ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus School of Economics

Bachelorscriptie programma BA and Quantitative Marketing

Het voorspellen van consumentenkeuze: een vergelijking van een statistische methode en machine learning technieken

Naam student: Sacha Roosenstein

Studentnummer: 457814

Begeleid door: Anoek Castelein

Datum definitieve versie: 25-06-2019

Abstract

Voor bedrijven is het cruciaal dat consumentenkeuze nauwkeurig wordt voorspeld. Op deze manier kunnen bedrijven marketing variabelen, zoals de prijs, op zo'n manier aanpassen dat de winstgevendheid stijgt. Al jaren is het Multinomiale Logit model het meest gebruikte model in het modelleren van consumentenkeuze. De laatste jaren is er een groeiende interesse ontstaan in het gebruik van machine learning technieken om consumentenkeuze te voorspellen. In dit onderzoek worden het Multinomiale Logit model en de machine learning technieken Feedforward Neural Network, CART, Bagging, Boosting en Random Forest met elkaar vergeleken. De prestatie van de methoden wordt vergeleken op basis van voorspelnauwkeurigheid en interpretatie. Hiervoor zijn twee datasets gebruikt, deze bevatten gegevens over de aankoopgeschiedenis van de productcategorieën ketchup en cracker. Voor deze datasets is gevonden dat Random Forest en CART de beste prestatie opleveren en dat Boosting en Feedforward Neural Network de minste voorkeur hebben.

Inhoudsopgave

1 Introductie

Het voorspellen van consumentengedrag is belangrijk voor bedrijven. Een nauwkeurige voorspelling kan zorgen voor een substantiële verbetering in de winstgevendheid van het bedrijf. Om te onderzoeken wat de impact is van marketingmix variabelen op consumentenkeuze, is het belangrijk om te begrijpen hoe consumentengedrag wordt beïnvloed door deze variabelen. Het doel is een wiskundige relatie te vinden tussen de afhankelijke variabele (de consumentenkeuze) en de onafhankelijke variabelen. Omdat de afhankelijke variabele discreet is, kan dit worden gezien als een classificatieprobleem. Door de toenemende beschikbaarheid van scannergegevens van aankoopgeschiedenissen kan hier steeds meer onderzoek naar worden gedaan (Bentz & Merunka, 2000). In dit onderzoek wordt de prestatie van verschillende methoden vergeleken in het voorspellen van consumentenkeuze. De onderzoeksvraag van dit onderzoek is: *Welke methode is het best geschikt voor het voorspellen van consumentenkeuze?*

Verscheidene modellen kunnen worden gebruikt voor het voorspellen van consumentenkeuze. Het Multinomiale Logit model (MNL) (McFadden, 1973) is al vele jaren het meest gebruikte model. Dit model is aantrekkelijk vanwege de eenvoudige interpreteerbaarheid. Echter, voor niet-lineaire classificatieproblemen is MNL minder geschikt omdat het model is gebaseerd op een lineaire nutsfunctie (Bentz & Merunka, 2000). Een ander nadeel van MNL is dat deze uitgaat van *Independence of Irrelevant Alternatives* (IIA) (Hagenauer & Helbich, 2017). Wanneer deze aanname niet opgaat kan dit leiden tot inconsistente parameter schattingen (McFadden, 1973).

De afgelopen jaren is er een groeiende interesse in machine learning technieken. Deze technieken gaan in de meeste gevallen niet uit van strikte aannamen en bieden mogelijkheden om complexe relaties tussen de afhankelijke variabele en onafhankelijke variabelen te achterhalen (Bentz & Merunka, 2000). Het Artificial Neural Network (ANN) is één van de populairste machine learning technieken. ANN's zijn gebaseerd op het biologische concept van neuronen structuur in het menselijk brein en kunnen worden gebruikt voor classificatieproblemen. Echter, ANN's zijn data afhankelijk en daardoor hangt de prestatie af van de grootte van de dataset (Razi & Athappilly, 2005). Daarbij gaat het nauwkeuriger voorspellen van ANNs samen met een slechtere interpreteerbaarheid. Machine learning technieken die deze beperkingen wellicht voorkomen zijn *tree-based* methoden. Classification and Regression Trees (CART) is een beslissingsboom algoritme en is beter geschikt voor kleinere datasets (Razi & Athappilly, 2005). Echter, deze methode heeft de neiging de data

te overfitten. *Ensemble* methoden zoals Bagging, Boosting en Random Forest (RF) combineren meerdere beslissingsbomen en hebben daardoor minder de neiging de data te overfitten (Van Wezel & Potharst, 2007).

Verschillende onderzoeken zijn gedaan naar het vergelijken van methoden in het voorspellen van consumentenkeuze. Agrawal en Schorling (1996) vergelijken MNL met een ANN, genaamd Feedforward Neural Network (FNN), en vonden dat FNN in sommige gevallen beter presteert dan MNL. Van Wezel en Potharst (2007) vonden dat ensemble methoden beter presteren dan CART en MNL in het voorspellen van consumentenkeuze. In dit onderzoek worden MNL, FNN, CART, Bagging, RF en Boosting met elkaar vergeleken. Om de prestatie van de methoden te vergelijken worden de methoden beoordeeld op voorspelnauwkeurigheid en interpretatie. Voor dit onderzoek wordt gebruik gemaakt van 2 data sets van de aankoop categorieën ketchup en cracker.

Dit onderzoek is als volgt ingedeeld. Sectie 2 bevat een overzicht van de relevante literatuur. Een beschrijving van de geanalyseerde data wordt gegeven in Sectie 3. Sectie 4 geeft de methoden weer die gebruikt worden in dit onderzoek en Sectie 5 geeft de verkregen resultaten weer. De conclusie wordt gegeven in Sectie 6.

2 Literatuur

In deze sectie wordt een overzicht van de relevante literatuur gegeven. Sectie 2.1 beschrijft klassieke statistische methoden. Sectie 2.2 beschrijft machine learning technieken.

2.1 Klassieke statistische methoden

Er bestaat een aantal veel gebruikte statistische methoden voor het modelleren van discrete keuze. Een aantal voorbeelden zijn: Logistische regressie (Kumar, Rao & Soni, 1995), Multiple regressie (Neter, Kutner, Nachtsheim & Wasserman, 1996), Discriminant Analyse (Press & Wilson, 1978), het Multinomiale Probit model (Paap & Franses, 2000) en het Multinomiale Logit Model (Agrawal & Schorling, 1996). Van deze methoden is MNL het meest gebruikte model voor het modelleren van discrete consumentenkeuze (Bentz & Merunka, 2000).

2.2 Machine learning

Het gebruik van machine learning is de afgelopen jaren flink gegroeid. Uit de literatuur is gebleken dat de ontwikkeling en toepassing van machine learning technieken zich niet beperkt tot een specifiek gebied. In Sectie 2.2.1 en 2.2.2 worden respectievelijk ANN en tree-based methoden besproken.

2.2.1 Artificial Neural Network

ANN's behoren tot de meest gebruikte machine learning technieken en zijn geïnspireerd door het menselijk brein. ANN's bestaan uit een gelaagde structuur waarin zich artificial neuronen en verbindingen tussen de neuronen bevinden (Rojas, 2013). In het menselijk brein leert men door de verbinding tussen verschillende neuronen te versterken of te verzwakken. Het ANN leert door gewichten tussen de neuronen aan te passen. Een aantal voorbeelden van Neural Networks zijn het Convolutional Neural Network (CNN), het Recurrent Neural network (RNN) en het Feedforward Neural Network (FNN). Het CNN is met name populair op het gebied van afbeelding herkenning (Krizhevsky, Sutskever & Hinton, 2012; Sermanet e.a., 2013). RNNs worden veel toegepast voor spraakherkenningsdoeleinden (Sak, Senior, Rao & Beaufays, 2015) en worden toenemend gebruikt in tijdreeksanalyse (Malhotra, TV, Vig, Agarwal & Shroff, 2017). Het FNN wordt vaker gebruikt voor het voorspellen van consumentenkeuze (Agrawal & Schorling, 1996; Kumar e.a., 1995).

ANN's overtreffen in verschillende onderzoeksgebieden de statistische methoden. Uit het onderzoek van Shang, S, Yu-sand en Goetz (2000) bleek dat ANN's soms beter presteren dan logistische regressie in het opsporen van antibioticaresistente infecties. Tam en Kiang (1992) pasten discriminant analyse en ANN toe om het falen van banken te toetsen. Hieruit bleek dat ANN's beter kunnen voorspellen dan discriminant analyse modellen. Ook in de marketing zijn er talloze toepassingen van ANN's beschikbaar in de literatuur waar ANN de statistische methoden overtreft (West, Brockett & Golden, 1997; Kumar e.a., 1995).

