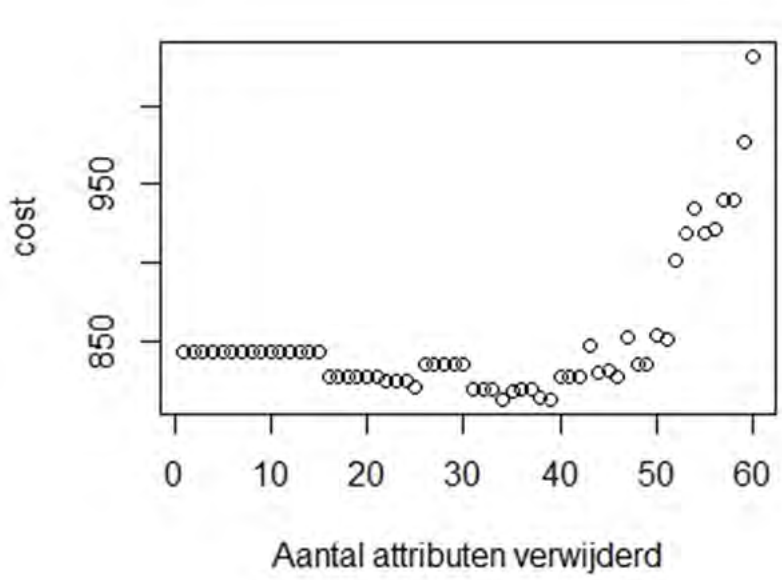
*Uur*
Dit wordt omgezet naar een binaire variabele die aangeeft of de metingen in de ochtendspits (07:00-09:00) plaatsvinden.

*Dag van de week*
Dit wordt omgezet naar een variabele die aangeeft of het een doordeweekse dag is of het weekend is.

*Maand*
Hiervan wordt een variabele gemaakt die aangeeft of het een zomermaand (juli & augustus) is of niet.

Daarna wordt de correlatie van het percentage reizigers in elke bak met de onafhankelijke variabelen berekend. Per attribuut worden de absolute waarden van de correlatiecoëfficiënten bepaald waarna het gemiddelde over de bakken wordt genomen. Deze kunnen dan van hoog naar laag gesorteerd worden, waarbij een hoge waarde met een hoge correlatie correspondeert.

Vervolgens wordt op dezelfde manier als in paragraaf 5.3.1 backward elimination met MARS toegepast, maar bij elke iteratie wordt telkens het attribuut met de minste correlatie verwijderd. Dit herhaalt zich tot er slechts één attribuut overblijft om mee te trainen.




**Figuur 38: Kosten van model met attributen selectie op basis van correlatie met reizigersverdeling**

In Figuur 38 is het minimum bereikt na het verwijderen van 39 attributen. De resterende attributen zijn hieronder weergegeven:

- *Huidig station*
- *Richting traject*
- *Richting trein*
- *Het reizigersgewicht per bak voor aankomst (behalve ABv3/4)*
- *Het reizigerspercentage per bak voor aankomst (behalve mBvk1)*
- *Het gemiddelde percentage reizigers per bak bij vertrek in het verleden (behalve mBv7 en ABv6)*

Daarnaast behoort ook het gemiddeld percentage reizigers bij aankomst in het verleden voor mBvk1 en mBv7 tot de lijst van belangrijke attributen. Deze hebben we echter van de lijst verwijderd, omdat het gemiddeld percentage reizigers bij aankomst in het verleden slechts voor twee bakken relevant is.

Zowel uit Figuur 37 en Figuur 38 blijkt dat een goed model ten minste rond 10 attributen moet bevatten. Verder leverden beide attributen selectie methoden bijna dezelfde variabelen op. Het enig verschil is dat bij de eerste methode nog gekozen is voor het aantal reizigers bij vertrek en de temperatuur, terwijl bij backward elimination op basis van correlatie met reizigersverdeling blijkt dat de richting van de trein een belangrijke rol speelt.

### 5.3.3 Optimaal attributen verzameling

In paragraaf 5.3.1 en 5.3.2 waren twee attributen selectie methoden toegepast. De attributen die in beide methoden zijn geselecteerd, worden gebruikt voor het voorspellingsmodel. Attributen die slechts door één van de attributen selectie methoden zijn geselecteerd, worden opnieuw getest of deze daadwerkelijk significant zijn voor het model. Het gaat om de volgende drie attributen:

- *Aantal reizigers bij vertrek*
- *Temperatuur*
- *Richting trein*

Hiervoor wordt opnieuw MARS modellen met 5-fold cross-validation toegepast met de attributen die in beide selectie methoden zijn geselecteerd plus een combinatie van bovenstaande attributen. Hiermee wordt onderzocht of het toevoegen van het aantal reizigers bij vertrek, richting van de trein en de temperatuur verbetering oplevert voor het model. De kosten van deze modellen zijn weergegeven in Tabel 24.

| *+ richting trein* | *+ temperatuur & aantal reizigers bij vertrek* | Kosten |
|---|---|---|
| Nee | Nee | 808,07 |
| Ja | Nee | 769,91 |
| Nee | Ja | 815,74 |
| Ja | Ja | 776,43 |

**Tabel 24: Kosten van model bij enkele combinaties van attributen**

De attributen *temperatuur & aantal reizigers bij vertrek* zijn samengevoegd, omdat beide attributen uit dezelfde selectie methode kwamen. Het valt op dat de richting van de trein een positieve bijdrage levert aan het model, terwijl het toevoegen van de temperatuur en het aantal reizigers bij vertrek tot hogere kosten leiden. Uiteindelijk worden de volgende attributen gebruikt voor het trainen van de modellen:

- *Huidig station*
- *Richting traject*
- *Richting trein*
- *Het gewicht en percentage per bak voor aankomst (12x)*
- *Het gemiddelde percentage reizigers per bak bij vertrek in het verleden (6x)*

Merk op dat deze attributen dezelfde zijn als de attributen die uit de correlatiegebaseerde selectie methode kwamen. In totaal zijn het er 21. De geselecteerde attributen zijn grotendeels naar verwachting. Ten eerste is het belangrijk om het percentage reizigers per bak voor aankomst te weten om de verdeling van de reizigers bij vertrek te kunnen voorspellen, omdat de verdeling bij

vertrek afhankelijk is van hoe reizigers op het vorig traject zijn verdeeld in de trein en hoeveel reizigers er per bak in- en uitstappen. Daarnaast geeft het gewicht per bak voor aankomst een indicatie van hoe druk het is in elke bak op het vorig traject. Met de combinatie van het huidig station en de richting van het traject kan per traject onderscheid gemaakt worden in de verdeling van de reizigers. Daarna zagen we eerder dat de verdeling van reizigers anders is wanneer de trein omgedraaid is. Als laatst geeft het gemiddeld percentage reizigers per bak bij vertrek in het verleden een indicatie hoe de verdeling in het verleden was op hetzelfde traject rond dezelfde tijd en dagdeel. In Tabel 25 zijn deze attributen weergegeven met de afkortingen die zijn gebruikt in de dataset.

| Attribuut | Afkorting | Waarden |
|---|---|---|
| *Onafhankelijke variabelen* | | |
| *Huidig station* | V2 | (0 = 's-Hertogenbosch, 1 = Eindhoven) |
| *Richting traject* | Oneven | (0 = Richting Utrecht, 1 = Richting Weert) |
| *Richting trein* | Richting | (0 = mBvk1 achterste bak, 1 = mBvk1 voorste bak) |
| *Reizigersgewicht mBvk1 voor aankomst* | Gotcha_1_gew1 | In kg |
| *Reizigersgewicht ABv5 voor aankomst* | Gotcha_5_gew1 | In kg |
| *Reizigersgewicht mBv7 voor aankomst* | Gotcha_7_gew1 | In kg |
| *Reizigersgewicht ABv6 voor aankomst* | Gotcha_6_gew1 | In kg |
| *Reizigersgewicht ABv34 voor aankomst* | Gotcha_34_gew1 | In kg |
| *Reizigersgewicht mBvk2 voor aankomst* | Gotcha_2_gew1 | In kg |
| *Reizigerspercentage mBvk1 voor aankomst* | Gotcha_1_perc1 | (0% - 100%) |
| *Reizigerspercentage ABv5 voor aankomst* | Gotcha_5_ perc1 | (0% - 100%) |
| *Reizigerspercentage mBv7 voor aankomst* | Gotcha_7_ perc1 | (0% - 100%) |
| *Reizigerspercentage ABv6 voor aankomst* | Gotcha_6_ perc1 | (0% - 100%) |
| *Reizigerspercentage ABv34 voor aankomst* | Gotcha_34_ perc1 | (0% - 100%) |
| *Reizigerspercentage mBvk2 voor aankomst* | Gotcha_2_ perc1 | (0% - 100%) |
| *Reizigerspercentage mBvk1 bij vertrek in verleden* | ma_1_perc2 | (0% - 100%) |
| *Reizigerspercentage ABv5 bij vertrek in verleden* | ma _5_ perc2 | (0% - 100%) |
| *Reizigerspercentage mBv7 bij vertrek in verleden* | ma _7_ perc2 | (0% - 100%) |
| *Reizigerspercentage ABv6 bij vertrek in verleden* | ma _6_ perc2 | (0% - 100%) |
| *Reizigerspercentage ABv34 bij vertrek in verleden* | ma _34_ perc2 | (0% - 100%) |
| *Reizigerspercentage mBvk2 bij vertrek in verleden* | ma _2_ perc2 | (0% - 100%) |
| *Afhankelijke variabelen* | | |
| *Reizigersgewicht mBvk1 bij vertrek* | Gotcha_1_gew2 | In kg |
| *Reizigersgewicht ABv5 bij vertrek* | Gotcha_5_gew2 | In kg |
| *Reizigersgewicht mBv7 bij vertrek* | Gotcha_7_gew2 | In kg |
| *Reizigersgewicht ABv6 bij vertrek* | Gotcha_6_gew2 | In kg |
| *Reizigersgewicht ABv34 bij vertrek* | Gotcha_34_gew2 | In kg |
| *Reizigersgewicht mBvk2 bij vertrek* | Gotcha_2_gew2 | In kg |

**Tabel 25: Attributen**

# 6 Resultaten

In dit hoofdstuk worden de modellen getraind met de attributen die zijn verkregen uit de attributen selectie methoden, waarna 5-fold cross-validation wordt gebruikt om de modellen te evalueren. Daarna worden de modellen ook getraind op de volledige dataset, zodat de modellen visueel weergegeven kunnen worden en gekeken kan worden welke attributen in de modellen een grote rol spelen. Om aan te tonen dat de machine learning algoritmes toegevoegde waarde hebben, is er een benchmark vastgesteld. Als benchmark wordt de reizigersverdeling bij vertrek in het verleden gebruikt. De voorspelde verdelingen van de modellen kan dan vergeleken worden met de benchmark verdelingen.

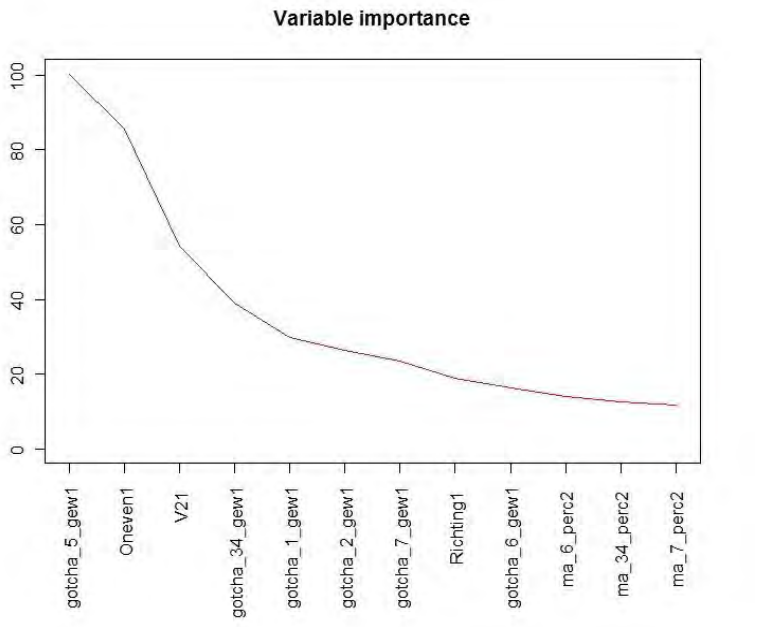
## 6.1 Resultaat Multivariate adaptive regression splines

Bij het toepassen van MARS in R zijn de standaard parameter instellingen gebruikt. Onder andere dat tijdens de eerste fase van het MARS algoritme maximaal 20 termen toegevoegd kan worden aan het model en het minimum aantal observaties tussen knopen gelijk is aan 0. In Tabel 26 zijn de kosten van de MARS modellen in elke fold weergegeven. De gemiddelde kosten hiervan is gelijk aan 732,31. De rekentijd is gelijk aan 4 seconden.

| fold | 1 | 2 | 3 | 4 | 5 | Gemiddeld |
|---|---|---|---|---|---|---|
| Kosten | 753,84 | 642,87 | 735,54 | 720,42 | 808,87 | 732,31 |

Tabel 26: Kosten van MARS met 5-fold cross-validation

Daarna is een MARS model getraind op de hele dataset. Het MARS algoritme selecteert automatisch de beste attributen waarmee het model wordt getraind. Om de significantie van de attributen beter te kunnen vergelijken wordt eerst het verschil in GCV (Generalized Cross-Validation) tussen elke subset van attributen met de subset uit de vorige iteratie tijdens de opbouwfase van het MARS algoritme bepaald [25]. De reductie in GCV na het toevoegen van een bepaald attribuut is in Figuur 39 weergegeven. Deze waarden zijn genormaliseerd naar waarden tussen 0 en 100.



