

# Regressie of Beslisboom?

Een onderzoek naar regressiemodellen en beslisbomen  
en een tool voor het maandelijks uitscoren van regressiemodellen

Stageverslag BWI  
Y.S.M. Karg  
Begeleiding: Dr. G.Jongbloed en Drs. W.B. Tip

Juni 2004

**ING**   
Retail/ Customer intelligence/  
Research & Modelling

# Regressie of Beslisboom?

Een onderzoek naar regressiemodellen en beslisbomen en een tool voor het maandelijks uitscoren van regressiemodellen

Stageverslag BWI  
Y.S.M. Karg  
Begeleiding: Dr. G.Jongbloed en Drs. W.B.Tip

Vrije Universiteit   
Faculteit der Exacte Wetenschappen  
Studierichting Bedrijfswiskunde & Informatica  
De Boelelaan 1081  
1081 HV Amsterdam

Stagebedrijf:  
ING Nederland  
Retail / Customer Intelligence / Research & Modelling  
Haarlemmerweg 520  
1014 BL Amsterdam

Juni 2004



## **Voorwoord**

Voor u ligt het verslag van mijn afstudeerstage voor de studie Bedrijfskunde & Informatica. De stage is uitgevoerd bij ING Retail op de afdeling Customer Intelligence / Research and Modelling. Ik heb onderzoek gedaan naar de logistische regressietechniek en de beslisboomtechniek. Deze technieken worden op de afdeling Customer Intelligence gebruikt om modellen te bouwen voor het selecteren van klanten voor marketingacties van de Postbank.

Verder heb ik een tool ontwikkeld die het beschikbaar maken van de resultaten van de logistische regressiemodellen verbetert.

Gedurende mijn stage heb ik een hele fijne tijd gehad en een uitstekende begeleiding. Hiervoor wil ik mijn begeleiders Dr. G. Jongbloed en Drs. W. Tip hartelijk bedanken. Zij hebben altijd tijd voor mij vrij gemaakt en zijn daardoor een grote steun geweest. Mijn dank gaat ook uit naar de medewerkers op de afdeling R&M voor hun hulp en hun bijdrage aan een prettige en gezellige sfeer.

Yonina Karg,

Amsterdam, juni 2004

## **Samenvatting**

Voor het selecteren van klanten voor marketingacties van de Postbank worden op de afdeling Customer Intelligence modellen gebouwd. Hiervoor wordt gebruik gemaakt van de technieken logistische regressie (in SAS), CHAID (in SPSS) en Model Builder (in DataDistilleries (DD)).

In eerder onderzoek bij de Postbank waarin deze technieken met elkaar werden vergeleken, deed logistische regressie het in een bepaalde situatie beter dan de beslisboomtechnieken CHAID en DD. Naar aanleiding hiervan wordt in dit onderzoek een vergelijking gemaakt tussen de logistische regressietechniek en de beslisboomtechniek. Er wordt getracht om enkele algemene richtlijnen te geven die aangeven wanneer de technieken het beste gebruikt kunnen worden.

De technieken worden onder andere in geometrische termen met elkaar vergeleken. Verder worden de technieken op een aantal punten met elkaar vergeleken, namelijk robuustheid, gevoeligheid, interpreteerbaarheid, implementeerbaarheid en de mate waarin de technieken met verschillende typen variabelen kunnen omgaan.

In dit onderzoek is er ook gekeken naar de methoden Bagging en Boosting. Deze methoden dienen voor het verbeteren van classificatietechnieken. Het idee achter deze methoden is dat classificatie niet meer gebeurt op basis van één model, maar op basis van een combinatie van meerdere modellen.

Er is onder andere gebleken dat de implementeerbaarheid van logistische regressiemodellen binnen de Postbank minder goed is dan die van de beslisbomen. Dit, omdat het een handmatig proces is, terwijl het proces in geval van beslisbomen volledig geautomatiseerd is. Het handmatige proces kan heel wat tijd in beslag nemen en brengt tevens een groot risico voor het maken van fouten met zich mee.

In het tweede deel van dit verslag wordt een tool beschreven, die in het kader van dit probleem is ontwikkeld.

# Inhoud

<b>1. INLEIDING</b> .....	<b>1</b>
1.1 PROBLEEMSTELLINGEN .....	1
1.1.1 <i>Probleemstelling I</i> .....	1
1.1.2 <i>Probleemstelling II</i> .....	1
1.2 ING RETAIL/CUSTOMER INTELLIGENCE/RESEARCH & MODELLING .....	2