2.2.2 Tree-based methoden

Breiman, Friedman, Olshen en Stone (1984) ontwikkelden de machine learning techniek CART. Dit is een beslissingsboom algoritme dat gebruikt kan worden voor classificatie en regressie problemen. Deze machine learning techniek is zowel geschikt voor het modeleren van complexe datasets als voor kleinere datasets. Razi en Athappilly (2005) vergelijken CART met een ANN en een niet-lineair regressie model voor het modelleren van een continue afhankelijke variabele. Zij vonden dat CART beter presteerde dan het regressie model maar niet beter dan ANN. Een reden hiervoor kan zijn dat CART te afhankelijk is van de trainingsperiode waardoor het de data overfit. Om dit te voorkomen kunnen ensemble methoden worden gebruikt (Van Wezel & Potharst, 2007).

De meest gebruikte ensemble methoden zijn Bagging (Breiman, 1996), RF (Breiman, 2001) en Boosting (Schapire, 1990). Bagging is een relatief simpele ensemble techniek waarbij vele beslis-

singsbomen simultaan worden getraind door bootstrap samples van de dataset te gebruiken. RF lijkt op Bagging, maar voegt een extra laag willekeurigheid toe aan het model (Liaw, Wiener e.a., 2002). Naast het construeren van de beslissingsbomen door middel van bootstrapping, verandert RF de manier waarop de beslissingsbomen worden geconstrueerd. Boosting is een meer geavanceerde techniek waarbij weak-learners opeenvolgend worden getraind om vervolgens een "strong" leeralgortime te verkijgen. Elke weak-learner probeert de voorgaande weak-learner te verbeteren. Een weak-learner is een leeralgortime dat net beter presteert dan een willekeurige gok. Een voorbeeld van een weak-learner is een beslissingsboom met één splitsing (Freund, Schapire & Abe, 1999). Kearns (1988) was de eerste die zich afvroeg of een "weak" leeralgortime kan worden "geboost" om zo een "strong" leeralgoritme te verkrijgen. Schapire (1990) kwam met het eerste Boosting algortime. Later werd een efficienter Boosting algortime, AdaBoost, ontwikkeld door Freund en Schapire (1997). Uit empirische resultaten blijkt dat de gecombineerde voorspellingen vaak accurater zijn dan individuele voorspellingen (Van Wezel & Potharst, 2007).

3 Data beschrijving

Voor dit onderzoek worden twee datasets gebruikt. Dataset 1 (verkregen door A.C. Nielsen) bevat gegevens van 300 huishoudens over de aankoopgeschiedenis van vier soorten ketchup producten, met een totaal van 2798 aankopen. De vier producten, genaamd Heinz41, Heinz32, Heinz28 en Hunts32, zijn respectievelijk in 7%, 52% 30% en 11% van de aankopen gekocht. Hieruit blijkt dat de verdeling relatief scheef is. Het getal achter de merknaam staat voor de inhoud van het product, uitgedrukt in de Amerikaanse maat *oz*. Dataset 2 (verkregen via Information Resourses Incorporated) bevat gegevens van 136 huishoudens over de aankoopgeschiedenis van vier producten in de categorie crackers: Sunshine, Kleebler, Nabisco en de private labels (Private), met een totaal van 3292 aankopen. Sunshine, Kleebler, Nabisco en Private zijn respectievelijk in 7%, 7%, 54% en 31 % van de aankopen gekocht. Voor dataset 2 geldt wederom dat de verdeling scheef is.

Voor beide datasets is er informatie over de marketingmix variabelen *display*, *feature* en *price*. Display en feature zijn binaire variabelen. Een product is in display als het product op een speciale plek in het schap wordt tentoongesteld en een product is in feature wanneer het product is geadverteerd in een nieuwsblad. Voor de prijsvariabele geldt dat dit de prijs weergeeft van het gekochte product, na aftrek van persoonlijke kortingen (Jain, Vilcassim & Chintagunta, 1994). Voor dataset

1 zijn in dit onderzoek de prijzen aangepast alsof de inhoud van alle producten 32 oz is. Op deze manier hangen de prijzen niet van de inhoud af. De afhankelijke variabele is de discrete aankoopkeuze van de consument. In appendix A zijn karakteristieken van de twee datasets weergegeven.

3.1 Train en test periode

In dit onderzoek wordt een trainingsperiode van 80% van de huishoudens gebruikt voor het trainen van de modellen. De overige 20% huishoudens wordt gebruikt als testperiode om de voorspelnauwkeurigheid te bepalen.

4 Methodologie

In deze sectie worden de methoden beschreven die gebruikt worden om de onderzoeksvraag te beantwoorden. Sectie 4.1 beschrijft het MNL en Sectie 4.2 beschrijft het FNN. Sectie 4.3 beschrijft het CART algoritme en de ensemble methoden Bagging, RF en Boosting. Sectie 4.4 beschrijft hoe de voorspelnauwkeurigheid wordt bepaald. De methoden die zijn toegepast in dit onderzoek zijn geprogrammeerd in MATLAB.

4.1 Multinomiale logit model

Het MNL is gebaseerd op een lineaire nutsfunctie. Ga er van uit dat huishouden i nut U_{ijt} bereikt als het product j koopt op tijdstip t, en laat k het totaal aantal onafhankelijke variabelen zijn. Het nut wordt als volgt berekend (McFadden, 1973):

$$U_{ijt} = \alpha_j + x'_{ijt}\beta + \varepsilon_{ijt}, \quad i = 1, ..., n \quad t = 1, ..., T \quad j = 1, ..., J \tag{1}$$

waarbij x_{ijt} een (k x 1) vector is van onafhankelijke variabelen, β een (k x 1) parameter vector en α_j de intercept parameter behorend bij product j. De parameter vector hangt niet af van j, wat wilt zeggen dat ze een gelijk effect op de keuzekans van de J producten hebben. Verder geldt dat ε_{ijt} een willekeurige kans variabele is. Ga er van uit dat Y de discrete afhankelijke variabele weergeeft. De kans dat huishouden i product j kiest op tijdstip t is:

$$\pi_{ijt} = Pr[Y_{it} = j|X_{it}] = Pr[max(U_{i1t}, ..., U_{iJt}) = U_{ijt}]. \tag{2}$$

Wanneer wordt aangenomen dat ε_{ijt} een onafhankelijke, identieke *type-I extreme value* verdeling volgt met parameters α_j en β, kan vergelijking 2 geschreven worden als het MNL (McFadden, 1973):

$$\pi_{ijt} = \frac{exp(\alpha_j + x'_{ijt}\beta)}{\sum_{l=1}^{J} exp(\alpha_l + x'_{ilt}\beta)}. \tag{3}$$

Er moet worden opgemerkt dat voor parameter identificatie de identificatie restrictie $\alpha_J = 0$ wordt gesteld (een bewijs hiervoor is gegeven in appendix B).

4.1.1 Product loyaliteit

Het model kan worden uitgebreid door de extra variabele product loyaliteit toe te voegen (Guadagni & Little, 1983). Loyaliteit aan product j van huishouden i op tijdstip t is als volgt gedefinieerd:

$$b_{i,j,t} = \delta b_{i,j,t-1} + (1 - \delta) I[y_{i,t} = j], \tag{4}$$

waarbij $0 < \delta < 1$, $I[\cdot]$ een $0/1$ indicator functie is en $y_{i,t}$ de werkelijke keuze van huishouden i op tijdstip t weergeeft. Als initialisatie wordt gebruikt dat $b_{i,j,0}$=0 en dat

$$b_{i,j,1} = \begin{cases} \delta, & \text{als } y_{i,1} = j \\ (1 - \delta)/(J - 1), & \text{als } y_{i,1} \neq j. \end{cases}$$

Om δ te bepalen wordt gebruik gemaakt van *cross-validation*; Voor verschillende waarden voor δ wordt het model geschat met 75 procent van de training dataset en voorspeld met 25 procent van de training dataset, ook wel de validatieperiode genoemd. Als prestatiemaatstaaf wordt de Mean Absolute Error (MAE) gebruikt, deze is als volgt gedefinieerd:

$$MAE = \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} \sum_{j=1}^{J} |\hat{\pi}_{ijt} - I[y_{i,t} = j]|}{n \text{ x } T \text{ x } J}, \tag{5}$$

waarbij $\hat{\pi}_{ijt}$ de voorspelde kans weergeeft dat huishouden i op tijdstip t product j kiest. De waarde voor δ met de laagste MAE wordt gebruikt. Ga er van uit dat γ de loyaliteit parameter weergeeft. Het model kan nu als volgt worden geschreven:

$$\pi_{ijt} = \frac{exp(\alpha_j + x_{ijt}\beta + b_{i,j,t-1}\gamma)}{\sum_{l=1}^{J} exp(\alpha_l + x_{ilt}\beta + b_{i,k,t-1}\gamma)}. \tag{6}$$