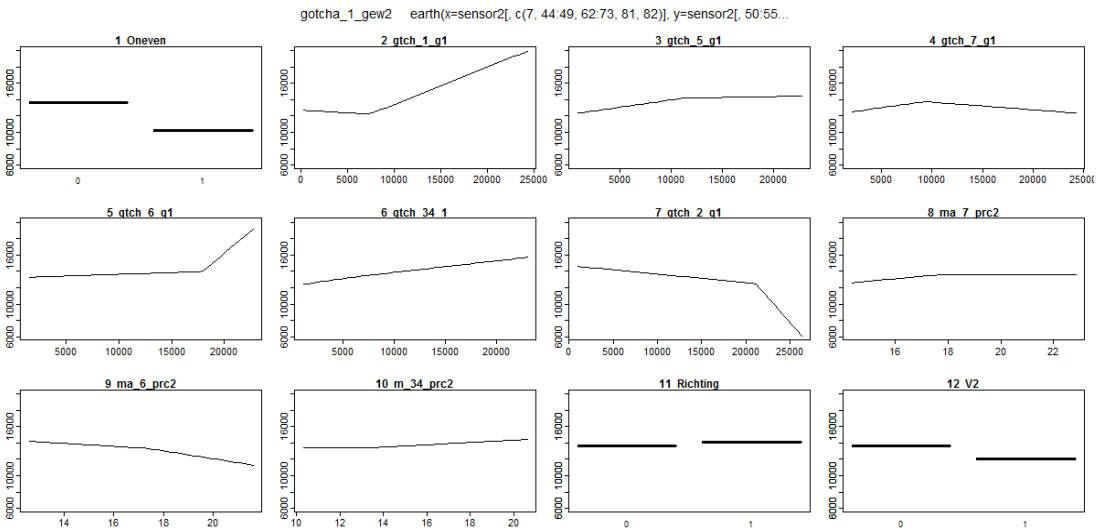

Figuur 39: Significantie variabelen in MARS

Uit de grafiek is te zien dat het reizigersgewicht in ABv5 bij aankomst het meeste effect heeft op het model. Daarna volgen de richting van het traject en het huidig station. Het model heeft geen gebruik gemaakt van het reizigerspercentage bij aankomst, omdat het weinig toegevoegde waarde heeft voor het model.

In Figuur 40 is te zien wat voor effect elk attribuut in het model heeft op het reizigersgewicht in mBvk1 bij vertrek, wanneer de overige attributen constant worden gehouden. Hieronder is ook de bijhorende regressie formule weergegeven voor het reizigersgewicht in de mBvk1. De invloed van de attributen op het reizigersgewicht bij vertrek in de overige bakken en de regressie formules voor het reizigersgewicht per bak is in Appendix A weergegeven. De regressie formules bestaan uit dezelfde basis functies en knooppunten. Het enig verschil is dat per model andere gewichten zijn toegewezen aan de basis functies.

In Figuur 40 zien we bijvoorbeeld in de grafiek van het attribuut 'Oneven' dat wanneer de trein richting Weert rijdt er minder reizigers in de mBvk1 zitten. Ook zien we dat het reizigersgewicht in mBvk1 bij vertrek toeneemt, wanneer het reizigersgewicht in de mBvk1, ABv5, ABv6 en ABv3/4 bij aankomst stijgt. Aan de andere kant leidt een toename van het reizigersgewicht in mBv7 bij aankomst tot een daling van het reizigersgewicht in mBvk1 bij vertrek.

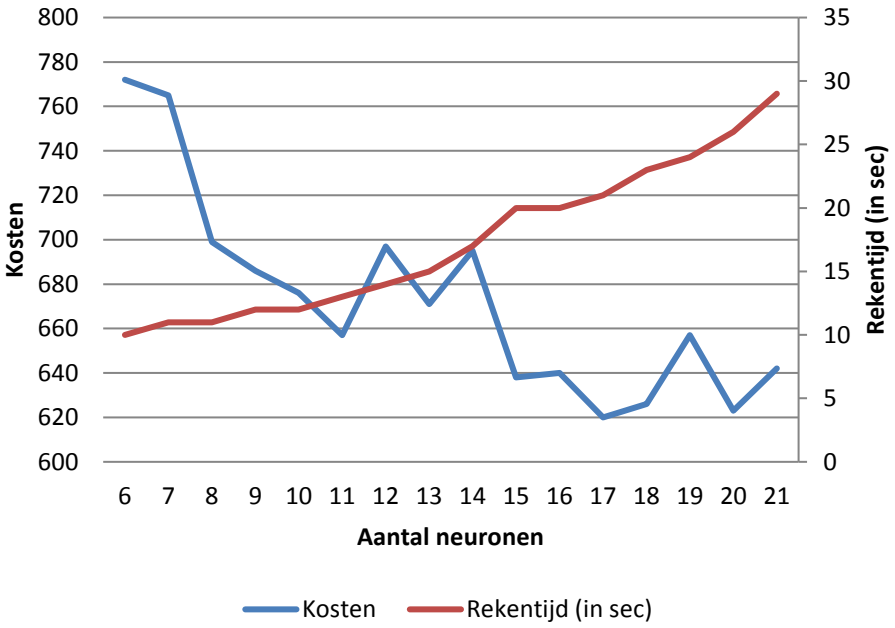
```
gotcha_1_gew2 (reizigersgewicht mBvk1) =
  1.2e+04
  -    3311 * Oneven1
  +     551 * Richting1
  -    1517 * V21
  +   0.063 * max(0,           7400 -  gotcha_1_gew1)
  +    0.46 * max(0,  gotcha_1_gew1 -           7400)
  -    0.18 * max(0,          11188 -  gotcha_5_gew1)
  +   0.022 * max(0,  gotcha_5_gew1 -          11188)
  -    0.17 * max(0,           9284 -  gotcha_7_gew1)
  -   0.094 * max(0,  gotcha_7_gew1 -           9284)
  -   0.042 * max(0,          17860 -  gotcha_6_gew1)
  +     1.1 * max(0,  gotcha_6_gew1 -          17860)
  -    0.18 * max(0,           6858 - gotcha_34_gew1)
  +    0.14 * max(0, gotcha_34_gew1 -           6858)
  +     0.1 * max(0,          21033 -  gotcha_2_gew1)
  -     1.2 * max(0,  gotcha_2_gew1 -          21033)
  -     290 * max(0,             18 -     ma_7_perc2)
  +     178 * max(0,             17 -     ma_6_perc2)
  -     477 * max(0,     ma_6_perc2 -             17)
  +     142 * max(0,    ma_34_perc2 -             13)
```




**Figuur 40: Invloed van variabelen op het reizigersgewicht in mBvk1 bij vertrek**

## 6.2 Resultaat neuraal netwerk

Bij het trainen van een neuraal netwerk moet een keuze gemaakt worden voor het aantal verborgen lagen en het aantal neuronen per verborgen laag. Het is belangrijk dat het model zo simpel mogelijk moet zijn. Voor de meeste problemen is één verborgen laag voldoende. Het gebruik van twee verborgen lagen zal nauwelijks verbetering opleveren en het vergroot de kans dat het netwerk naar een lokaal minimum convergeert [26]. Naast het aantal verborgen lagen is het aantal neuronen in een verborgen laag net zo belangrijk. Bij te weinig neuronen is het netwerk niet in staat om alle patronen te ontdekken in de data. Te veel neuronen kan weer leiden tot overfitting en toename van de tijd om het netwerk te trainen. Er bestaan verschillende regels voor het bepalen van het beste aantal verborgen neuronen. Één van de regels is dat dit aantal ligt tussen het aantal neuronen in de input laag en het aantal in de output laag [27]. Daarom wordt 5-fold cross-validation toegepast op neurale netwerken met 6 t/m 21 neuronen in één verborgen laag. Hierbij wordt de logistische functie gebruikt als activerende functie. Het leeralgoritme is het standaard backpropagation met een learning rate van 0,1 en het maximum aantal iteraties is gelijk aan 100. De kosten van de getrainde neurale netwerken met verschillend aantal neuronen zijn weergegeven in Figuur 41. Hierin is ook de rekentijd van elk netwerk aangegeven. 5-fold cross-validation met het netwerk van 17 neuronen in de verborgen laag leverde de laagste kosten op. De kosten van het model in elke fold zijn in Tabel 27 weergegeven. De rekentijd hiervan is gelijk aan 21 seconden.



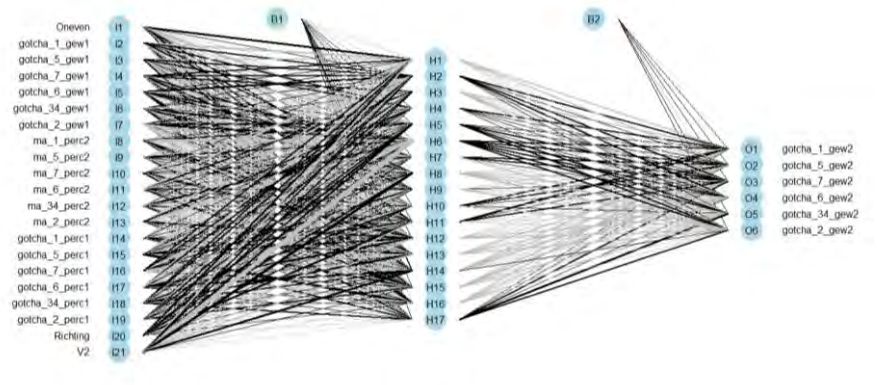

Figuur 41: Kosten en rekentijd van NN bij verschillend aantal neuronen

| fold | 1 | 2 | 3 | 4 | 5 | Gemiddeld |
|---|---|---|---|---|---|---|
| Kosten | 651,88 | 608,91 | 588,88 | 577,70 | 673,72 | 620,22 |

Tabel 27: Kosten van NN met 5-fold cross-validation

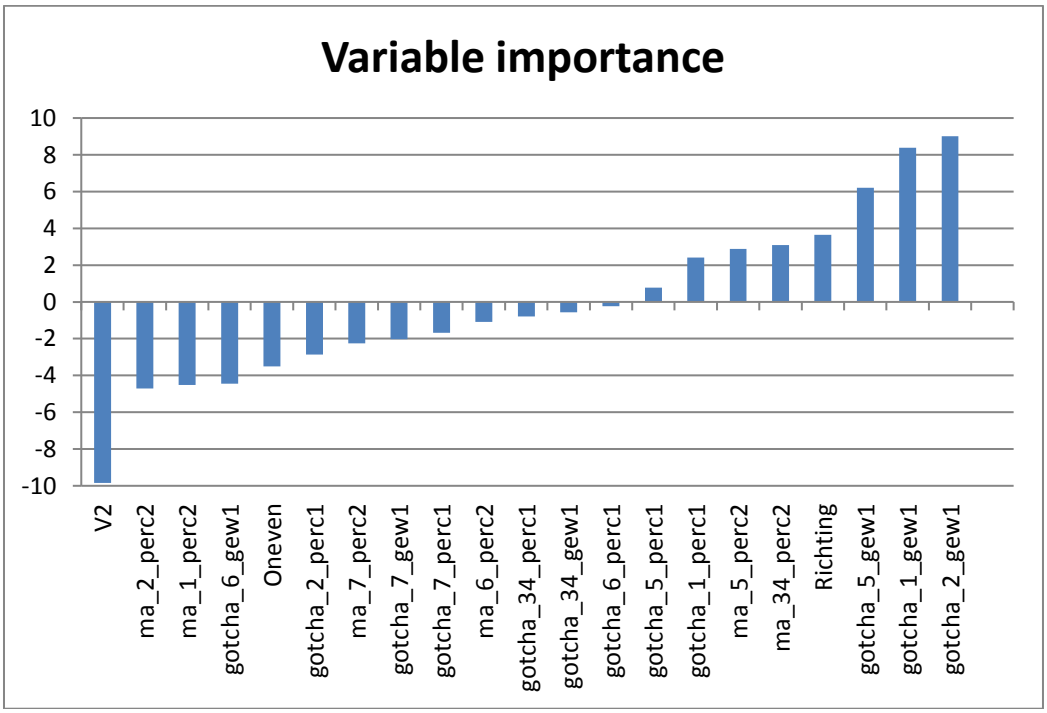
Vervolgens wordt een model met 17 verborgen neuronen getraind op de volledige dataset. Hiervoor zijn alle variabelen geschaald naar waarden tussen 0 en 1 om het trainingsproces sneller te laten verlopen. Bovendien leidt normalisatie van de data vaak tot betere resultaten [28]. Wanneer het model gebruikt wordt bij het voorspellen van de output variabelen, moeten deze weer teruggeschaald worden. Het neuraal netwerk is weergegeven in Figuur 42. Positieve gewichten zijn

met zwarte lijnen aangegeven en negatieve gewichten met grijze lijnen. De breedte van een lijn geeft de grootte van het gewicht aan t.o.v. andere lijnen.