## DEEL 1

<b>1. INLEIDING DEEL 1</b> .....	<b>6</b>
<b>2. CLASSIFICATIE</b> .....	<b>7</b>
<b>3. REGRESSIE</b> .....	<b>9</b>
3.1 LINEAIRE REGRESSIE .....	9
3.2 NIET-LINEAIRE REGRESSIE .....	12
3.3 LOGISTISCHE REGRESSIE .....	12
<b>4. BESLISBOMEN</b> .....	<b>17</b>
4.1 CHAID .....	19
4.2 DATADISTILLERIES .....	21
<b>5. REGRESSIE VERSUS BESLISBOOM</b> .....	<b>23</b>
5.1 VERGELIJKEN VAN CLASSIFICATIETECHNIKEN .....	24
5.1.1 <i>Juistheid van het model</i> .....	24
5.1.2 <i>Robuustheid en gevoeligheid</i> .....	27
5.1.3 <i>Interpreeteerbaarheid</i> .....	28
5.1.4 <i>Implementeerbaarheid binnen de Postbank</i> .....	28
5.1.5 <i>Typen variabelen</i> .....	28
5.2 ONDERZOEK AFDELING R&M .....	29
5.2.1 <i>Resultaten onderzoek R&amp;M</i> .....	31
5.3 DISCUSSIE .....	34
<b>6. BOOSTING EN BAGGING</b> .....	<b>36</b>
6.1 BOOSTING .....	36
6.1.1 <i>Adaboost</i> .....	37
6.2 BAGGING .....	38
6.3 BOOSTING VERSUS BAGGING .....	39
6.4 BOOSTING EN BAGGING VOOR DE POSTBANK .....	40
6.4.1 <i>Boosting</i> .....	40
6.4.2 <i>Bagging</i> .....	43
6.4.3 <i>Boosting en Bagging met SAS Enterprise Miner</i> .....	43
<b>7. CONCLUSIES EN AANBEVELINGEN</b> .....	<b>46</b>
7.1 CONCLUSIES .....	46
7.2 AANBEVELINGEN .....	47

## DEEL 2

<b>1. INLEIDING DEEL 2</b> .....	<b>52</b>
<b>2. HET HANDMATIG PROCES</b> .....	<b>53</b>
<b>3. OPLOSSINGSANALYSE</b> .....	<b>55</b>
<b>4. AANPAK PROBLEEM</b> .....	<b>57</b>
4.1. DE TOOL .....	57
4.1.1 <i>Definitie van eisen</i> .....	57
4.1.2 <i>Gebruikers en functies</i> .....	58
4.1.3 <i>GUI's</i> .....	61
4.1.4 <i>Enkele opmerkingen</i> .....	66
4.2. HET TEKSTBESTAND .....	66
4.3. HET PROGRAMMA .....	67
4.4. EISEN VAN HET MODEL .....	70

## 1. Inleiding

Aan de hand van gegevens over klanten van de Postbank worden er binnen de afdeling Customer Intelligence / Research & Modelling (CI/R&M) modellen gemaakt voor marketingacties. Het doel van deze modellen is om voor een actie klanten te vinden, waarbij de kans op responderen het grootst is. Voor het maken van de modellen maakt de afdeling onder andere gebruik van de regressietechniek en de beslisboomtechniek.

### 1.1 Probleemstellingen

Dit verslag bestaat uit twee delen:

- Het eerste deel is een onderzoek naar de regressietechniek en de beslisboomtechniek ten behoeve van het selecteren van klanten voor marketingacties.
- Het tweede deel beschrijft een tool die het eenvoudig moet maken om regressiemodellen maandelijks automatisch te scoren<sup>1</sup>.

#### 1.1.1 Probleemstelling I

Bij CI/R&M is ontdekt dat er marketingacties zijn waarbij de reacties van klanten veel beter te voorspellen zijn met een (logistisch) regressiemodel dan met een beslisboom. Het gaat hier om marketingacties, waarin het aantal respondenten erg klein is. De reden waarom een regressiemodel het in die situaties beter doet, is niet bekend. Het gevolg is dat men binnen CI/R&M voor elke marketingactie een regressiemodel en een beslisboom wil bouwen. Dat is dubbel werk. Omdat een beslisboom in de huidige situatie bij de Postbank goedkoper te produceren is, wordt er in de praktijk standaard gekozen voor een beslisboom, terwijl een regressiemodel soms beter zou kunnen zijn.

Het eerste deel is een onderzoek naar (logistische) regressiemodellen en beslisbomen. De volgende onderzoeksvraag is geformuleerd:

*In welke situaties is het beter om een regressiemodel te gebruiken in plaats van een beslisboom (of andersom)?*

#### 1.1.2 Probleemstelling II

Het opleveren van de resultaten van een regressiemodel is momenteel binnen de Postbank een “handmatig” proces dat bestaat uit meerdere stappen. Hierdoor is het productieproces afhankelijk van de aanwezigheid van de onderzoeker, die het model bouwde en is er onderzoekstijd kwijt aan productiewerk. De kans op het maken van fouten in dit handmatig proces is tevens erg groot. Deze factoren zijn onwenselijk. De opdracht van het tweede deel is als volgt:

*Produceer een tool voor het (automatisch) uitscoren van regressiemodellen.*

Om ervoor te zorgen dat de onderzoekers geen tijd kwijt zijn aan het “handmatig” opleveren van de scores, moet het productieproces onafhankelijk gemaakt worden van de aanwezigheid

---

<sup>1</sup> Uitscoren: de modellen worden gedraaid en de resultaten komen op de juiste plaats terecht voor verdere verwerking.



van de onderzoeker. Er is dus een geautomatiseerd proces nodig, waarmee de regressiemodellen van CI/R&M automatisch gescoord worden.