4.1.2 Maximum likelihood

Met gebruik van de trainingsperiode worden de parameters in MNL geschat door middel van Maximum Likelihood. De likelihood functie is gedefinieerd als het product van de kansen van het gekozen product voor alle huishoudens. In vergelijking (7) en (8) zijn respectievelijk de likelihood functie en de log-likelihoodfunctie weergegeven. Hierbij geeft θ de model parameters weer en is $I[\cdot]$ een $0/1$ indicator functie. De parameter schattingen worden verkregen door het maximaliseren van de (log-)likelihood functie. Omdat deze niet-lineair is in de parameters wordt hiervoor een numeriek

optimalisatie algoritme, quasi-Newton, gebruikt. Voor het optimaliseren is de MATLAB functie *fminunc* gebruikt.

$$L(\theta) = \prod_{i=1}^{N}\prod_{t=1}^{T}\prod_{j=1}^{J}\pi_{ijt}^{I[y_{it}=j]} \qquad (7) \qquad l(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{j=1}^{J} I[y_{it}=j]log(\pi_{ijt}). \qquad (8)$$

Om de standaard fout te berekenen wordt eerst de covariantie matrix als volgt bepaald:

$$\widehat{var}(\hat{\theta}) = \mathcal{I}_n^{-1}(\hat{\theta}) = E[\frac{\delta l}{\delta\theta}\frac{\delta l}{\delta\theta'}]^{-1}, \qquad (9)$$

waarbij $\mathcal{I}_n$ de Fisher informatie matrix is. Vervolgens wordt de standaardfout van parameter i berekend door de wortel te nemen van het i^{de} diagonaal element van $\widehat{var}(\hat{\theta})$.

4.1.3 Odds ratio

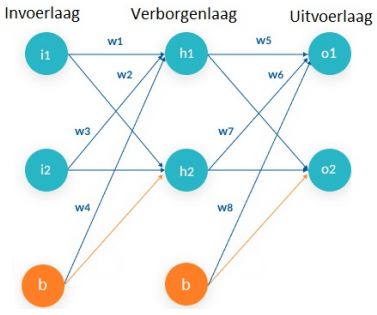
Het MNL heeft zijn populariteit te danken aan de goede interpreteerbaarheid. De interpreteerbaarheid is niet rechtstreeks uit het model af te leiden aangezien het model niet-lineair is in de model parameters. Om het effect te begrijpen worden *odds ratios* gebruikt. De odds ratio van product j versus product l is de relatieve voorkeur van product j vergeleken met product l, en is als volgt gedefinieerd (Franses & Paap, 2001):

$$\Omega_{j|l}(X_{it}) = \frac{\pi_{ijt}}{\pi_{ilt}} = \frac{exp(\alpha_j + x_{ijt}\beta)}{exp(\alpha_l + x_{ilt}\beta)} = exp((\alpha_j - \alpha_l) + \beta(x_{ijt} - x_{ilt})) \qquad (10)$$

De intercept parameters kunnen worden gezien als basis voorkeur van de huishoudens; Product j wordt geprefereerd boven product l als $\alpha_j > \alpha_l$. Voor positieve waarden van β heeft product j de voorkeur boven product l, wanneer $x_{ijt} > x_{ilt}$.

4.2 Feedforward Neural Network

Een ANN kan worden gezien als een niet-lineair statistisch model (Bentz & Merunka, 2000). Een ANN is opgebouwd uit verschillende lagen: een *input layer*, één of meer *hidden layers* en een *output layer*. In dit onderzoek wordt één hidden layer gebruikt. Figuur 1 geeft een illustratie van een ANN met drie lagen. De verschillende lagen bestaan uit knooppunten. Elke verbinding tussen de knooppunten is geassocieerd met een gewicht (weergegeven door w). Het aantal knooppunten in de input layer is gelijk aan (k x J), waarbij k het aantal onafhankelijke variabelen is en J het aantal producten. Er kunnen eventueel *bias* knooppunten toegevoegd worden aan het netwerk, deze zijn weergegeven door knooppunt b in Figuur 1. Door



Figuur 1: Drie laagse ANN

het toevoegen van bias knooppunten kunnen net zoals bij MNL intercept parameters meegenomen worden (Trevor, Robert & JH, 2009). In dit onderzoek worden de bias knooppunten toegevoegd omdat deze kunnen worden gezien als basis voorkeur van de huishoudens. Het aantal knooppunten in de input layer is daardoor (k x J)+1. Het aantal knooppunten in de output layer is gelijk aan J en het aantal knooppunten in de hidden layer wordt gekozen door de gebruiker.

Voor FNN geldt dat de output van de knooppunten in laag m wordt getransformeerd en dient als input voor de knooppunten in laag $m+1$, waarbij de knooppunten hun output alleen in een voorwaartse richting kunnen doorgeven. De waarde van knooppunt j in hidden layer m wordt als volgt berekend:

$$h_{j,m} = \sigma(\sum_{i=1}^{J_{(m-1)}} w_{ij,m} h_i^{m-1}), \quad j = 1, ..., J_m, \quad m = 1, ..., M \tag{11}$$

waarbij $\sigma(v_j)$ een activatie functie is. Hiervoor wordt in dit onderzoek de sigmoid functie gebruikt: $\sigma(v_j) = \frac{1}{1+exp(-v_j)}$. De input knooppunten worden weergegeven door h_i^0. Verder geldt dat $w_{ij,m}$ het gewicht van de verbinding tussen knooppunt i in laag $m-1$ en knooppunt j in laag m weergeeft en J_m het aantal knooppunten in hidden layer m. De output knooppunten π_j worden als volgt berekend:

$$\pi_j = \frac{exp(\sum_{i=1}^{J_M} w_{ij,m} h_i^M)}{\sum_{l=1}^{J} exp(\sum_{i=1}^{J_M} w_{il,m} h_i^M))}, \quad j = 1, ..., J \tag{12}$$

waarbij M correspondeert met de laatste hidden layer en J het aantal producten is, oftewel het aantal outpout knooppunten. De transformatie in vergelijking (12) wordt ook wel de softmax functie genoemd en is exact de transformatie die gebruikt wordt in MNL en geeft als uitkomst J kolomvectoren van kansen die per rij optellen tot 1.

4.2.1 Backpropagation

Om de gewichten te vinden waarvoor het FNN de optimale output geeft wordt het *back-propagation* algoritme gebruikt. Dit algoritme begint met kleine, willekeurige initiële gewichten. Vervolgens wordt het feedforward algoritme toegepast en de output berekend. Om te bepalen of de gewichten moeten worden aangepast, wordt gebruik gemaakt van een *loss*-functie. Hiervoor kan de *Cross-Entropy* functie of de *Mean Squared Error* (MSE) functie gebruikt worden. Deze zijn beiden geschikt voor classificatie (Trevor e.a., 2009). in dit onderzoek wordt de MSE functie gebruikt:

$$MSE = \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} \sum_{j=1}^{J} (\hat{\pi}_{ijt} - I[y_{it} = j])^2}{n \text{ x } T \text{ x } J} \tag{13}$$

Wanneer de loss-functie te groot is worden de gewichten als volgt aangepast:

$$\Delta w_{ij,m} = -\eta \frac{\partial L}{\partial w_{ij,m}}, \tag{14}$$

waarbij $\Delta w_{ij,m}$ de verandering in het gewicht is van de verbinding tussen knooppunt i in laag $m-1$ en knooppunt j in laag m, η de *learning rate* is (deze wordt gekozen door de gebruiker) en L de loss-functie is. Vervolgens wordt met de nieuwe gewichten opnieuw de output berekend. Elke keer wanneer de gewichten worden aangepast wordt het netwerk doorlopen. Het doorlopen van het netwerk wordt een *training epoch* genoemd. Dit proces wordt toegepast op de trainingsperiode en wordt herhaald totdat de loss-functie voldoende dichtbij nul zit of totdat het maximum aantal training epochs is bereikt.

4.2.2 K-fold cross validation

Om het FNN te kunnen toepassen moet bepaald worden hoeveel knooppunten zich in de hidden layer bevinden en wat het maximum aantal training epochs is. Om de beste combinatie van training epochs en knooppunten te vinden wordt *k-fold cross validation* toegepast op de trainingsperiode met verschillende combinaties van knooppunten en training epochs. Hierbij wordt de dataset die gebruikt wordt voor het trainen opgedeeld in $k = 10$ willekeurige sub samples van gelijke grootte. Voor elke subset wordt een voorspelling gedaan door de overige k-1 subsets te gebruiken voor het trainen van het netwerk. Dit wordt gedaan zodat de uitkomsten niet afhangen van één bepaalde periode. Voor de tien verschillende voorspellingen wordt de Mean Absolute Error (MAE) gebruikt als prestatiemaatstaaf (zie vergelijking (5)). Om vervolgens 1 prestatiemaatstaaf te verkrijgen wordt het gemiddelde genomen over de tien MAE uitkomsten. De combinatie knooppunten en training epochs met de kleinste gemiddelde MAE wordt samen met de optimale gewichten toegepast op de testperiode om de voorspelnauwkeurigheid van het netwerk te bepalen.