Figuur 42: Neuraal netwerk met 17 verborgen neuronen

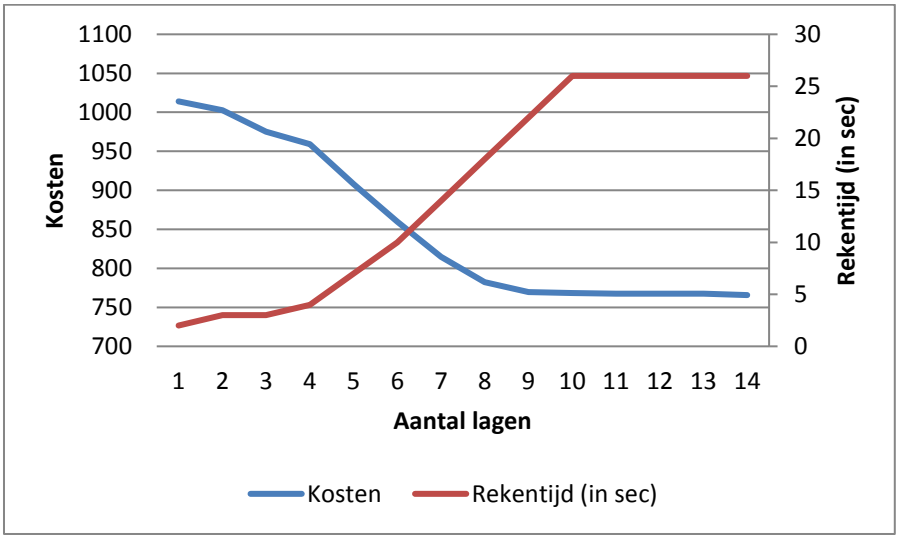
Om het effect van de input variabelen op de output variabelen te onderzoeken is gebruik gemaakt van de Olden functie [29]. Deze functie berekent het effect van een input variabele door het product te nemen van de gewichten die de input laag met de verborgen laag verbinden en de gewichten die de verborgen laag aan de output laag koppelen. Daarna wordt de som genomen van de producten over alle verborgen neuronen. Op deze manier geeft de Olden functie niet alleen aan hoe groot het effect van een variabele is op de output, maar ook of het de output positief of negatief beïnvloedt. In Figuur 43 is de Olden functie bepaald voor alle input variabelen van het model. Uit de grafiek blijkt dat het huidig station en het reizigersgewicht bij aankomst tot de belangrijkste variabelen behoren. Het reizigerspercentage bij aankomst daarentegen is minder significant t.o.v. andere variabelen.




Figuur 43: Significantie attributen in NN met Olden functie

## 6.3 Resultaat Conditional Inference Tree

Zoals eerder genoemd is CIT een boom dat uit meerdere lagen knopen bestaat. Om overfitting te voorkomen mag de boom niet te diep zijn. Daarom is 5-fold cross-validation toegepast op CIT modellen van verschillende dieptes om te onderzoeken wat de optimale aantal lagen knopen is. De kosten hiervan zijn weergegeven in Figuur 44 inclusief de rekentijd. Wanneer er geen restrictie is aan het aantal lagen waaruit een boom kan bestaan, kan de boom hooguit 14 lagen diep worden. Hierbij is de significantie waarde van de permutatie tests vastgesteld op 0,05. Uit de grafiek is te zien dat de kosten stabiel blijft vanaf 9 lagen in de boom. Bovendien neemt de rekentijd vanaf 9 lagen nauwelijks meer toe, omdat het aantal knopen dat wordt gesplitst afneemt. De kosten uit elk fold van de 5-fold cross-validation op de boom met 9 lagen zijn in Tabel 28 weergegeven.

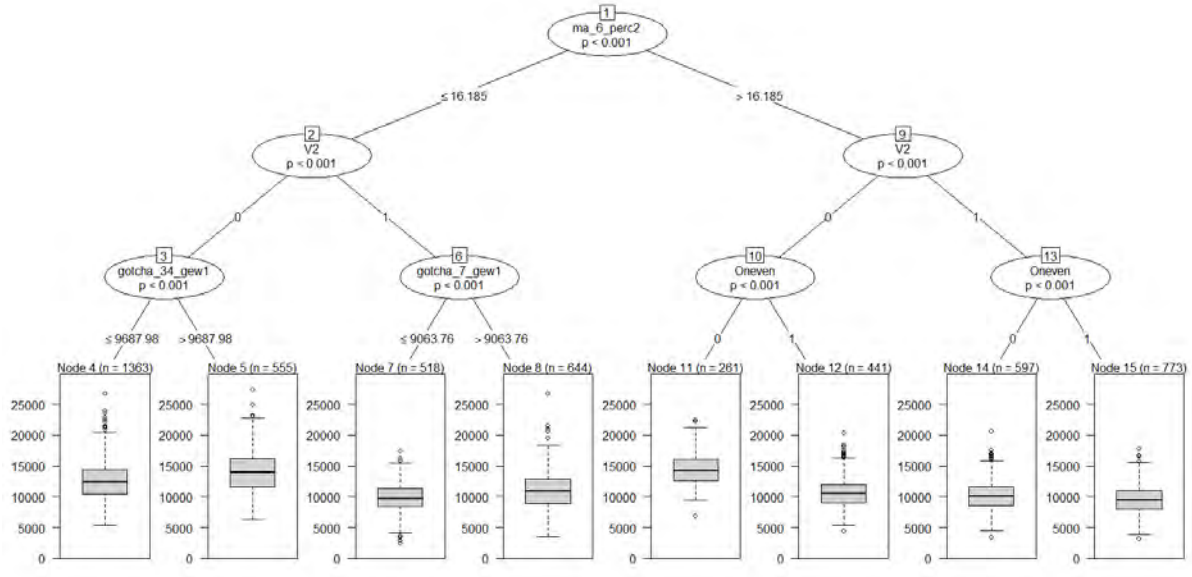



Figuur 44: Kosten en rekentijd van CIT bij verschillend aantal lagen

| fold | 1 | 2 | 3 | 4 | 5 | Gemiddeld |
|---|---|---|---|---|---|---|
| Kosten | 773,83 | 674,93 | 793,67 | 775,82 | 829,78 | 769,61 |

Tabel 28: Kosten van CIT met 5-fold cross-validation

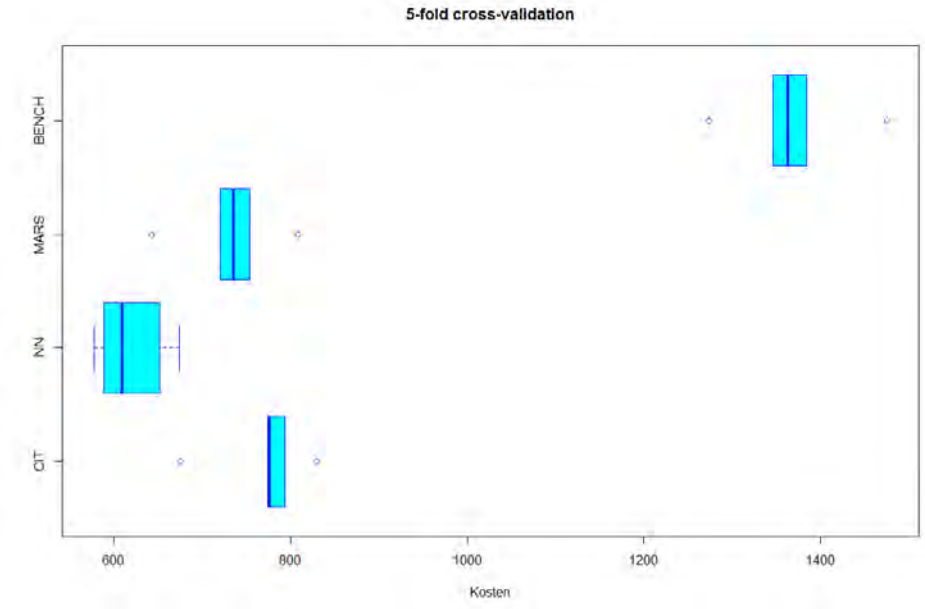
Daarna wordt dit model getraind op de volledige dataset. Een visualisatie van de boom is weergegeven in Figuur 45. De boom is ingekort tot 3 lagen voor beter overzicht. Daarnaast is om dezelfde reden in elke blad de verdeling van het reizigersgewicht in de mBvk1 weergeven. De variabelen in deze lagen zijn één van de belangrijkste variabelen in het model. In de bovenste laag wordt de boom gesplitst aan de hand van de variabele ‘ma_6_perc2’. Wanneer het gemiddeld reizigerspercentage in het verleden in de ABv6 bij vertrek groter is dan 16,185% wordt het rechter pad genomen, anders volgt men het linker pad. Vervolgens wordt de boom verder gesplitst door het huidig station en in de derde laag door het reizigersgewicht bij aankomst en de richting van het traject. Afhankelijk van het pad dat wordt genomen in de boom, varieert de verdeling van het reizigersgewicht in de bakken. We zien bijvoorbeeld dat wanneer ‘ma_6_perc2’ groter is dan 16,185%, het huidig station ‘s-Hertogenbosch is en de trein richting Utrecht rijdt (Node 11), de verdeling van het reizigersgewicht in mBvk1 groter is dan wanneer de trein richting Weert rijdt (Node 12).




Figuur 45: CIT-boom met drie lagen

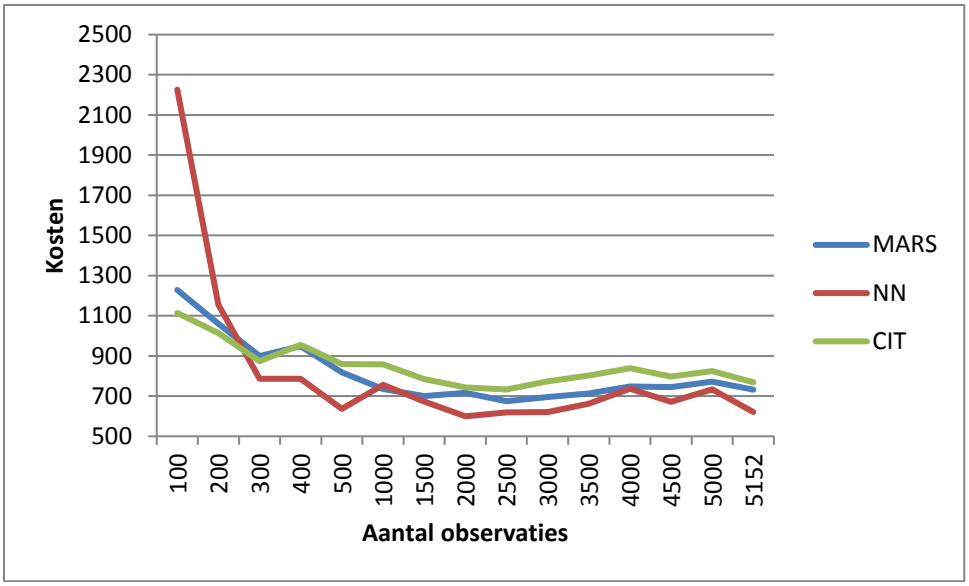
## 6.4 Vergelijking modellen

In Figuur 46 zijn de kosten van de drie modellen en het benchmark met 5-fold cross-validation onder elkaar gezet. Als eerste valt meteen op dat de drie modellen significant beter presteren dan het benchmark. Met de Wilcoxon signed rank test kan aangetoond worden dat de kosten van NN significant lager is dan de kosten van MARS en CIT. Bij het vergelijken van MARS met CIT resulteerde de Wilcoxon signed rank test in een p-waarde van 0,03, waardoor we de hypothese dat er geen verschil is tussen de twee verdelingen kunnen verwerpen. CIT heeft dus gemiddeld de hoogste kosten van de drie machine learning algoritmes.



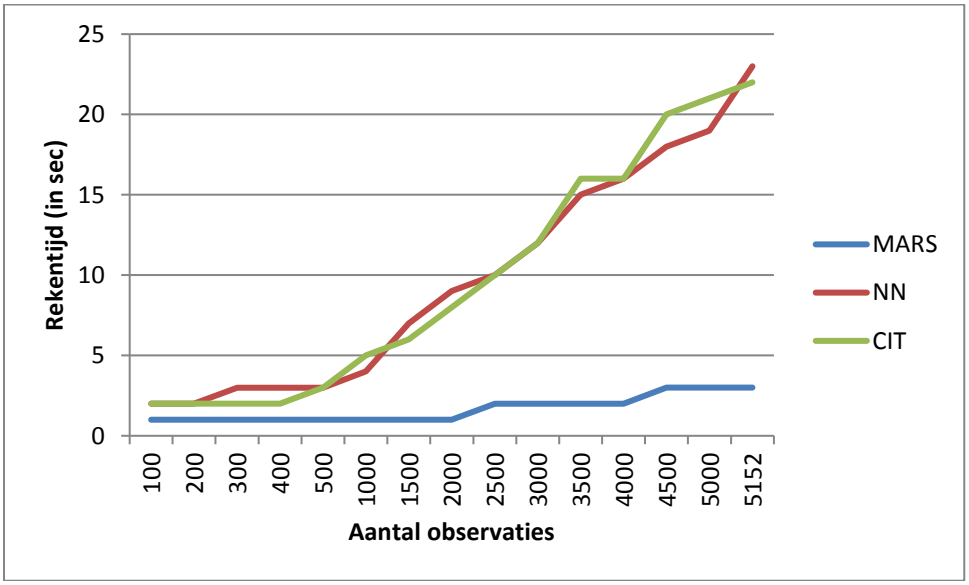

Figuur 46: Vergelijking kosten van MARS, NN,CIT en het benchmark met 5-fold cross-validation

Verder is de nauwkeurigheid van de modellen onderzocht met datasets van verschillende grootte. Hierbij wordt opnieuw gebruik gemaakt van 5-fold cross-validation.