4.3 Tree-based methoden

Er worden vier tree-based methoden toegepast in dit onderzoek. Sectie 4.3.1 beschrijft het beslissingsboom algoritme CART. Vervolgens worden er drie tree-based ensemble technieken beschreven. Deze technieken combineren een groot aantal beslissingsbomen om zo een accuratere classificatie te verkrijgen dan wanneer een enkele beslissingsboom wordt gebruikt. Sectie 4.3.2, 4.3.3 en 4.3.4 beschrijven respectievelijk de ensemble methoden Bagging, RF en Boosting.

4.3.1 Classification And Regression Tree

CART is een beslissingsboom algoritme dat gebruikt kan worden voor classificatieproblemen. Een beslissingsboom bestaat uit knooppunten die de voorspelruimte recursief opdelen door middel van splitsingen die zijn gebaseerd op de onafhankelijke variabelen. Deze splitsingen worden bepaald aan de hand van de onzuiverheid van de mogelijke splitsingen. Voor binaire variabelen is er slechts één splitsing mogelijk en hoeft de onzuiverheid uitsluitend voor deze splitsing bepaald te worden. Voor continue variabelen geldt dat voor elke mogelijke splitsing van de dataset de onzuiverheid wordt bepaald. Uit alle mogelijke splitsingen wordt de 'beste' splitsing, oftewel de splitsing met de laagste onzuiverheid, gekozen. De boom stopt met het splitsen van een knooppunt wanneer alle observaties in dat knooppunt bij dezelfde klasse (product) behoren. Mogelijke maatstaven voor onzuiverheid zijn de Gini index en Entropy functie (Breiman, 2017). In dit onderzoek wordt de Gini index gebruikt. De Gini index van de splitsing van knooppunt p in k sub knooppunten is als volgt berekend:

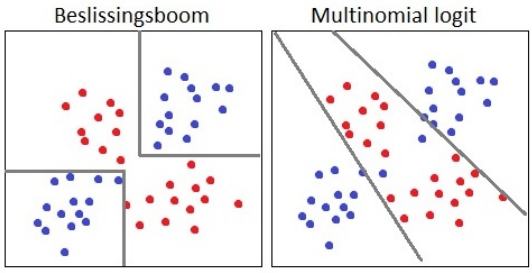
$$GINI_{split(p)} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i), \tag{15}$$

waarbij n het aantal observaties in knooppunt p is en n_i het aantal observaties in sub knooppunt i. $GINI(i)$ geeft de onzuiverheid van sub knooppunt i weer, deze is als volgt is berekend:

$$GINI(i) = 1 - \sum_{j} (p(j|i))^2, \tag{16}$$

waarbij $p(j|i)$ de relatieve frequentie van klasse j gegeven knooppunt i is. Voor het CART algoritme geldt dat er uitsluitend binaire splitsingen worden toegepast. In figuur 2 is weergegeven hoe MNL en CART een classificatie probleem aanpakken. Het is duidelijk dat CART beter geschikt is voor een niet-lineair classificatie probleem. Ondanks dat beslissingsbomen goede benaderingen zijn voor niet-lineaire classificatiegrenzen zijn ze instabiel. Zoals vermeld in Sectie 2 hebben beslissingsbomen de neiging data te overfitten. Door de strikte beslissingsregels die beslissingsbomen gebruiken kunnen ze de data perfect fitten. Dit zorgt ervoor dat de implementatie van beslissingsbomen sterk afhangt van de compositie van de trainingsperiode (Van Wezel & Potharst, 2007). Ensemble methoden kunnen het overfitten wellicht voorkomen.

Het CART algoritme is een simpel algoritme. Echter, het programmeren van de beslissingsboom-structuur is ingewikkeld wat betreft het opslaan van informatie en het ophalen van de opgeslagen informatie. Om deze reden wordt hiervoor de ingebouwde MATLAB functie *fitctree* gebruikt.



Figuur 2: Vergelijking CART en MNL

4.3.2 Bagging

Bootstrap aggregating, afgekort 'Bagging', is een eenvoudige ensemble techniek waarbij bootstrapping wordt gebruikt om een geaggregeerd model te creëren. De input is de training dataset. Ga er van uit dat de trainingsperiode N observaties bevat. Er worden M bootstrap datasets gecreëerd door telkens N willekeurige datapunten te trekken uit de training dataset, met terugleggen. Op deze manier bevatten de bootstrap datasets evenveel observaties als de training dataset. Voor elke bootstrap dataset wordt een beslissingsboom getraind met behulp van CART. De verkregen M beslissingsbomen worden gebruikt om de testset M keer te voorspellen. Om een geaggregeerd model te krijgen wordt gebruik gemaakt van *voting*; Voor elk huishouden wordt het product dat het meest voorspeld is door de M beslissingsbomen gekozen als uiteindelijke consumentenkeuze. Het aantal beslissingsbomen M dat wordt getraind wordt bepaald met cross-validation, waarbij 25 procent van de training dataset wordt gebruikt als validatieperiode.

4.3.3 Random Forest

Random Forest is een ensemble techniek die nauw verwant is aan Bagging. Net zoals bij Bagging worden M beslissingsbomen getraind met gebruik van M bootstrap datasets. Het verschil is dat er extra willekeurigheid wordt toegevoegd; Wanneer een beslissingsboom wordt gecreëerd wordt naast het willekeurig kiezen van N observaties, J onafhankelijke variabelen waarmee de splitsingen worden bepaald willekeurig gekozen, met terugleggen. Op deze manier is een beslissingsboom afhankelijk van een willekeurig aantal onafhankelijke variabelen. Hierdoor zijn de bomen beter in staat verschillende patronen te ontdekken in de dataset (Van Wezel & Potharst, 2007). Het aantal beslissingsbomen dat wordt gebruikt voor het trainen wordt met cross-validation bepaald, op eenzelfde manier als bij Bagging.

4.3.4 Boosting

De derde ensemble methode die in dit onderzoek wordt onderzocht is 'Boosting'. In dit onderzoek wordt gebruik gemaakt van het AdaBoost.M1 algoritme, weergegeven in Algoritme 1 (Freund, Schapire e.a., 1996). Dit is een complexere techniek dan Bagging. Het algemene idee bij Boosting is dat er een opeenvolging van weak-learners wordt gecreëerd, waarbij elke weak-learner de onjuiste classificatie van de voorgaande weak-learners probeert te verbeteren.

Input: Training dataset, S, met observaties (x_{it}, y_{it}), $i = 1, ..., n$, $t = 1, ..., T$.

Weak-learner algoritme *WeakLearn*

Aantal iteraties M

Initialisatie: $D_{1(it)} = 1/(n \text{ x } T)$, $\forall\ i, t$

Voor $m = 1, ..., M$

1. Voer algoritme *WeakLearn* uit door te samplen uit de verdeling D_m, en krijg als uitkomst de weak-learner, h_m. Zet na het samplen de gewichten terug op $1/(n \text{ x } T)$.
2. Bereken de error van h_m met input training dataset S: $\epsilon_m = \sum_{i=1}^{N} D_{m(i)} I[h_m(x_{it}) \neq y_{it}]$.
3. Bereken het belang van h_m: $\alpha_m = \frac{1-\epsilon_m}{\epsilon_m}$.
4. Update de verdeling D_m: $D_{m+1(it)} = \begin{cases} \frac{D_m(it)}{Z_m}, & \text{if} \quad h_m(x_{it}) = y_{it} \\ \frac{D_m(it)\alpha_t}{Z_m}, & \text{if} \quad h_m(x_{it}) \neq y_{it} \end{cases}$, waarbij Z_t een normalisatie factor is. Deze wordt gekozen zodat D_{t+1} een verdeling is (optelt tot één).

output: $\hat{y_{it}} = \underset{y \in Y}{\text{argmax}} \sum_{m=1}^{M} log(\alpha_m) I[h_m(x_{it}) = y_{it}]$

Algorithm 1: AdaBoost.M1

AdaBoost creëert een opeenvolging van M weak-learners. Vaak wordt als weak-learner een beslissingsboom gebruikt met slechts 1 splitsing, ook wel beslissingsstomp genoemd (Freund e.a., 1999). Evenzo wordt in dit onderzoek als weak-learner een beslissingsstomp gebruikt. Het algoritme heeft als input de training dataset met observaties (x_{it}, y_{it}), $i = 1, ..., n$, $t = 1, ..., T$. Er wordt een verdeling, of set, van gewichten bijgehouden waarbij het gewicht van de verdeling voor huishouden i op tijdstip t in de m^{de} beslissingsstomp wordt aangeduid door $D_{m(it)}$. Als initialisatie wordt gebruikt dat $D_{1(it)} = \frac{1}{n \text{ x } T}$, voor alle $i = 1, ..., n$ en $t = 1, ..., T$. In iteratie m wordt een beslissingsstomp gecreëerd door te samplen van de verdeling D_m. Bijvoorbeeld, de sample die gebruikt wordt om de