Figuur 47: Kosten van MARS, NN en CIT bij verschillend aantal observaties

In Figuur 47 zien we dat voor alle drie modellen de laagste kosten zijn bereikt rond 2000 en 2500 observaties in de dataset. Dat is ongeveer de helft van onze oorspronkelijke dataset. Daarnaast is ook voor verschillend aantal observaties in de dataset de rekentijd van de modellen bepaald in Figuur 48.

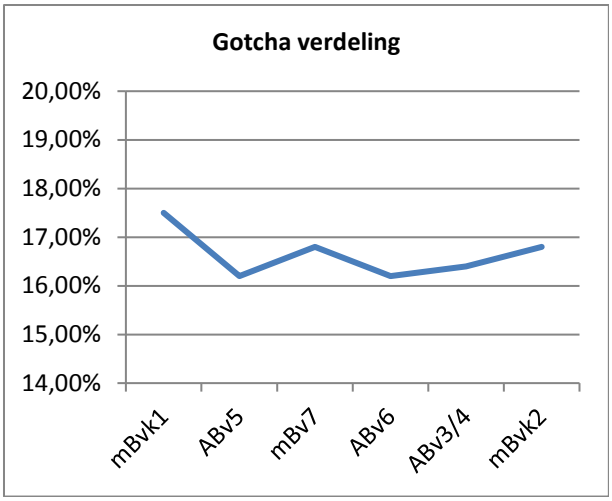



Figuur 48: Rekentijd van MARS, NN en CIT bij verschillend aantal observaties

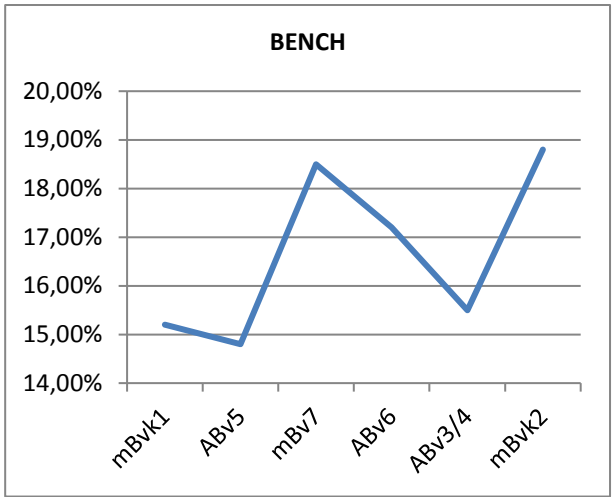
De rekentijd van NN en CIT neemt ongeveer op dezelfde wijze toe bij het verhogen van het aantal observaties in de dataset. De rekentijd van MARS daarentegen neemt nauwelijks toe in vergelijking met NN en CIT.

## 6.5 Evaluatie

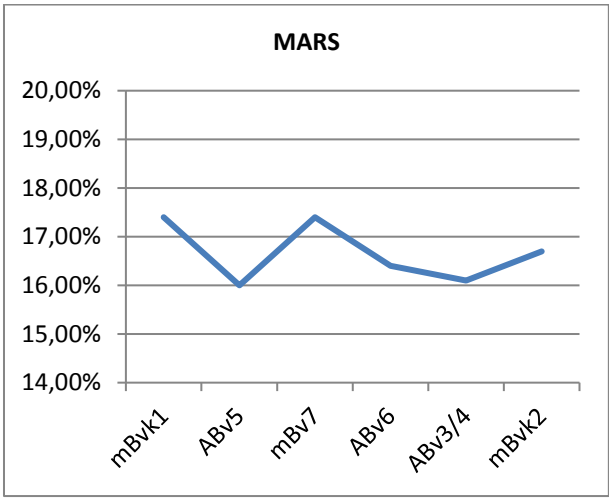
Om een beter beeld te krijgen van de nauwkeurigheid van de modellen worden eerst de drie modellen getraind met 2000 observaties en getest op 500 observaties, zodat een paar verdelingen uit de test set vergeleken kan worden met de voorspelde verdelingen van de modellen. Drie verdelingen uit de test set met verschillende kenmerken zijn hiervoor geselecteerd.



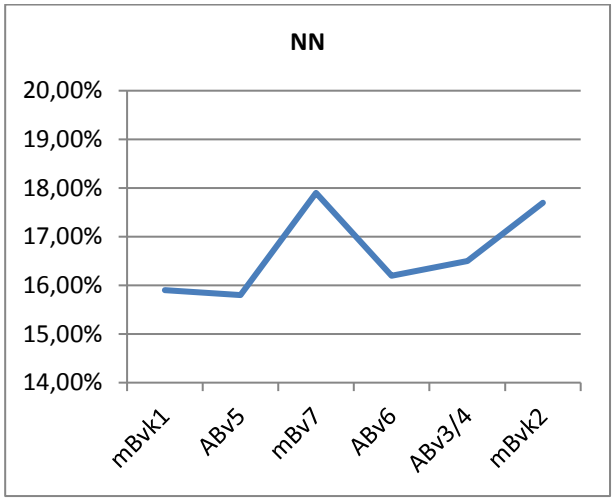

Figuur 49: Reizigersverdeling Gotcha (1)



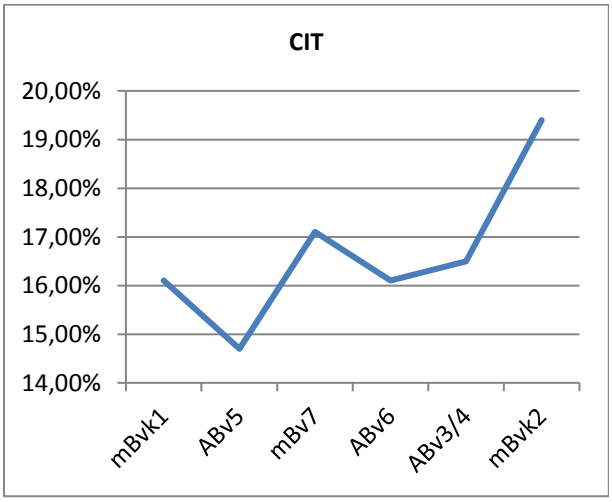

Figuur 50: Reizigersverdeling benchmark (1)




Figuur 51: Reizigersverdeling MARS (1)




Figuur 52: Reizigersverdeling NN (1)

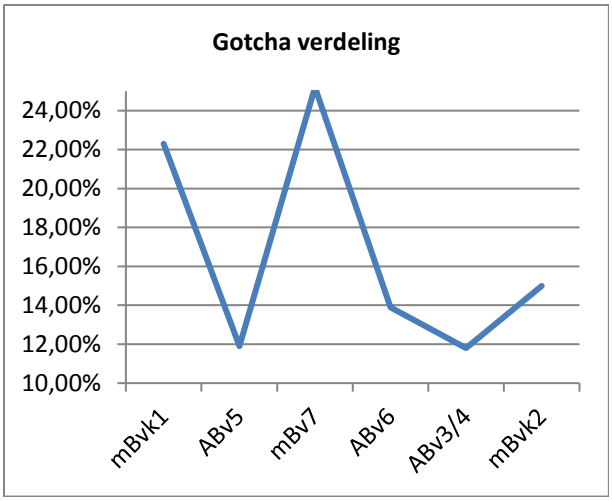



Figuur 53: Reizigersverdeling CIT (1)

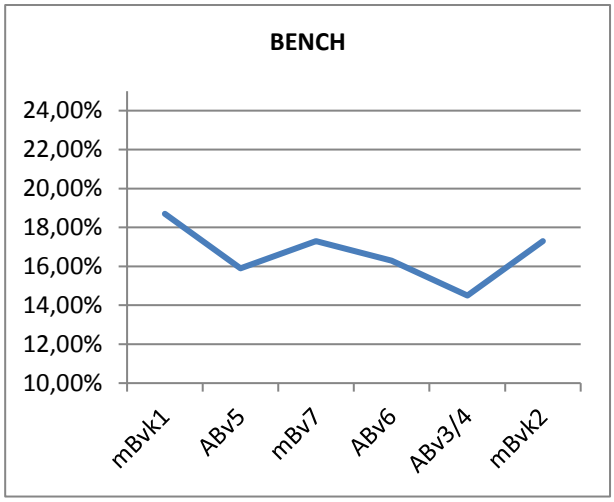
| | Kosten |
|---|---|
| BENCH | 207,80 |
| MARS | 5,05 |
| NN | 40,50 |
| CIT | 93,09 |

Tabel 29: Kosten van voorspelling (1)

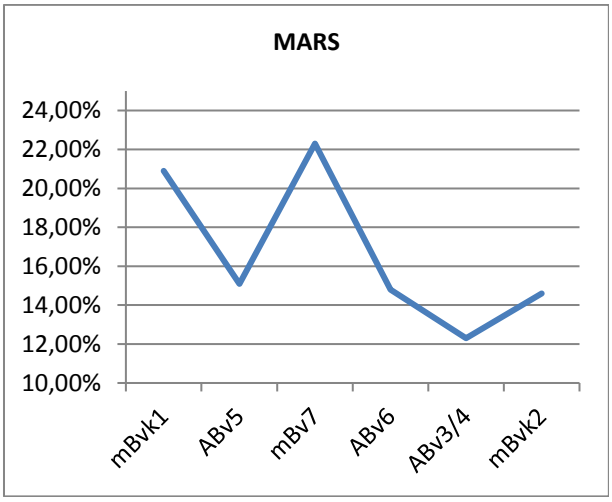
In de eerste observatie is de variatie in reizigerspercentage tussen de bakken uit Gotcha het kleinst. Bovenstaande figuren geven de verschillende voorspellingen hiervan weer. Daarnaast zijn ook de kosten van de drie voorspellingen bepaald in Tabel 29. Slechts de voorspellingen van MARS en NN komen het meest overeen met de Gotcha verdeling.



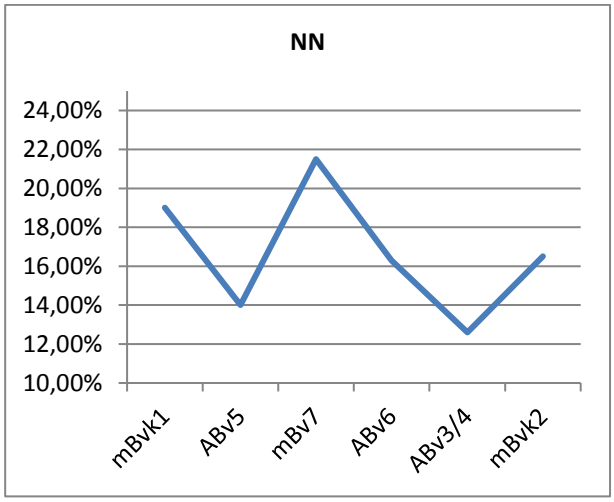

Figuur 54: Reizigersverdeling Gotcha (2)



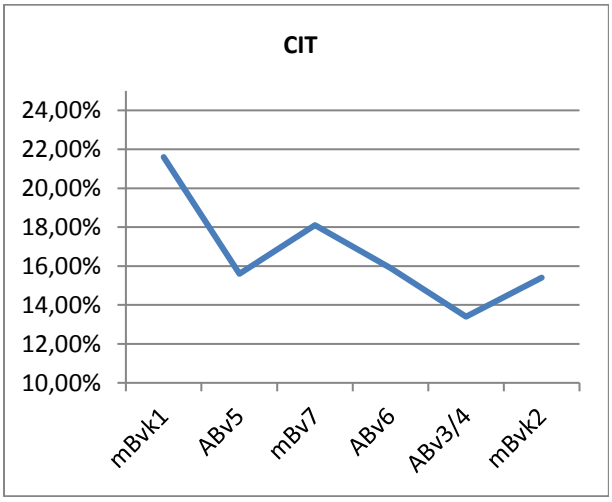

Figuur 55: Reizigersverdeling benchmark (2)




Figuur 56: Reizigersverdeling MARS (2)




Figuur 57: Reizigersverdeling NN (2)

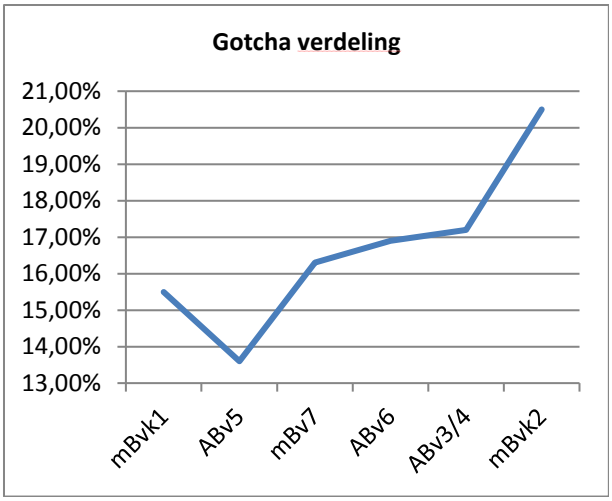



Figuur 58: Reizigersverdeling CIT (2)

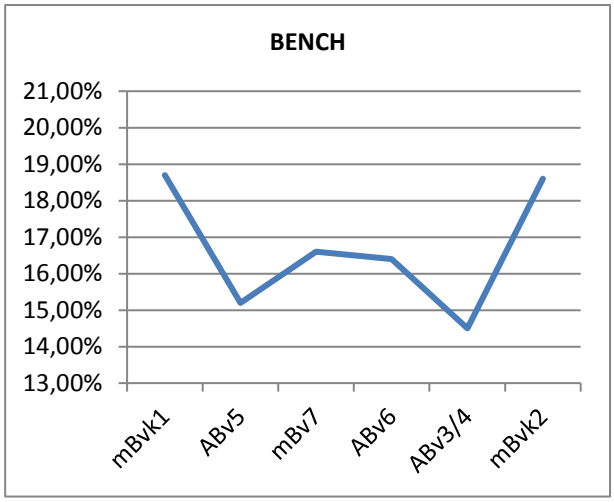
| | Kosten |
|---|---|
| BENCH | 15052,35 |
| MARS | 2406,50 |
| NN | 5345,60 |
| CIT | 6822,60 |

Tabel 30: Kosten van voorspelling (2)

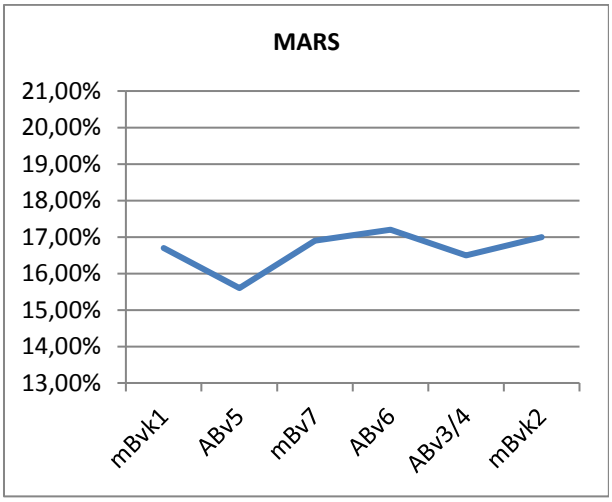
In de volgende observatie is de variatie in reizigerspercentage tussen de bakken het grootst bij Gotcha. Ook hier komen de verdelingen van NN en MARS het meest overeen met de Gotcha verdeling. Uit de kosten in Tabel 30 is dit ook terug te zien.



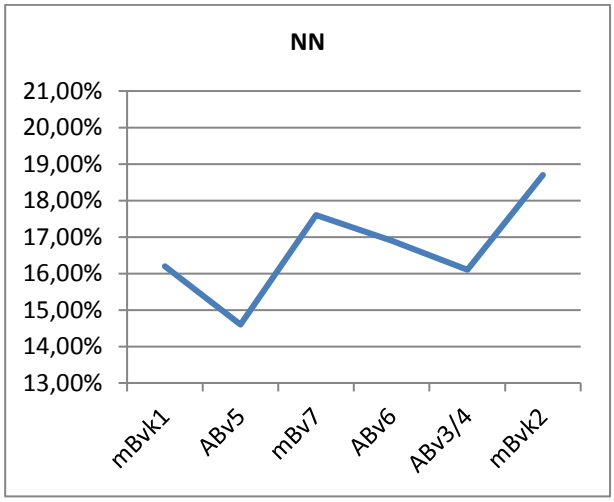

Figuur 59: Reizigersverdeling Gotcha (3)



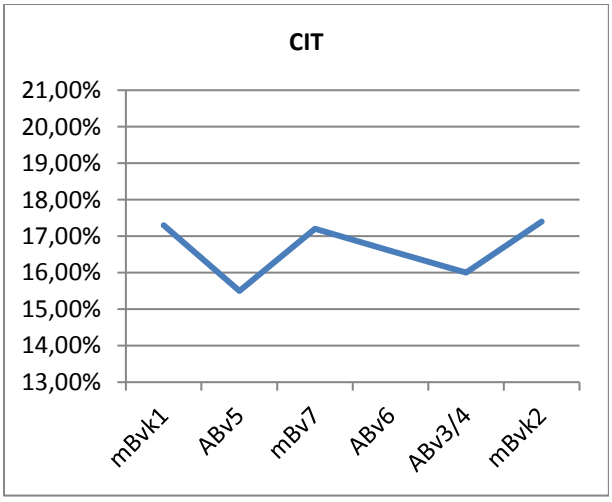

Figuur 60: Reizigersverdeling benchmark (3)




Figuur 61: Reizigersverdeling MARS (3)




Figuur 62: Reizigersverdeling NN (3)



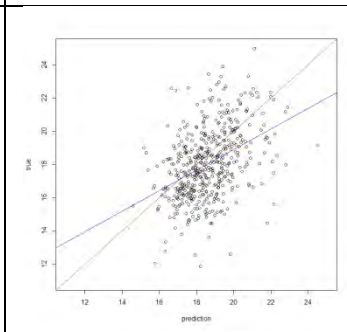
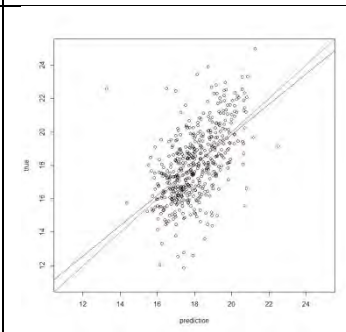
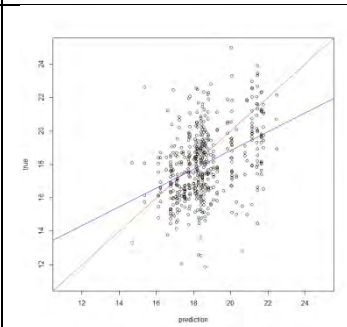
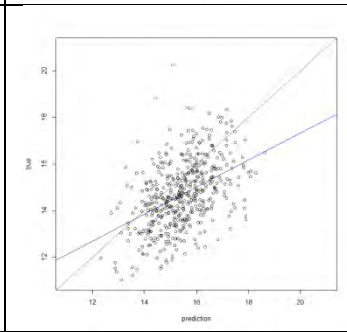
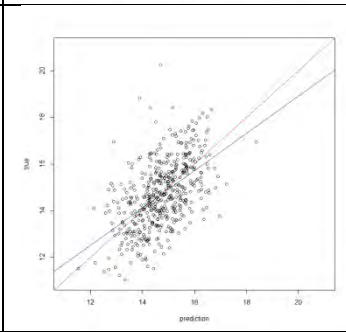
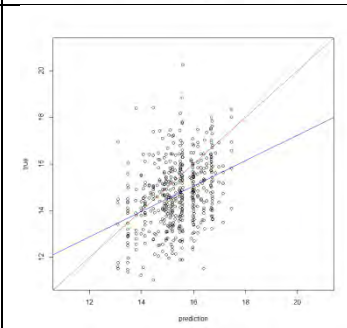
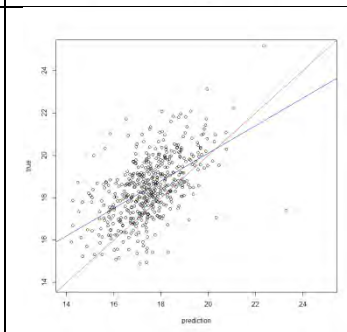
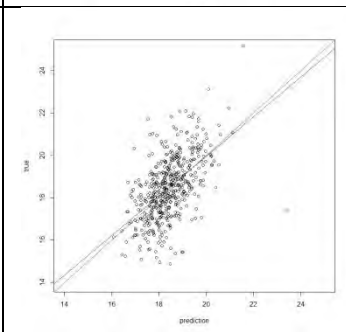
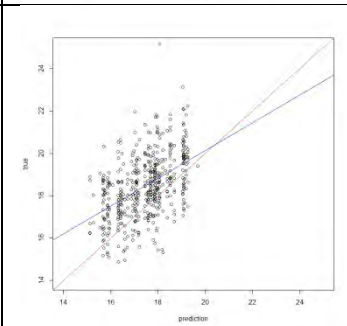
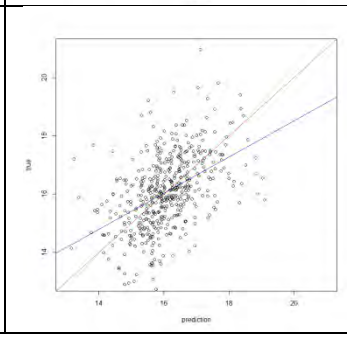
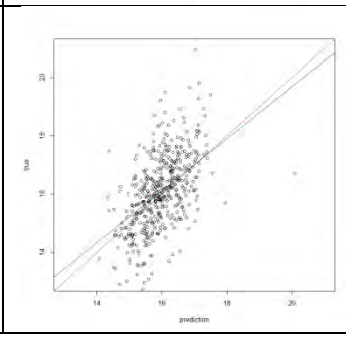
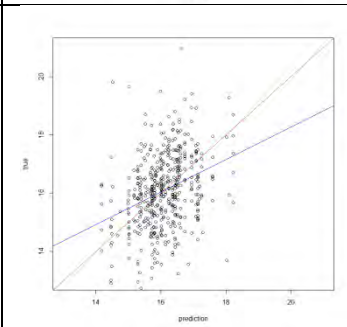

Figuur 63: Reizigersverdeling CIT (3)

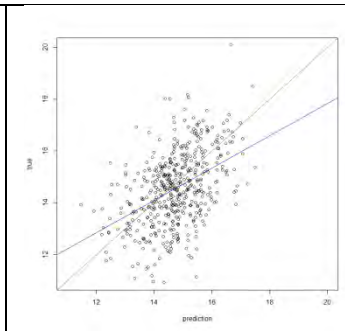
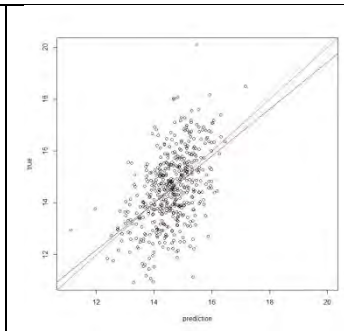
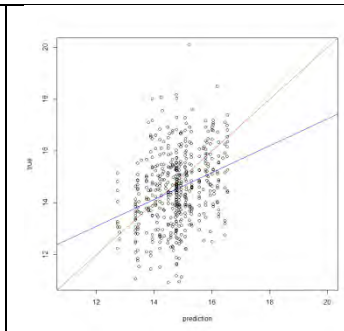
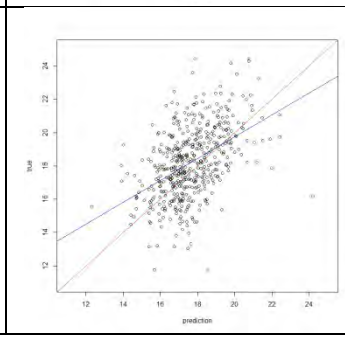
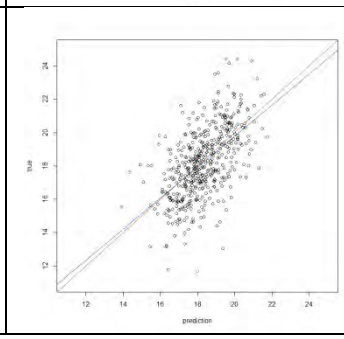
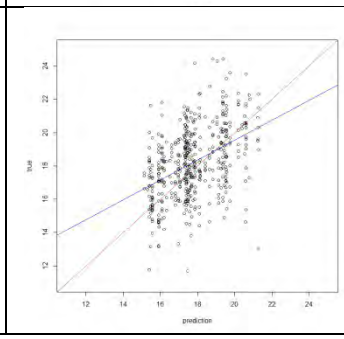
| | Kosten |
|---|---|
| BENCH | 942,49 |
| MARS | 637,44 |
| NN | 311,80 |
| CIT | 775,34 |

Tabel 31: Kosten van voorspelling (3)

De laatste observatie is een voorbeeld van een scheve reizigersverdeling, waarbij het in de mBvk2 overvol zit, terwijl er nog plaatsen beschikbaar zijn in de mBvk1. Het verschil tussen deze twee bakken is alleen terug te zien in de verdeling van NN. In Tabel 31 geeft NN ook de laagste kosten.