$m+1^{de}$ beslissingsstomp te trainen wordt verkregen door N keer een willekeurig getal tussen 0 en 1 te trekken. Wanneer het getal tussen 0 en $D_{m(11)}$ ligt wordt x_{11} in de sample gestopt, wanneer het getal tussen $D_{m(11)}$ en $D_{m(12)}$ ligt wordt x_{12} in de sample gestopt, enzovoort. Vervolgens worden de error, ϵ_m, en 'het belang', α_m, van de beslissingsstomp berekend, door als input de training dataset te gebruiken. Deze zijn weergegeven in stap 2 en 3 in Algoritme 1 respectievelijk. Het is duidelijk dat het belang groter is wanneer de error klein is. Daaropvolgend worden de gewichten aangepast zoals is weergegeven in stap 4 in Algoritme 1. Wanneer een consumentenkeuze verkeerd is geclassificeerd wordt het gewicht verhoogd, terwijl de gewichten die behoren tot correct geclassificeerde consumentenkeuze worden verlaagd. Door de manier van samplen hebben observaties met een hoger gewicht meer kans om in de sample te komen, waardoor de classificatieprocedure wordt gedwongen zich te focussen op de observaties die moeilijk te classificeren zijn. Belangrijk is dat de gewichten na het samplen terug op $\frac{1}{n \text{ x } T}$ worden gezet, omdat de gewichten al zijn meegenomen via de sample procedure. De uiteindelijke classificatie is als volgt bepaald: $\hat{y_{it}} = \underset{y \in Y}{\text{argmax}} \sum_{m=1}^{M} log(\alpha_m) I[h_m(x_{it}) = y_{it}]$. Voor de uiteindelijke classificatie worden dus alle beslissingsbomen gecombineerd, waarbij bomen met een groter belang meer invloed hebben. Het aantal weak-learners wordt wederom gekozen met behulp van cross-validation.

Een weak-learner is een leeralgoritme dat net iets beter presteert dan een willekeurige gok. Een nadeel van AdaBoostM1 is dat het slecht presteert wanneer de weak-learner een error groter dan $1/2$ geeft (Freund, Schapire e.a., 1996). Voor een binair classificatie probleem is dit geen probleem aangezien de verwachte error rate van een willekeurige gok $1/2$ is. Echter, dit is wel een probleem in een multi-class classificatie probleem, waar de error van een willekeurige gok $(J-1)/J$ is. Hierdoor is het lastiger om aan deze eis te voldoen.

4.4 Voorspellen

Voor het bepalen van de voorspelnauwkeurigheid van de methoden wordt het percentage correcte voorspellingen bepaald van een *out-of-sample* voorspelling, waarvoor de testperiode wordt gebruikt. Ga er van uit dat $\hat{\pi}_{ijt}$ de voorspelde kans is dat huishouden i op tijdstip t product j kiest. De voorspelling van de discrete keuze dat huishouden i op tijdstip t product j kiest wordt als volgt bepaald:

$$\hat{y}_{it} = j \quad if \quad \hat{\pi}_{ijt} = max(\hat{\pi}_{i1t}, ..., \hat{\pi}_{iJt}), \tag{17}$$

5 Resultaten

In deze sectie worden de resultaten weergegeven van de toegepaste methoden die zijn besproken in de methodologie. Eerst worden in secties 5.1, 5.2 en 5.3 de resultaten van het trainen van MNL, het FNN en de tree-based methoden respectievelijk weergegeven. Vervolgens bespreekt Sectie 5.4 de voorspelresultaten van de methoden.

5.1 Trainen MNL

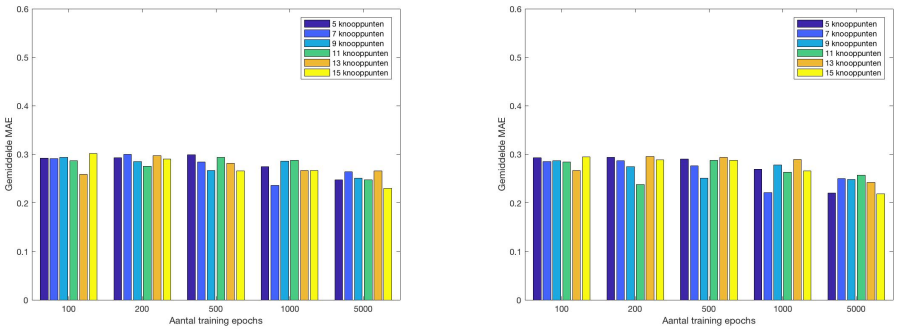
MNL wordt gebruikt als benchmark model om te bepalen welke variabelen meegenomen worden. Hiervoor is de trainingsperiode gebruikt. Het model is eerst geschat met uitsluitend de marketingmix variabelen price, feature en display. Vervolgens is het model geschat met een extra variabele *feature x display*. Dit is een binaire *cross*-variabele die 1 is als het product zowel in display als in feature is en 0 in de andere gevallen. Het derde model dat is geschat bevat de marketingmix variabelen en de loyaliteit variabele zoals gedefinieerd in Sectie 4.1.1, waarbij δ is bepaald met cross-validation. Uit het cross-validation experiment resulteren $\delta_{Ketchup}$=0,3 en $\delta_{Cracker}$=0,4. De uitkomsten van cross-validation en de parameter schattingen zijn weergegeven in appendix C en D respectievelijk. Voor beide datasets geldt dat de cross-variabele niet significant is. Om deze reden wordt deze variabele niet meegenomen in het model. Het toevoegen van de loyaliteit variabele zorgt echter voor een aanzienlijke verbetering. De likelihood waarden stijgen van -2041 naar -368 voor dataset 1 en van -2038 naar -209 voor dataset 2. In sommige gevallen is display niet significant. Echter, omdat dit één van de drie marketingmix variabelen is blijft deze in het model. Voor het vergelijken van de machine learning methoden worden twee modellen gebruikt:

Model 1: marketing mix variabelen (display, feature, price)

Model 2: marketing mix variabelen en loyaliteit variabele

5.2 Trainen FNN

Zoals vermeld in Sectie 4.2.2 moet een keuze worden gemaakt over het aantal knopen in de hidden layer en het maximum aantal training epochs. Er is 10-fold cross validatie toegepast op het FNN met 5, 7, 9, 11 en 15 knopen in de hidden layer, gecombineerd met 100, 200, 500, 1000 en 5000 training epochs. In figuur 3a en 3b zijn de resultaten van dit experiment weergegeven, toegepast op model 1 voor dataset 1 en dataset 2 respectievelijk.



(a) Dataset 1

(b) Dataset 2

Figuur 3: MAE voor 10-fold cross validation experiment

In figuur 3 is te zien dat voor dataset 1 de combinatie van vijftien knooppunten met vijfduizend training epochs optimaal is. Voor dataset 2 is de optimale combinatie zeven knooppunten gecombineerd met duizend training epochs. Voor model 2 zijn de uitkomsten van k-fold te vinden in appendix E. Verder geldt dat voor het trainen van het FNN een learning rate, η, van 0,001 is gebruikt. Wanneer de learning rate te groot is heeft het FNN de neiging voor alle huishoudens de meerderheid categorie te voorspellen.

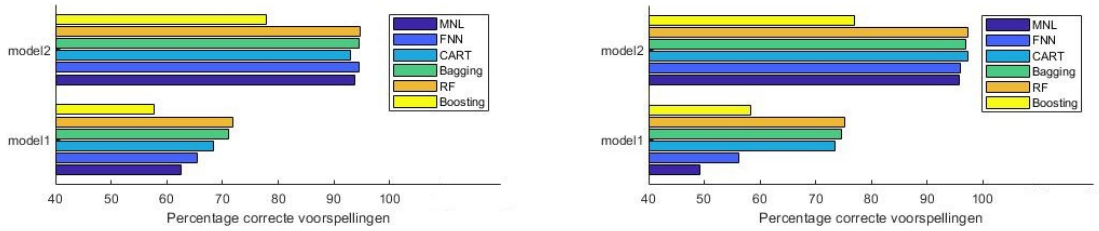
5.3 Trainen tree-based methoden

Voor de tree-based methoden worden één of meer beslissingsbomen gecreëerd. Voor Bagging, Boosting en RF is met behulp van cross-validation bepaald hoeveel beslissingsbomen er getraind worden. Voor alle modellen geldt dat er vijftig of honderd beslissingsbomen zijn gebruikt. Deze resultaten zijn weergegeven in Appendix F.