Verder is in Tabel 32 per bak en model het reizigerspercentage uit Gotcha geplot tegenover het voorspelde reizigerspercentage van de observaties in de test set. Correct voorspelde observaties liggen op de rode lijn en de blauwe lijn is de regressie lijn op basis van de kleinste-kwadratenmethode. Beide lijnen liggen het dichts op elkaar bij de scatterplots van NN.

| | MARS | NN | CIT |
|---|---|---|---|
| mBvk1 |  |  |  |
| ABv5 |  |  |  |
| mBv7 |  |  |  |
| ABv6 |  |  |  |

| | | | |
|---|---|---|---|
| ABv3/4 |  |  |  |
| mBvk2 |  |  |  |

Tabel 32: Scatterplot voorspelde met daadwerkelijke reizigerspercentages per bak en model

In Tabel 33 is ook per bak en model de correlatie bepaald tussen Gotcha en de voorspelde reizigerspercentages en is de Wilcoxon signed rank test toegepast, waarvan de p-waardes zijn gegeven. Gemiddeld is de correlatie hoger bij NN en de Wilcoxon signed rank test geeft aan met een significantie waarde van 0,05 dat het gemiddelde van de Gotcha reizigerspercentages voor alle bakken niet afwijkt van het gemiddelde van de voorspelde reizigerspercentages behalve bij ABv6. Bij MARS en CIT is er wel een significant verschil tussen de twee gemiddeldes op de ABv6 na. Wanneer het gemiddelde van de daadwerkelijke reizigerspercentages groter is dan het gemiddelde van de voorspelde reizigerspercentages wordt dit met een '-' aangeduid. Anders is een '+' weergegeven. Aan het aantal observaties boven en onder de rode lijnen in Tabel 32 kunnen we ook zien of het gemiddelde van de daadwerkelijke reizigerspercentages kleiner of groter is dan het gemiddelde van de voorspelde reizigerspercentages per bak en model.

| | MARS | | NN | | CIT | |
|---|---|---|---|---|---|---|
| | Correlatie | p-waarde | Correlatie | p-waarde | Correlatie | p-waarde |
| mBvk1 | 0,42 | < 0,01 (+) | 0,55 | 0,77 | 0,40 | < 0,01 (+) |
| ABv5 | 0,45 | < 0,01 (+) | 0,54 | 0,20 | 0,36 | < 0,01 (+) |
| mBv7 | 0,55 | < 0,01 (-) | 0,55 | 0,29 | 0,48 | < 0,01 (-) |
| ABv6 | 0,47 | 0,81 | 0,52 | 0,01 (-) | 0,32 | 0,89 |
| ABv3/4 | 0,46 | < 0,01 (+) | 0,50 | 0,41 | 0,31 | < 0,01 (+) |
| mBvk2 | 0,42 | < 0,01 (-) | 0,57 | 0,17 | 0,43 | < 0,01 (-) |

Tabel 33: Correlatie en Wilcoxon signed-rank test tussen voorspelde en daadwerkelijke reizigerspercentages per bak en model

# 7 Conclusie

In het laatste hoofdstuk van dit verslag worden de voornaamste conclusies besproken. Daarna worden ook enkele aanbevelingen gegeven en mogelijk verder onderzoek toegelicht.

## 7.1 Selectie databron

Ten eerste is gekeken welk databron geschikt is voor het bepalen van de reizigersverdeling in de trein. De WiFi data is als eerste afgevallen, omdat de data alleen op treinstel niveau beschikbaar is. Hoewel er in potentie meer mee mogelijk is. Vervolgens worden twee databronnen die betrekking hebben op het gewicht van de trein uitgebreid vergeleken. Uiteindelijk is voor Gotcha gekozen, omdat hiermee het gewicht van alle treinbakken kan worden bepaald en de metingen van de verschillende bakken hebben onderling een positieve correlatie, wat van belang is om de juiste verhouding van het reizigersgewicht over de bakken te kunnen bepalen. Het enig nadeel van Gotcha is dat er veel variatie is in de metingen, waardoor het minder geschikt is voor het bepalen van het aantal reizigers in een treinbak. Ons doel is echter het voorspellen van de reizigersverdeling in de trein, dus het volstaat als Gotcha het reizigerspercentage in elke treinbak kan bepalen.

## 7.2 Selectie attributen

Met de attributen selectie methoden zijn de volgende attributen verkregen:

- *Huidig station*
- *Richting traject*
- *Richting trein*
- *Het gewicht en percentage per bak voor aankomst*
- *Het gemiddeld percentage reizigers per bak bij vertrek in het verleden*

Wanneer de drie modellen (MARS, NN en CIT) werden toegepast met de bovenstaande attributen, bleek dat voornamelijk het huidig station, richting van het traject en het reizigersgewicht per bak voor aankomst de belangrijkste attributen zijn voor de modellen. Het minst significante attribuut is het reizigerspercentage per bak voor aankomst. Een verklaring hiervoor is dat het reizigersgewicht per bak voor aankomst reeds als attribuut aanwezig is in de modellen. Het reizigerspercentage is eigenlijk slechts een afleiding van het reizigersgewicht per bak. Uit het reizigersgewicht per bak is niet alleen de reizigersverdeling af te leiden, maar ook de drukte in de trein.

Attributen die niet geselecteerd zijn door de attributen selectie methoden, maar waarvan we hadden verwacht dat deze belangrijk zouden zijn voor het bepalen van de reizigersverdeling zijn het tijdstip van de meting en de dag van de week. Het feit dat deze attributen niet zijn geselecteerd, zou erop kunnen duiden dat deze slechts een indicatie zijn voor de drukte in de trein dat reeds afgeleid kan worden uit het reizigersgewicht per bak.

## 7.3 Selectie model

Met 5-fold cross-validation zijn MARS, NN en CIT met elkaar vergeleken. Aan de hand van een Wilcoxon signed rank test is aangetoond dat de kosten van de drie modellen significant verschillen. Van de drie modellen heeft NN de laagste kosten behaald en presteerde CIT het slechtst. Desondanks deden de drie modellen het allemaal beter dan het benchmark. Verder is de rekentijd van MARS significant lager dan de rekentijd van NN en CIT. Daarna zijn de modellen getraind met 2000

observaties, omdat de laagste kosten zijn behaald bij het toepassen van 5-fold cross-validation op een dataset van 2500 observaties. Uit de testset van 500 observaties zijn drie Gotcha verdelingen vergeleken met de voorspellingen van de drie modellen om een inzicht te krijgen hoe nauwkeurig deze modellen de Gotcha verdeling kan weergeven. Twee van de drie observaties (één met de minste variatie in reizigerspercentage tussen de bakken en één met de meeste variatie in reizigerspercentage) komen het best overeen met de voorspellingen van MARS en NN. Bij de derde observatie (scheve verdeling) is NN het best in staat om de scheve reizigersverdeling in de trein weer te geven. CIT presteert het slechtst op deze drie observaties. Ten slotte is per bak en model de correlatie bepaalt tussen de voorspelde reizigerspercentages en de reizigerspercentages uit Gotcha. Bij NN is de correlatie het grootst en het kleinst bij CIT. Met Wilcoxon signed rank test is aangetoond dat bij MARS en CIT het gemiddelde van de metingen en de voorspellingen van elkaar afwijken bij alle bakken behalve de ABv6, terwijl dit bij NN andersom is.
Uit de bovenstaande resultaten blijkt dat MARS en NN het beste hebben gepresteerd.

## 7.4 Beperkingen

Er zijn enkele aannames gemaakt om tot een voorspellingsmodel uit te komen. Het eerste probleem waarop we tegen zijn gelopen is dat de tijden uit de beladingssensor data niet overeenkomen met de tijden uit de dataset 'Treinactiviteiten'. Met behulp van de treinsnelheid worden deze twee databronnen aan elkaar gekoppeld. Echter gaan er veel metingen verloren om zeker van te zijn dat de metingen aan het juiste traject worden gekoppeld. Om te voorkomen dat er geen metingen aan een traject wordt gekoppeld, moet er ten minste 10 minuten reistijd zijn tussen twee stations. Daarnaast is beladingssensor data niet voor alle VIRM treinstellen beschikbaar, waardoor een specifiek treinsamenstelling (één VIRM-VI treinstel) moet worden geselecteerd die het vaakst voorkwam.

Vervolgens is aangenomen dat het gewicht van de reizigers in een treinbak bepaald kan worden door het gewicht van een lege treinbak af te trekken van bijvoorbeeld de Gotcha-meting. Hierbij is geen rekening gehouden met het gewicht van bijvoorbeeld reizigersbagage. Daarnaast is het reizigersgewicht in een treinbak niet direct te vertalen naar het aantal reizigers, omdat niet iedereen hetzelfde weegt. Wanneer bijvoorbeeld een grote groep kinderen in een treinbak zitten, geeft het laag reizigersgewicht in de bak een verkeerd beeld over de drukte in de trein. In paragraaf 4.3.3 is wel aangetoond dat het reizigersgewicht correleert met het voorspeld aantal reizigers uit SOFA.

Verder is het probleem met Gotcha dat er veel variatie is in de metingen, waardoor vaak negatieve reizigersgewichten zijn verkregen. Om deze negatieve waarden om te zetten naar percentages is een lager gewicht van de Gotcha-metingen afgetrokken. Dit heeft natuurlijk invloed op de reizigersverdeling, maar het verschil tussen het aftrekken van het theoretisch gewicht en een lager gewicht is acceptabel wanneer beide aanpakken zijn vergeleken met de daadwerkelijke reizigersverdeling van de tellingen in paragraaf 4.3.4.

Als laatst zijn de modellen minder in staat om verdelingen met grote verschillen in reizigerspercentages tussen bakken te voorspellen. In Tabel 32 kunnen we namelijk zien dat er minder variatie is in de voorspellingen van het reizigerspercentage dan de reizigerspercentages uit Gotcha. De modellen kunnen bijvoorbeeld wel scheve verdelingen identificeren, maar de voorspelde reizigersverdeling is vaak minder scheef dan de verdeling uit Gotcha.

## 7.5 Aanbevelingen

Voor het voorspellen van de reizigersverdeling bij vertrek gaat ons voorkeur naar een NN-model. De rekentijd van NN is weliswaar langer dan MARS, maar is snel genoeg om binnen een minuut een model te trainen. Bovendien is met 5-fold cross-validation aangetoond dat het voorspellingsmodel van NN significant beter presteert dan MARS. Het belangrijkst is dat NN in staat is om de globale reizigersverdeling in de trein te voorspellen. Om het voorspellingsmodel op basis van Gotcha gangbaar te maken is het van belang dat er meer Gotcha meetpunten worden aangelegd zodat de reizigersverdeling op meer stations voorspeld kan worden.

## 7.6 Verder onderzoek

Als verder onderzoek kan de daadwerkelijke reizigersverdeling uit reizigerstellingen vergeleken worden met de reizigersverdeling bepaald met beladingssensoren. In dit onderzoek zijn de tellingen alleen vergeleken met Gotcha, omdat de data van de beladingssensoren niet beschikbaar was tijdens het tellen in de trein. Wanneer beide databronnen worden vergeleken met de reizigerstellingen, zou dit een beter beeld kunnen geven van het verschil tussen Gotcha en beladingssensoren bij het bepalen van de reizigersverdeling.

In dit onderzoek wordt de reizigersverdeling alleen voorspeld op ‘s-Hertogenbosch en Eindhoven. Bij het trainen van de modellen is geen onderscheid gemaakt tussen deze stations. Een idee is om te onderzoeken of een apart model voor elk station betere voorspellingen levert dan een globaal model voor alle stations. Dit kan onderzocht worden wanneer meer Gotcha meetpunten beschikbaar zijn. Daarnaast is het interessant om te onderzoeken wat voor effect het wel en niet tonen van informatie over de reizigersverdeling in de trein heeft op de manier waarop reizigers instappen. Er kan bijvoorbeeld onderzocht worden welke informatie aan reizigers op het perron getoond moet worden, zodat bij vertrek reizigers gelijkmatig over de trein zijn verdeeld.

De voorspelde reizigersverdeling kan gecombineerd worden met het voorspeld aantal reizigers in de hele trein (nog niet beschikbaar) om het aantal reizigers per bak bij vertrek te bepalen. Hierna is het handig om te weten wat het percentage $1^e$ en $2^e$ klas reizigers is per bak, zodat het aantal lege $1^e$ en $2^e$ klas zitplaatsen weergegeven kan worden.

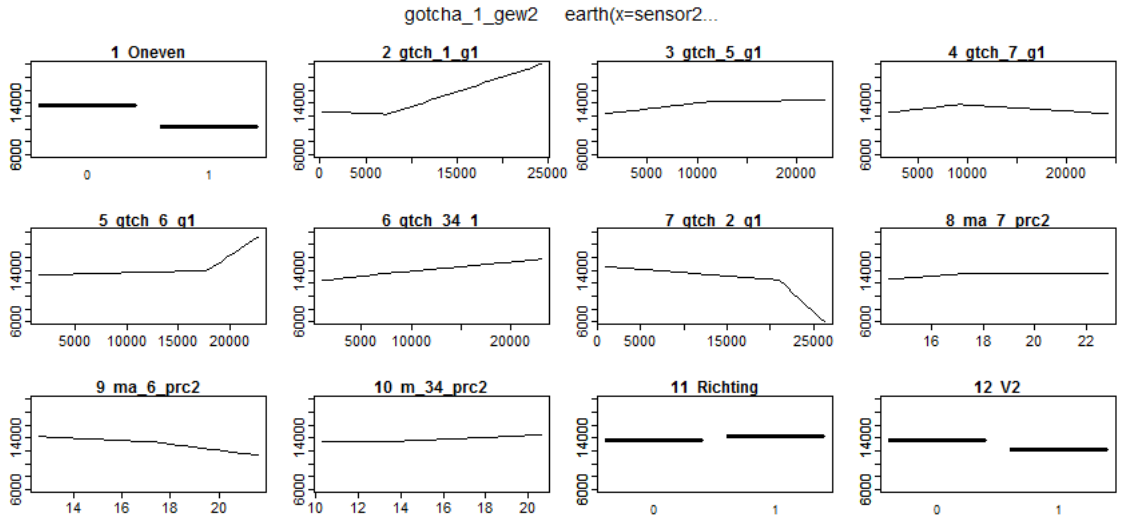
Tenslotte kan onderzocht worden of andere machine learning technieken betere resultaten opleveren. In dit onderzoek is het selecteren van attributen gebaseerd op backward selection en de kost functie voor het evaleren van een model. Wellicht leidt een ander attribuut selectie methode en kost functie tot een nieuwe verzameling van significante attributen.

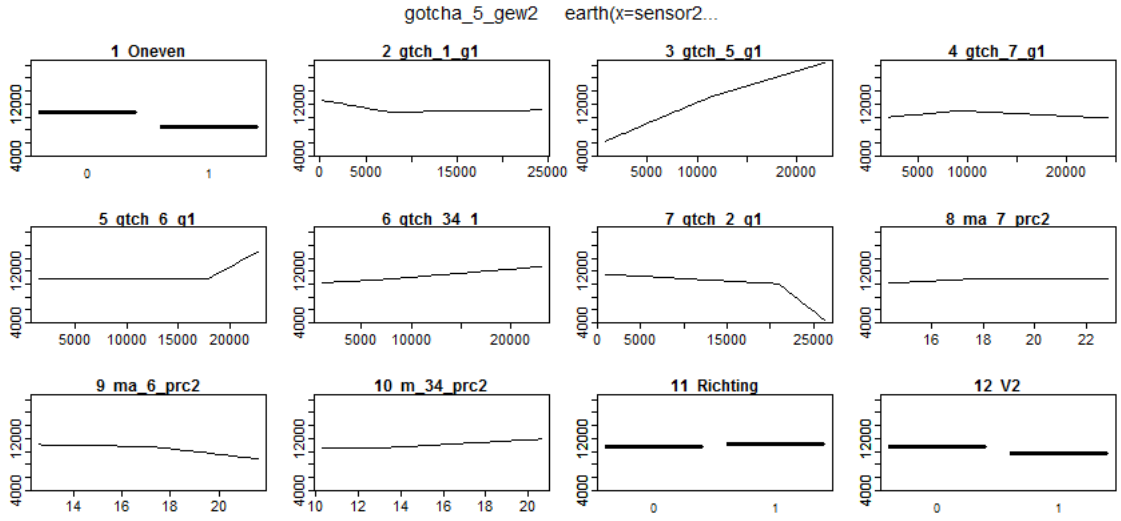
# Appendix

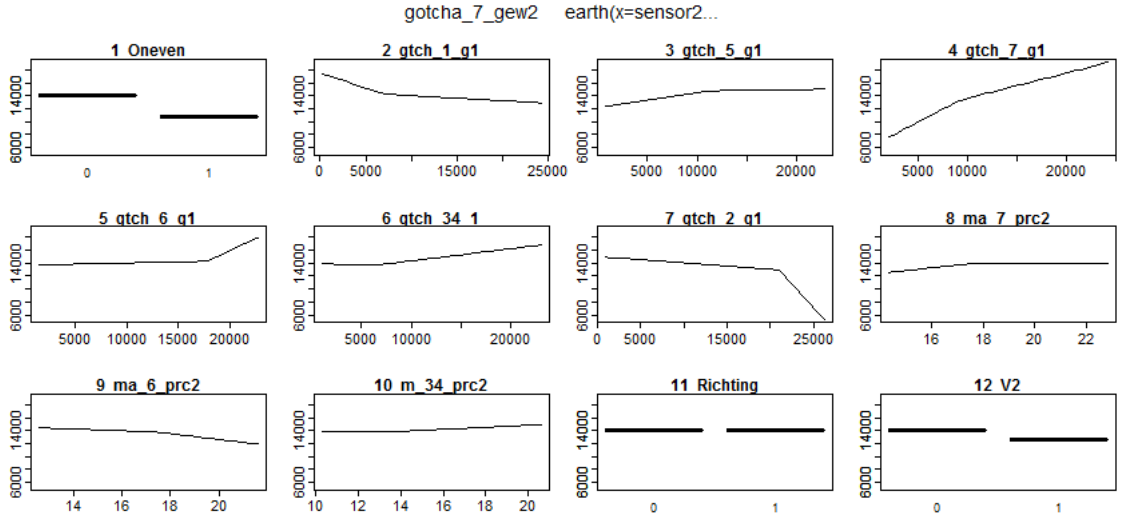
## A MARS regressie formule per bak

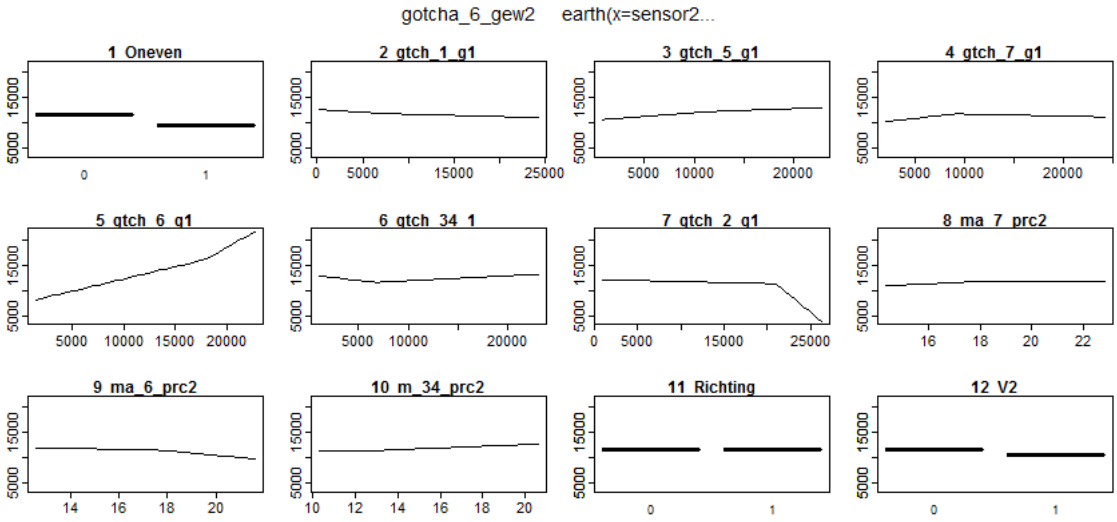
```
gotcha_1_gew2 =
  1.2e+04
  -    3311 * Oneven1
  +     551 * Richting1
  -    1517 * V21
  +   0.063 * max(0,            7400 -  gotcha_1_gew1)
  +    0.46 * max(0,  gotcha_1_gew1 -            7400)
  -    0.18 * max(0,           11188 -  gotcha_5_gew1)
  +   0.022 * max(0,  gotcha_5_gew1 -           11188)
  -    0.17 * max(0,            9284 -  gotcha_7_gew1)
  -   0.094 * max(0,  gotcha_7_gew1 -            9284)
  -   0.042 * max(0,           17860 -  gotcha_6_gew1)
  +     1.1 * max(0,  gotcha_6_gew1 -           17860)
  -    0.18 * max(0,            6858 - gotcha_34_gew1)
  +    0.14 * max(0, gotcha_34_gew1 -            6858)
  +     0.1 * max(0,           21033 -  gotcha_2_gew1)
  -     1.2 * max(0,  gotcha_2_gew1 -           21033)
  -     290 * max(0,              18 -     ma_7_perc2)
  +     178 * max(0,              17 -     ma_6_perc2)
  -     477 * max(0,     ma_6_perc2 -             17)
  +     142 * max(0,    ma_34_perc2 -             13)

gotcha_5_gew2 =
  1.2e+04
  -    2346 * Oneven1
  +     395 * Richting1
  -    1173 * V21
  +    0.26 * max(0,            7400 -  gotcha_1_gew1)
  +   0.017 * max(0,  gotcha_1_gew1 -            7400)
  -    0.65 * max(0,           11188 -  gotcha_5_gew1)
  +    0.46 * max(0,  gotcha_5_gew1 -           11188)
  -    0.13 * max(0,            9284 -  gotcha_7_gew1)
  -   0.077 * max(0,  gotcha_7_gew1 -            9284)
  +  0.0041 * max(0,           17860 -  gotcha_6_gew1)
  +    0.87 * max(0,  gotcha_6_gew1 -           17860)
  -    0.08 * max(0,            6858 - gotcha_34_gew1)
  +    0.13 * max(0, gotcha_34_gew1 -            6858)
  +   0.074 * max(0,           21033 -  gotcha_2_gew1)
  -     1.1 * max(0,  gotcha_2_gew1 -           21033)
  -     193 * max(0,              18 -     ma_7_perc2)
  +      68 * max(0,              17 -     ma_6_perc2)
  -     435 * max(0,     ma_6_perc2 -             17)
  +     181 * max(0,    ma_34_perc2 -             13)

gotcha_7_gew2 =
  1.3e+04
  -    3302 * Oneven1
  +      72 * Richting1
  -    1318 * V21
  +    0.43 * max(0,            7400 -  gotcha_1_gew1)
  -   0.078 * max(0,  gotcha_1_gew1 -            7400)
  -    0.23 * max(0,           11188 -  gotcha_5_gew1)
  +   0.022 * max(0,  gotcha_5_gew1 -           11188)
  -     0.8 * max(0,            9284 -  gotcha_7_gew1)
  +    0.39 * max(0,  gotcha_7_gew1 -            9284)
  -   0.034 * max(0,           17860 -  gotcha_6_gew1)
  +    0.76 * max(0,  gotcha_6_gew1 -           17860)
  +   0.016 * max(0,            6858 - gotcha_34_gew1)
  +    0.19 * max(0, gotcha_34_gew1 -            6858)
  +   0.098 * max(0,           21033 -  gotcha_2_gew1)
  -     1.5 * max(0,  gotcha_2_gew1 -           21033)
  -     419 * max(0,              18 -     ma_7_perc2)
  +     129 * max(0,              17 -     ma_6_perc2)
  -     438 * max(0,     ma_6_perc2 -             17)
  +     156 * max(0,    ma_34_perc2 -             13)
```

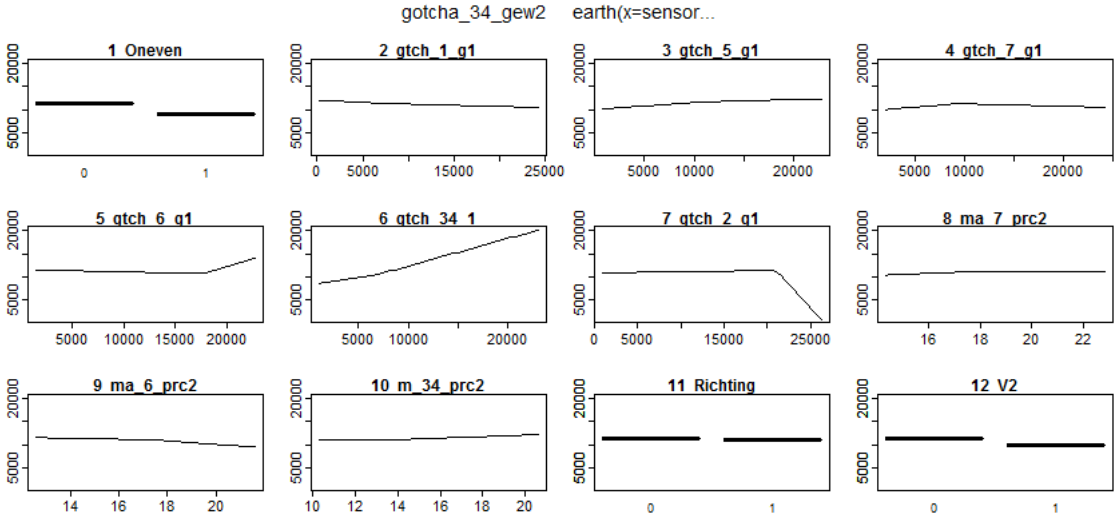
```
gotcha_6_gew2 =
  1.6e+04
  -    2078 * Oneven1
  -     114 * Richting1
  -    1143 * V21
  +     0.1 * max(0,            7400 -  gotcha_1_gew1)
  -   0.047 * max(0,  gotcha_1_gew1 -            7400)
  -    0.15 * max(0,           11188 -  gotcha_5_gew1)
  +   0.069 * max(0,  gotcha_5_gew1 -           11188)
  -    0.21 * max(0,            9284 -  gotcha_7_gew1)
  -   0.041 * max(0,  gotcha_7_gew1 -            9284)
  -    0.48 * max(0,           17860 -  gotcha_6_gew1)
  +     1.1 * max(0,  gotcha_6_gew1 -           17860)
  +     0.2 * max(0,            6858 - gotcha_34_gew1)
  +   0.095 * max(0, gotcha_34_gew1 -            6858)
  +   0.039 * max(0,           21033 -  gotcha_2_gew1)
  -     1.4 * max(0,  gotcha_2_gew1 -           21033)
  -     249 * max(0,              18 -     ma_7_perc2)
  +      38 * max(0,              17 -     ma_6_perc2)
  -     434 * max(0,     ma_6_perc2 -              17)
  +     167 * max(0,    ma_34_perc2 -              13)

gotcha_34_gew2 =
  1.1e+04
  -    2261 * Oneven1
  -     209 * Richting1
  -    1291 * V21
  +   0.092 * max(0,            7400 -  gotcha_1_gew1)
  -   0.053 * max(0,  gotcha_1_gew1 -            7400)
  -    0.15 * max(0,           11188 -  gotcha_5_gew1)
  +   0.055 * max(0,  gotcha_5_gew1 -           11188)
  -    0.18 * max(0,            9284 -  gotcha_7_gew1)
  -   0.053 * max(0,  gotcha_7_gew1 -            9284)
  +   0.046 * max(0,           17860 -  gotcha_6_gew1)
  +    0.67 * max(0,  gotcha_6_gew1 -           17860)
  -    0.32 * max(0,            6858 - gotcha_34_gew1)
  +    0.59 * max(0, gotcha_34_gew1 -            6858)
  -   0.025 * max(0,           21033 -  gotcha_2_gew1)
  -       2 * max(0,  gotcha_2_gew1 -           21033)
  -     221 * max(0,              18 -     ma_7_perc2)
  +      96 * max(0,              17 -     ma_6_perc2)
  -     365 * max(0,     ma_6_perc2 -              17)
  +     165 * max(0,    ma_34_perc2 -              13)

gotcha_2_gew2 =
  1.9e+04
  -    3215 * Oneven1
  -     555 * Richting1
  -    1589 * V21
  +   0.068 * max(0,            7400 -  gotcha_1_gew1)
  -   0.084 * max(0,  gotcha_1_gew1 -            7400)
  -    0.22 * max(0,           11188 -  gotcha_5_gew1)
  +   0.053 * max(0,  gotcha_5_gew1 -           11188)
  -    0.29 * max(0,            9284 -  gotcha_7_gew1)
  -   0.079 * max(0,  gotcha_7_gew1 -            9284)
  +   0.055 * max(0,           17860 -  gotcha_6_gew1)
  +    0.88 * max(0,  gotcha_6_gew1 -           17860)
  +    0.34 * max(0,            6858 - gotcha_34_gew1)
  +    0.25 * max(0, gotcha_34_gew1 -            6858)
  -    0.41 * max(0,           21033 -  gotcha_2_gew1)
  -     1.9 * max(0,  gotcha_2_gew1 -           21033)
  -     297 * max(0,              18 -     ma_7_perc2)
  +     187 * max(0,              17 -     ma_6_perc2)
  -     514 * max(0,     ma_6_perc2 -              17)
  +     137 * max(0,    ma_34_perc2 -              13)
```