5.4 Voorspelresultaten

De voorspelnauwkeurigheid is bepaald met behulp van het percentage correcte voorspellingen. De uitkomsten hiervan zijn weergegeven in figuur 4. Voor alle classificatie technieken geldt dat het model met toevoeging van de loyaliteit variabele veel beter voorspelt dat het model met uitsluitend de marketingmix variabelen. Voor model 2 is te zien dat Boosting relatief minder goed voorspelt. Echter, voor de overige methoden liggen de percentages zo dicht bij elkaar dat hier geen duidelijke conclusie uit kan worden getrokken. Wanneer wordt gekeken naar model 1 zijn de verschillen in voorspelnauwkeurigheid duidelijker zichtbaar. RF geeft voor beide datasets de beste voorspelre-

sultaten, gevolgd door Bagging. Deze resultaten zijn overeenstemmend met die van Hagenauer en Helbich (2017). Hoewel de voorspelnauwkeurigheid van MNL voor model 2 dichtbij die van RF ligt, is dit voor model 1 niet het geval. De methode die het slechtst presteert is Boosting, gevolgd door MNL. Dit komt overeen met de verwachtingen; MNL is minder geschikt voor niet-lineaire classificatie problemen. Voor het Boosting algoritme geldt dat de error van de beslissingsbomen in sommige gevallen groter is dan 1/2, en zoals vermeld in Sectie 4.3.4 presteert AdaBoost.M1 in dit geval slecht. CART presteert in de meeste gevallen minder goed dan de ensemble methoden Bagging en RF, wat in overeenstemming is met de resultaten van Van Wezel en Potharst (2007). Echter, opvallend is dat het verschil tussen CART en de ensemble methoden niet groot is, wat betekent dat voor de gebruikte datasets in dit onderzoek overfitting geen groot probleem vormt. Het FNN presteert altijd beter dan MNL, wat overeenkomt met het onderzoek van Kumar e.a. (1995). Echter, het voorspelt slechter dan CART, Bagging en RF. Een mogelijke verklaring hiervoor is dat de datasets redelijk klein zijn, waardoor ANN's minder goed presteren.



(a) Dataset 1

(b) Dataset 2

Figuur 4: Percentage correcte voorspellingen per classifier

Zoals vermeld in Sectie 3 is de verdeling van de productkeuze voor beide datasets relatief scheef. Voor beide datasets zijn er twee producten die frequenter voorkomen dan de andere twee producten in de consumentenkeuze. Scheve datasets kunnen leiden tot modellen met een voorkeur voor de meerderheid categorieën. Om inzicht te krijgen hoe de verschillende methoden hier mee omgaan, is naast het percentage correcte voorspellingen tevens bepaald hoeveel procent correct is voorspeld per productcategorie. Het percentage correcte voorspellingen voor product j is berekend door het aantal correcte voorspellingen van product j te delen door het werkelijke aantal dat product j is gekozen door de consument. Deze resultaten zijn weergegeven in Tabel 1. Voor model 1 geldt dat Boosting de scheef verdeelde data het slechtst kan modelleren, gevolgd door MNL en FNN. Deze methoden hebben hun correcte voorspellingen met name te danken aan het correct voorspellen van

de meerderheid categorieën. CART, Bagging en RF kunnen deze data beter modelleren maar er is nog steeds een lichte voorkeur te ontdekken voor de meerderheid categorieën. Voor model 2 bevat Boosting wederom enkel correcte voorspellingen voor de meerderheid categorieën. Voor de overige methoden geldt dat deze voor model 2 geen moeite hebben de scheve data te modelleren. Er is geen voorkeur meer te ontdekken voor de meerderheid categorieën.

Tabel 1: Percentage correcte voorspellingen per product

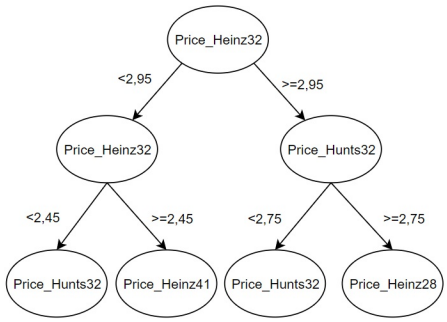
	Model 1						Model 2					
Dataset 1	MNL	FNN	CART	Bag	RF	Boost	MNL	FNN	CART	Bag	RF	Boost
Heinz41	3,23	0,00	19,35	32,26	16,13	0,00	96,77	96,77	96,77	96,77	96,77	0,00
Heinz32	80,29	85,77	81,75	83,94	87,59	79,93	96,72	98,91	95,26	96,35	96,35	98,91
Heinz28	58,38	51,78	62,94	63,96	62,94	55,33	92,39	91,37	91,37	91,88	94,92	86,80
hunts32	28,79	43,94	51,52	53,03	53,03	0,00	84,85	84,85	86,36	86,36	86,36	0,00
Dataset 2												
Sunshine	0,00	0,00	13,51	20,27	16,22	0,00	90,54	90,54	93,24	93,24	91,89	0,00
Kleebler	0,00	0,00	21,05	34,21	23,68	0,00	86,84	89,47	89,47	89,47	89,47	0,00
Nabisco	88,32	96,91	91,07	88,66	93,13	87,29	98,97	99,31	99,66	98,63	99,66	93,81
Private	17,65	22,99	80,75	82,89	81,28	48,13	94,65	94,12	97,33	97,33	96,79	96,79

5.4.1 Interpreteerbaarheid

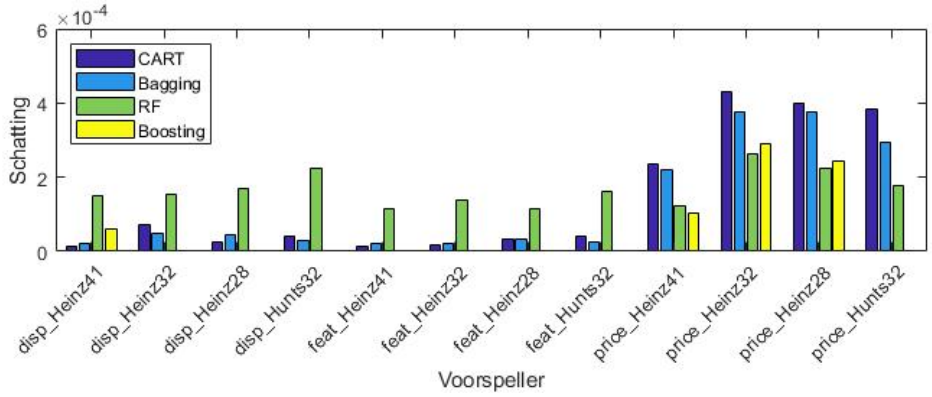
Naast het goed voorspellen van de consumentenkeuze is het belangrijk dat de resultaten interpreteerbaar zijn. Een model met goede interpretatie kan worden gebruikt door bedrijven om te beoordelen hoe manipulaties van marketingvariabelen het marktaandeel zullen beïnvloeden. Zoals vermeld in Sectie 4.1.3 kan MNL goed worden geïnterpreteerd door middel van odds ratios. Met behulp hiervan kan bijvoorbeeld worden berekend wat het effect is van een prijsverhoging op de kans dat product j wordt gekozen vergeleken met product l. Het FNN daarentegen is niet goed interpreteerbaar. ANN's staan bekend om hun "black-box" probleem (Bentz & Merunka, 2000). Het is niet bekend hoe en waarom het netwerk een bepaalde output geeft. Dit zorgt er voor dat er niet kan worden uitgelegd waarom het ene product beter verkoopt dan het andere product.

Het CART algoritme is goed en eenvoudig interpreteerbaar. Doordat de splitsingen worden bepaald aan de hand van de laagste onzuiverheid is de interpretatie direct uit de beslissingsboom af te lezen. In figuur 5 is voor model 1 van de Ketchup dataset een begin van de beslissingsboom weergegeven. Hierin is te zien dat de prijs variabele van Heinz32 de laagste onzuiverheid heeft.

Daarnaast kan de invloed per variabele worden geschat. De invloed van variabele k wordt geschat door de gewogen daling in de onzuiverheid na een splitsing op te tellen voor alle knooppunten waarin variabele k wordt gebruikt (Breiman, 2001, 2002). Ensemble methoden hebben een slechtere interpreteerbaarheid dan CART omdat de uitkomst hiervan geen beslissingsboom is. Hierdoor valt de duidelijke interpretatie van een enkele beslissingsboom weg. De invloed per variabelen kan wel worden geschat. Voor Bagging en RF is voor alle beslissingsbomen de invloed geschat op eenzelfde manier als voor het CART algoritme. Vervolgens wordt het gemiddelde genomen over de beslissingsbomen. Voor het Boosting algoritme wordt tevens voor alle beslissingsbomen de invloed op eenzelfde manier geschat, maar wordt eerst de invloed per beslissingsboom vermenigvuldigd met het bijbehorende belang van de boom, voordat het gemiddelde wordt genomen. In Figuur 6 zijn deze resultaten weergegeven voor CART, Bagging, Boosting en RF, voor model 1 van dataset 1. Er is duidelijk te zien dat voor CART, Bagging en Boosting de prijs variabelen de meeste invloed hebben op consumentenkeuze. Voor RF is de invloed van de variabelen meer gelijk verdeeld. Dit is naar de verwachtingen aangezien RF naast het willekeurig samplen over de observaties, random sampled over de onafhankelijke variabelen. De ensemble methoden zijn dus interpreteerbaar in de zin dat is te achterhalen welke variabelen de meeste invloed hebben op de consumentenkeuze. Echter, het is niet duidelijk welk effect het aanpassen van de variabelen precies heeft op de consumentenkeuze.