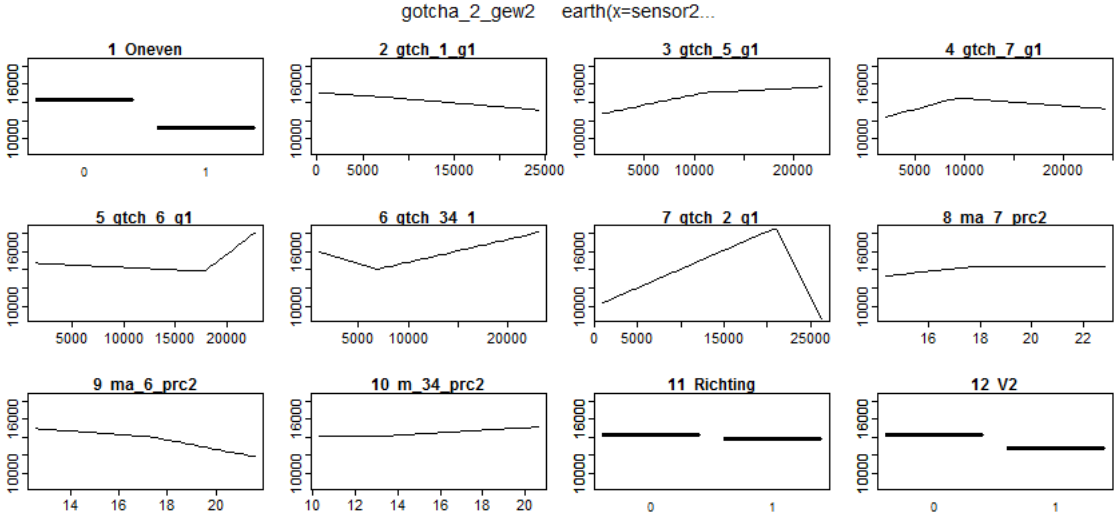

gotcha_1_gew2 earth(x=sensor2...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2


gotcha_5_gew2 earth(x=sensor2...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2


gotcha_7_gew2 earth(x=sensor2...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2


gotcha_6_gew2 earth(x=sensor2...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2


gotcha_34_gew2 earth(x=sensor...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2


gotcha_2_gew2 earth(x=sensor2...
1 Oneven
2 gtch_1_g1
3 gtch_5_g1
4 gtch_7_g1
5 gtch_6_g1
6 gtch_34_1
7 gtch_2_g1
8 ma_7_prc2
9 ma_6_prc2
10 m_34_prc2
11 Richting
12 V2

# B Overige treintellingen

| Datum: | 26-mei | Traject: | Ut-Ht |
|---|---|---|---|
| Treinnummer: | 833 | Aantal reizigers: | 252 |
| Materieelnummer: | 8642 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 17,9% | 16,4% | 1,5% |
| ABv5 | 13,1% | 8,7% | 4,4% |
| mBv7 | 24,2% | 22,9% | 1,3% |
| ABv6 | 11,1% | 15,6% | 4,5% |
| ABv3/4 | 13,5% | 14,1% | 0,6% |
| mBvk2 | 20,2% | 22,2% | 2,0% |
| | | Gemiddelde fout | 2,4% |
| | | Grootste fout | 4,5% |

| Datum: | 26-mei | Traject: | Ht-Ut |
|---|---|---|---|
| Treinnummer: | 300836 | Aantal reizigers: | 249 |
| Materieelnummer: | 8642 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 3,1% | 12,6% | 9,5% |
| ABv5 | 6,9% | 9,3% | 2,5% |
| mBv7 | 18,1% | 16,7% | 1,4% |
| ABv6 | 25,0% | 22,1% | 2,9% |
| ABv3/4 | 30,6% | 21,6% | 9,0% |
| mBvk2 | 16,3% | 17,6% | 1,3% |
| | | Gemiddelde fout | 4,4% |
| | | Grootste fout | 9,5% |

| Datum: | 30-mei | Traject: | Ut-Ht |
|---|---|---|---|
| Treinnummer: | 833 | Aantal reizigers: | 252 |
| Materieelnummer: | 8652 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 20,0% | 22,5% | 2,5% |
| ABv5 | 16,1% | 10,7% | 5,4% |
| mBv7 | 20,6% | 24,7% | 4,1% |
| ABv6 | 14,4% | 9,7% | 4,8% |
| ABv3/4 | 10,0% | 6,5% | 3,5% |
| mBvk2 | 18,9% | 25,8% | 6,9% |
| | | Gemiddelde fout | 4,5% |
| | | Grootste fout | 6,9% |

| Datum: | 30-mei | Traject: | Ht-Ehv |
|---|---|---|---|
| Treinnummer: | 833 | Aantal reizigers: | 197 |
| Materieelnummer: | 8652 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 25,4% | 21,4% | 4,0% |
| ABv5 | 16,8% | 15,3% | 1,4% |
| mBv7 | 19,3% | 18,6% | 0,7% |
| ABv6 | 12,2% | 14,2% | 2,0% |
| ABv3/4 | 11,7% | 12,1% | 0,4% |
| mBvk2 | 14,7% | 18,4% | 3,7% |
| | | Gemiddelde fout | 2,0% |
| | | Grootste fout | 4,0% |

| Datum: | 30-mei | Traject: | Ehv-Wt |
|---|---|---|---|
| Treinnummer: | 833 | Aantal reizigers: | 241 |
| Materieelnummer: | 8652 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 10,0% | 8,7% | 1,3% |
| ABv5 | 14,1% | 16,2% | 2,1% |
| mBv7 | 21,2% | 17,1% | 4,0% |
| ABv6 | 13,3% | 14,3% | 1,0% |
| ABv3/4 | 17,4% | 18,5% | 1,1% |
| mBvk2 | 24,1% | 25,2% | 1,1% |
| | | Gemiddelde fout | 1,8% |
| | | Grootste fout | 4,0% |

| Datum: | 30-mei | Traject: | Wt-Ehv |
|---|---|---|---|
| Treinnummer: | 840 | Aantal reizigers: | 201 |
| Materieelnummer: | 8719 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 12,9% | 6,5% | 6,4% |
| ABv5 | 8,5% | 13,6% | 5,2% |
| mBv7 | 24,4% | 24,6% | 0,2% |
| ABv6 | 17,4% | 23,6% | 6,2% |
| ABv3/4 | 20,9% | 21,7% | 0,8% |
| mBvk2 | 15,9% | 10,1% | 5,8% |
| | | Gemiddelde fout | 4,1% |
| | | Grootste fout | 6,4% |

| Datum: | 30-mei | Traject: | Ehv-Ht |
|---|---|---|---|
| Treinnummer: | 840 | Aantal reizigers: | 230 |
| Materieelnummer: | 8719 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 26,1% | 19,7% | 6,4% |
| ABv5 | 18,7% | 18,5% | 0,2% |
| mBv7 | 23,5% | 20,0% | 3,4% |
| ABv6 | 11,7% | 17,2% | 5,5% |
| ABv3/4 | 10,0% | 13,4% | 3,4% |
| mBvk2 | 10,0% | 11,3% | 1,3% |
| | | Gemiddelde fout | 3,4% |
| | | Grootste fout | 6,4% |

| Datum: | 30-mei | Traject: | Ht-Ut |
|---|---|---|---|
| Treinnummer: | 840 | Aantal reizigers: | 288 |
| Materieelnummer: | 8719 | | |
| Reizigersverdeling | | | |
| | Tellingen | Gotcha | Verschil |
| mBvk1 | 26,4% | 22,1% | 4,3% |
| ABv5 | 20,1% | 20,1% | 0,0% |
| mBv7 | 22,6% | 19,1% | 3,5% |
| ABv6 | 11,8% | 15,7% | 3,9% |
| ABv3/4 | 10,4% | 12,0% | 1,6% |
| mBvk2 | 8,7% | 11,0% | 2,3% |
| | | Gemiddelde fout | 2,6% |
| | | Grootste fout | 4,3% |

# Bibliografie


[1] H. Kim, S. Kwon, S. K. Wu en K. Sohn, „Why do passengers choose a specific car of a metro train during the morning peak hours?," *Transportation Research Part A: Policy and Practice,* nr. 61, pp. 249-258, 2014.

[2] P. B. Wiggenraad, „Alighting and boarding times of passengers at Dutch railway stations," TRAIL Research School, Delft, 2001.

[3] N. Krstanoski, „Modelling passenger distribution on metro station platform," Faculty for Technical Sciences, Department for Transport and Traffic Engineering, University "St. Kliment Ohridski, Bitola, Macedonia, 2014.

[4] B. Mori, "A Study of the Dwell Time at Urban Rail Transit Stations," Department of Civil Enigineering, University of Toronto, 1988.

[5] Z. Liu, D. Li en X. Wang, „Passenger Distribution and Waiting Position Selection Model on Metro Station Platform," in *International Conferencce on Civil, Structure, Environmental Engineering*, Beijing, 2016.

[6] D. Szplett en S. Wirasinghe, „An Investigation of Passenger Interchange and Train Standing at LRT Stations," *Journal of Advanced Transportation,* 1984.

[7] OpenCapacity, „http://opencapacity.co/," [Online].

[8] NS. [Online]. Available: http://www.ns.nl/over-ns/de-spoorsector/verantwoordelijkheden.html. [Geopend 15 Augustus 2016].

[9] NS. [Online]. Available: http://www.ns.nl/over-ns/treinen-van-ns. [Geopend 15 augustus 2016].

[10] R. Volgers, A. Kok en M. Golstein, „Massaoverzicht VIRM-1 & VIRMm-1," Nedtrain, 2015.

[11] „NS sharepoint," [Online]. Available: https://nsdigitaal.sharepoint.com/teams/kh/SiteAssets/Spoorkaart_Gotcha.gif. [Geopend 29 Augustus 2016].

[12] „KNMI klimatologie," [Online]. Available: http://www.knmi.nl/nederland-nu/klimatologie/uurgegevens. [Geopend 3 Juni 2016].

[13] T. Segaran, Programming Collective Intelligence, O'Reilly Media, 2007.

[14] J. H. Friedman, „Multivariate adaptive regression splines," *The Annals of Statistics,* nr. 19, pp. 1-141, 1991.

[15] „Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines. [Geopend 15 Augustus

2016].

[16] „Dell," [Online]. Available: https://documents.software.dell.com/statistics/textbook/multivariate-adaptive-regression-splines. [Geopend 15 Augustus 2016].

[17] S. Milborrow, „R-package: 'earth'," [Online]. Available: https://cran.r-project.org/web/packages/earth/earth.pdf.

[18] C. M. Bishop, Neural Networks for Pattern Recognition, 1995.

[19] „Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Geopend 14 Augustus 2016].

[20] F. Günther en S. Fritsch, „neuralnet: Training of Neural Networks," *The R Journal,* 2010.

[21] B. Ripley en W. Venables, „R-package: 'nnet'," [Online]. Available: https://cran.r-project.org/web/packages/nnet/nnet.pdf.

[22] H. Strasser en C. Weber, „On the asymptotic theory of permutation statistics," 1999.

[23] L. Breiman, J. Friedman, R. Ohlsen en C. Stone, „Classification and regression trees," Wadsworth International Group, 1984.

[24] T. Hothorn en A. Zeileis, „R-package: 'partykit'," [Online]. Available: https://cran.r-project.org/web/packages/partykit/partykit.pdf.

[25] S. Milborrow, „Notes on the earth package," 2016.

[26] G. Panchal, A. Ganatra, Y. P. Kosta en D. Panchal, „Behaviour Analysis of Multilayer Perceptrons," *International Journal of Computer Theory and Engineering,* nr. 3, 2011 .

[27] S. Karsoliya, „Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," *International Journal of Engineering Trends and Technology,* nr. 31, 2012.

[28] J. Sola en J. Sevilla, „Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science,* vol. 3, nr. 44, pp. 1464 - 1468, 1997.

[29] J. D. Olden, M. K. Joy en R. G. Death, „An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data," *Ecological Modelling,* p. 389–397, 2004.

[30] C. Molnar, „Recursive partitioning by conditional inference," Munich, 2013.