Figuur 5: Deel beslissingsboom



Figuur 6: Schattingen invloed van variabelen

6 Conclusie

Om als bedrijf in te kunnen spelen op consumentenkeuze is een accuraat model nodig voor het voorspellen van consumentenkeuze. Naast het nauwkeurig voorspellen is de interpreteerbaarheid van het model minstens zo belangrijk. In dit onderzoek zijn verschillende methoden vergeleken in het voorspellen van consumentenkeuze. Om te bepalen welk model het meest accuraat is wordt gekeken naar de voorspelnauwkeurigheid en de interpreteerbaarheid. Om dit te onderzoeken zijn twee datasets gebruikt. Dataset 1 bevat gegevens over de aankoopgeschiedenis van de productcategorie ketchup en dataset 2 over de aankoopgeschiedenis van de productcategorie cracker. Voor beide datasets geldt dat de verdeling van de consumentenkeuze over de producten scheef is. Er is tevens onderzocht hoe de technieken hiermee omgaan om hier meer inzicht in te krijgen.

De methoden die in dit onderzoek zijn vergeleken zijn MNL, FNN, CART, Bagging, RF en Boosting. Deze laatste drie worden ensemble methoden genoemd. RF overtreft de andere technieken in de voorspelnauwkeurigheid, gevolgd door Bagging. De techniek die het slechtst presteert op gebied van voorspelnauwkeurigheid is Boosting, gevolgd door MNL. Wanneer wordt gekeken hoe de technieken omgaan met de scheve data worden deze resultaten versterkt. Boosting is slecht in het voorspellen van de scheve data en heeft uitsluitend correcte voorspellingen voor de meerderheid categorieën. Ook MNL en FNN zijn niet in alle gevallen goed in het modelleren van de scheef verdeelde data. CART, Bagging en RF hebben daarentegen minder moeite deze data te modelleren. Op het gebied van interpreteerbaarheid presteert FNN het slechtst omdat niet bekend is hoe en waarom bepaalde uitkomsten resulteren. De methoden met de beste interpreteerbaarheid zijn MNL en CART. MNL is goed interpreteerbaar dankzij odds ratios en CART is direct interpreteerbaar doordat het resulteert in een beslissingsboom. De ensemble methoden zijn redelijk interpreteerbaar doordat bepaald kan worden welke variabelen de meeste invloed hebben op de consumentenkeuze. Echter, ze zijn slechter interpreteerbaar dan CART omdat ze niet in een enkele beslissingsboom resulteren. In tabel 2 is een overzicht weergegeven van de relatieve prestatie van de zes technieken op het gebied van voorspellen, interpretatie en het modelleren van scheve data.

De technieken die beter niet gebruikt kunnen worden voor het voorspellen van consumentenkeuze zijn Boosting en FNN. Voor boosting komt dit met name doordat de voorspelnauwkeurigheid slecht is en voor FNN omdat de interpretatie slecht is. Er zijn twee technieken die goed gebruikt kunnen

worden voor het modelleren van consumentenkeuze. Als eerste is RF goed bruikbaar dankzij de goede voorspelnauwkeurigheid. Daarbij is RF goed in het modelleren van scheve datasets en is er een redelijke interpreteerbaarheid in de zin dat het kan bepalen welke onafhankelijke variabelen de meeste invloed hebben op de consumentenkeuze. Een andere techniek die voor deze datasets goed gebruikt kan worden is CART. CART voorspelt iets minder nauwkeurig dan RF en Bagging, maar de goede interpretatie compenseert dit nadeel. Echter, uit de literatuur blijkt dat voor veel datasets CART de neiging heeft de data te overfitten, waardoor deze conclusie uitsluitend kan worden getrokken voor de datasets gebruikt in dit onderzoek.

Tabel 2: Relatieve prestatie

	MNL	FNN	CART	Bagging	Boosting	RF
Voorspelnauwkeurigheid	+/-	+/-	+/-	+	-	+
Interpreteerbaarheid	+	-	+	+/-	+/-	+/-
Modelleren van scheve data	+/-	+/-	+	+	-	+

\+ : relatief goede prestatie, +/- : relatief redelijke prestatie, - : relatief slechte prestatie

6.1 Discussie

Dit onderzoek heeft een aantal beperkingen. Om deze reden zou verder onderzoek nuttig zijn. Ten eerste is het interessant om dit onderzoek toe te passen op een grotere, complexere dataset om te onderzoeken of dit dezelfde resultaten oplevert. Ten tweede zijn dankzij tijdlimitaties niet alle bestaande machine learning technieken onderzocht. Er zijn nog een aantal machine learning technieken die toegepast kunnen worden, zoals Support Vector Machine (SVM) en Naive Bayes. Wellicht presteren deze technieken beter. Ten derde kunnen er andere Boosting algoritmen worden gebruik die mogelijk beter presteren dan het algoritme dat in dit onderzoek is gebruikt, AdaBoostM1. Het AdaBoostM2 algoritme blijkt bijvoorbeeld minder problemen te hebben met errors groter dan 1/2 (Freund, Schapire e.a., 1996). Als laatste kan er meer inzicht worden verkregen door te onderzoeken waarom bepaalde methoden goed of slecht presteren in het voorspellen van discrete consumentenkeuze.

Referenties

Agrawal, D. & Schorling, C. (1996). Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model. *Journal of Retailing*, *72*(4), 383–407.

Bentz, Y. & Merunka, D. (2000). Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. *Journal of Forecasting*, *19*(3), 177–200.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, *1*.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). Classification and regression trees. Wadsworth Int. *Group*, *37*(15), 237–251.

Franses, P. H. & Paap, R. (2001). *Quantitative models in marketing research*. Cambridge University Press.

Freund, Y., Schapire, R. E. e.a. (1996). Experiments with a new boosting algorithm. In *icml* (Deel 96, pp. 148–156). Citeseer.

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119–139.

Freund, Y., Schapire, R. & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, *14*(771-780), 1612.

Guadagni, P. M. & Little, J. D. (1983). A logit model of brand choice calibrated on scanner data. *Marketing science*, *2*(3), 203–238.

Hagenauer, J. & Helbich, M. (2017). A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications*, *78*, 273–282.

Jain, D. C., Vilcassim, N. J. & Chintagunta, P. K. (1994). A random-coefficients logit brand-choice model applied to panel data. *Journal of Business & Economic Statistics*, *12*(3), 317–328.

Kearns, M. (1988). Learning Boolean formulae or finite automata is as hard as factoring. *Technical Report TR-14-88 Harvard University Aikem Computation Laboratory*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kumar, A., Rao, V. R. & Soni, H. (1995). An empirical comparison of neural network and logistic regression models. *Marketing Letters*, *6*(4), 251–263.

Liaw, A., Wiener, M. e.a. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18–22.

Malhotra, P., TV, V., Vig, L., Agarwal, P. & Shroff, G. (2017). TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*.

McFadden. (1973). Conditional logit analysis of qualitative choice behavior, 105–142.

Neter, J., Kutner, M. H., Nachtsheim, C. J. & Wasserman, W. (1996). *Applied linear statistical models*. Irwin Chicago.

Paap, R. & Franses, P. H. (2000). A dynamic multinomial probit model for brand choice with different long-run and short-run effects of marketing-mix variables. *Journal of Applied Econometrics*, *15*(6), 717–744.

Press, S. J. & Wilson, S. (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, *73*(364), 699–705.

Razi, M. A. & Athappilly, K. (2005). A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. *Expert Systems with Applications*, *29*(1), 65–74.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Sak, H., Senior, A., Rao, K. & Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*(2), 197–227.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shang, J., S, Yu-sand, E. L. & Goetz, A. M. (2000). Diagnosis of MRSA with neural networks and logistic regression approach. *Health Care Management Science*, *3*(4), 287.

Tam, K. Y. & Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management science*, *38*(7), 926–947.

Trevor, H., Robert, T. & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction. New York, NY: Springer.

Van Wezel, M. & Potharst, R. (2007). Improved customer choice predictions using ensemble methods. *European Journal of Operational Research*, *181*(1), 436–452.

West, P. M., Brockett, P. L. & Golden, L. L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science*, *16*(4), 370–391.

A Data karakteristieken

Tabel 3: Karakteristieken van de afhankelijke en onafhankelijke variabelen

Data		Heinz28	Heinz32	Heinz41	Hunts32		Sunshine	Kleebler	Nabisco	Private
1	% keuze	30,41	52,11	6,50	10,97	2	7,26	6,68	54,44	31.44
	Gem. prijs	4,93	3,14	3,62	3,36		0,96	1,13	1,08	0,68
	% feature [a]	6,15	8,61	2,14	3,54		10,72	8,02	29,16	6,32
	% display [b]	5,36	5,22	3,15	3,65		1,61	1,64	3,80	1,15
	% feat & disp[c]	1,50	1,32	0,11	0,92		2,16	2,61	4,86	3,55

[a] Percentage van aankopen wanneer het merk alleen in display was

[b] Percentage van aankopen wanneer het merk alleen in feature was

[c] Percentage van aankopen wanneer het merk in display en feature was.

B Wiskundig bewijs parameter identificatie

$U_{ijt} = \alpha_j + x_{ijt}\beta + \varepsilon_{ijt}$

$\pi_{ijt} = Pr[Y_{it} = j|x_{it}] = Pr[max(U_{i1t}, ..., U_{iJt}) = U_{ijt}]$

$= Pr[U_{ijt} > U_{ilt} \quad \forall l \neq j]$

$= Pr[\alpha_j + x_{ijt}\beta > \alpha_l + x_{ilt}\beta \quad \forall l \neq j]$

$= Pr[(\alpha_j - \alpha_l) + x_{ijt}\beta > x_{ilt}\beta \quad \forall l \neq j]$

Kies nu $\hat{\alpha_k} = \alpha + \Delta$, dan volgt:

$(\alpha_j - \alpha_l) + x_{ijt}\beta = (\hat{\alpha_j} - \hat{\alpha_l}) + x_{ijt}\beta$

C Cross-validation voor delta

Tabel 4: Resultaten cross-validation

δ	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
MAE dataset1	0,257	0,231	**0,185**	0,206	0,279	0,390	0,517	0,646	0,758	0,848
MAE dataset2	0,112	0,101	0,075	**0,073**	0,094	0,136	0,183	0,234	0,307	0,523

Het dikgedrukte getal geeft de laagste MAE weer

D Parameter schattingen

Tabel 5: Paramter schattingen Ketchup MNL model

Model 1:		Model 2:		Model 3	
Variabele	Parameter	Variabele	Parameter	Variabele	Parameter
Intercept		*Intercept*		*Intercept*	
Heinz41	-0,005 (0,109)	Heinz41	-0,013 (0,109)	Heinz41	-0,715 (0,284)
Heinz32	1,514* (0,076)	Heinz32	1,511* (0,076)	Heinz32	1,125* (0,164)
Heinz28	3,052* (0,132)	Heinz28	3,054* (0,132)	Heinz28	3,912* (0,345)
Variabelen		*Variabelen*		*Variabelen*	
Display	0,9534* (0,110)	Display	1,044* (0,118)	Display	0,288 (0,317)
Feature	0,825* (0,128)	Feature	0,959* (0,140)	Feature	1,250* (0,368)
Price	-1,268* (0,062)	Price	-1,270* (0,062)	Price	-1,668* (0,151)
		Display x feature	-0,730 (0,318)	Loyalty	19,177* (2,252)
Max. log-likelihood -2041,1		Max. log-likelihood -2038,5		Max. log-likelihood -368,0	

Notes:

Het getal tussen de haakjes geeft de bijbehorende standaard error weer.

* significant op 1%

Tabel 6: Parameter schattingen Cracker MNL model

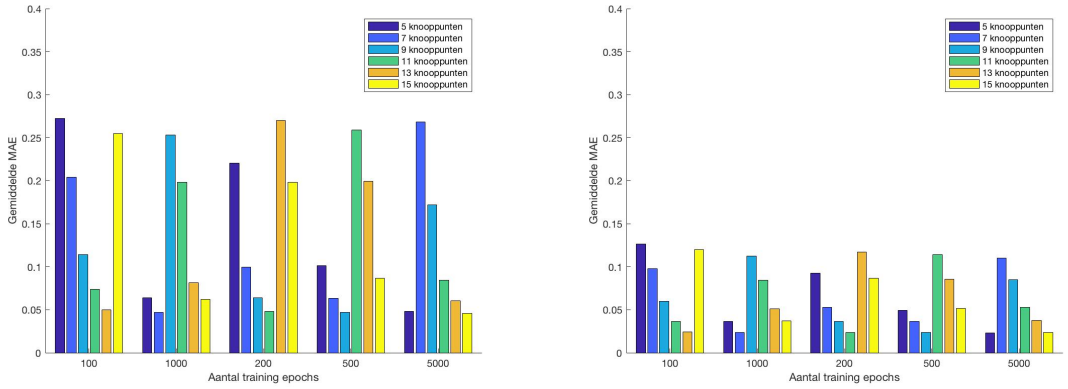
Model 1:		Model 2:		Model 3	
Variabele	Parameter	Variabele	Parameter	Variabele	Parameter
Intercept		*Intercept*		*Intercept*	
Sunshine	-0,813* (0,105)	Sunshine	-0,809* (0,105)	Sunshine	-0,042 (0,300)
Kleebler	-0,125 (0,130)	Kleebler	-0,123 (0,130)	Kleebler	0,329 (0,403)
Nabisco	1,851* (0,112)	Nabisco	1,859* (0,112)	Nabisco	1,699* (0,361)
Variabelen		*Variabelen*		*Variabelen*	
Display	0,061 (0,070)	Display	0,041 (0,074)	Display	-0,067 (0,345)
Feature	0,536* (0,106)	Feature	0,434* (0,166)	Feature	2,555* (0,470)
Price	-3,179* (0,232)	Price	-3,181* (0,233)	Price	-3,227* (0,909)
		Display x feature	0,171 (0,216)	Loyalty	16,360* (1,298)
Max. log-likelihood -2683,2		Max. log-likelihood -2682.9		Max. log-likelihood -208.9	

Notes:

Het getal tussen de haakjes geeft de bijbehorende standaard error weer.

* significant op 1%

E Resultaten 10-fold model 2



(a) Dataset 1

(b) Dataset 2

Figuur 7: 10 fold resultaten model 2

Tabel 7: Uitkomsten 10-fold cross validatie

Dataset		Model 1	Model 2
1	# Knooppunten	7	15
	# Epochs	1000	5000
2	# Knooppunten	5	11
	# Epochs	5000	200

F Cross validation aantal beslissingsbomen

Het dikgedrukte getal geeft weer welk aantal beslissingsbomen is gebruikt. Wanneer er meerdere beslissingsbomen het hoogste percentage correcte voorspellingen hebben wordt het laagste aantal gekozen in verband met een kortere rekentijd.

Tabel 8: Resultaten cross validation aantal beslissingsbomen model 1

	Percentage correcte voorspellingen cross validation model 1									
Aantal bomen	50	100	150	200	250	300	350	400	450	500
Dataset1										
Bagging	**76,68**	76,23	76,68	75,78	75,34	76,68	75,78	75,78	76,23	76,23
RF	**84,75**	83,86	83,41	82,96	83,86	82,96	83,86	83,41	82,06	83,41
Boosting	56,50	**57,40**	52,47	48,43	52,47	32,74	32,29	51,57	52,47	42,15
Dataset2										
Bagging	**83,33**	81,85	83,33	83,33	82,22	83,33	82,96	81,85	82,59	82,59
RF	81,48	**83,70**	82,59	82,59	82,59	82,59	82,59	82,22	82,96	82,96
Boosting	69,26	**72,96**	55,19	69,26	39,63	49,63	55,56	68,15	51,85	50,37

Tabel 9: Resultaten cross validation model 2 aantal beslissingsbomen

	Percentage correcte voorspellingen cross validation Model 1									
Aantal bomen	50	100	150	200	250	300	350	400	450	500
Dataset1										
Bagging	93,48	**93,91**	93,48	93,48	93,48	93,91	93,91	93,48	93,48	93,04
RF	**94,35**	94,35	94,35	94,35	94,35	94,35	93,91	93,91	94,35	94,91
Boosting	**73,48**	72,61	72,17	61,74	51,30	47,83	65,22	31,30	38,26	50,87
Dataset2										
Bagging	**97,78**	97,78	97,78	97,78	97,78	97,78	97,78	97,78	97,78	97,78
RF	**98,52**	98,52	98,52	98,52	98,52	98,52	98,15	98,52	98,52	98,52
Boosting	90,00	**91,11**	87,78	89,63	88,89	89,63	89,63	89,63	88,89	47,78