In dit proces moeten de volgende acties onderscheiden worden:

Eenvoudig toevoegen van bestaande regressiemodellen aan de tool.

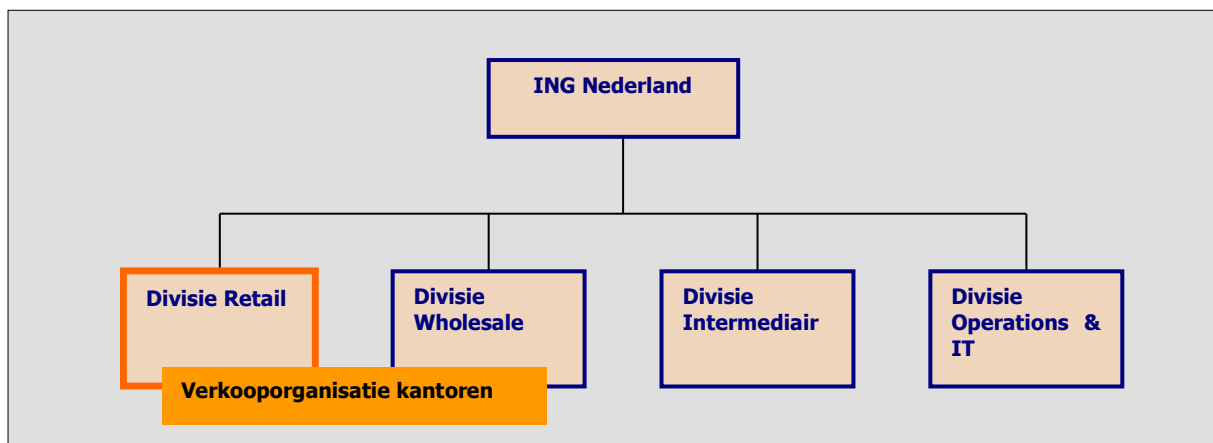
Maandelijks scores van regressiemodellen en het aanleveren van de scores voor verder gebruik

## 1.2 ING Retail/ Customer Intelligence/ Research & Modelling

Het onderzoek wordt uitgevoerd op de afdeling Research & Modelling. Deze paragraaf geeft een beeld over deze afdeling binnen de ING.

### ING Nederland

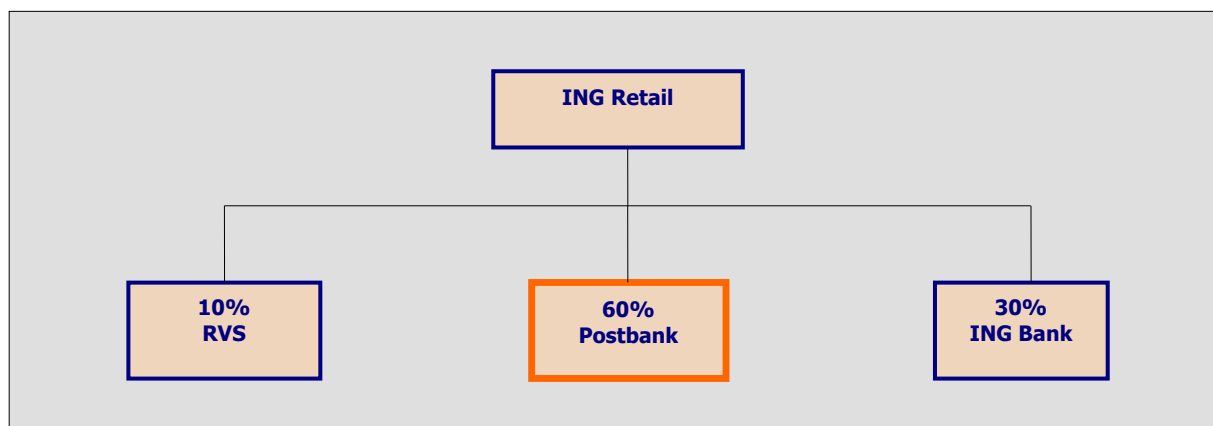
ING Nederland bestaat uit vier divisies. Het doel van iedere divisie is om de klant service, gemak, zekerheid en overzicht te bieden in zijn financiële situatie. De divisies staan weergegeven in onderstaand figuur.



Figuur I: Organogram ING Nederland

### ING Retail

De divisie Retail is opgericht in 2003 en bestaat uit de labels RVS, Postbank en ING Bank.

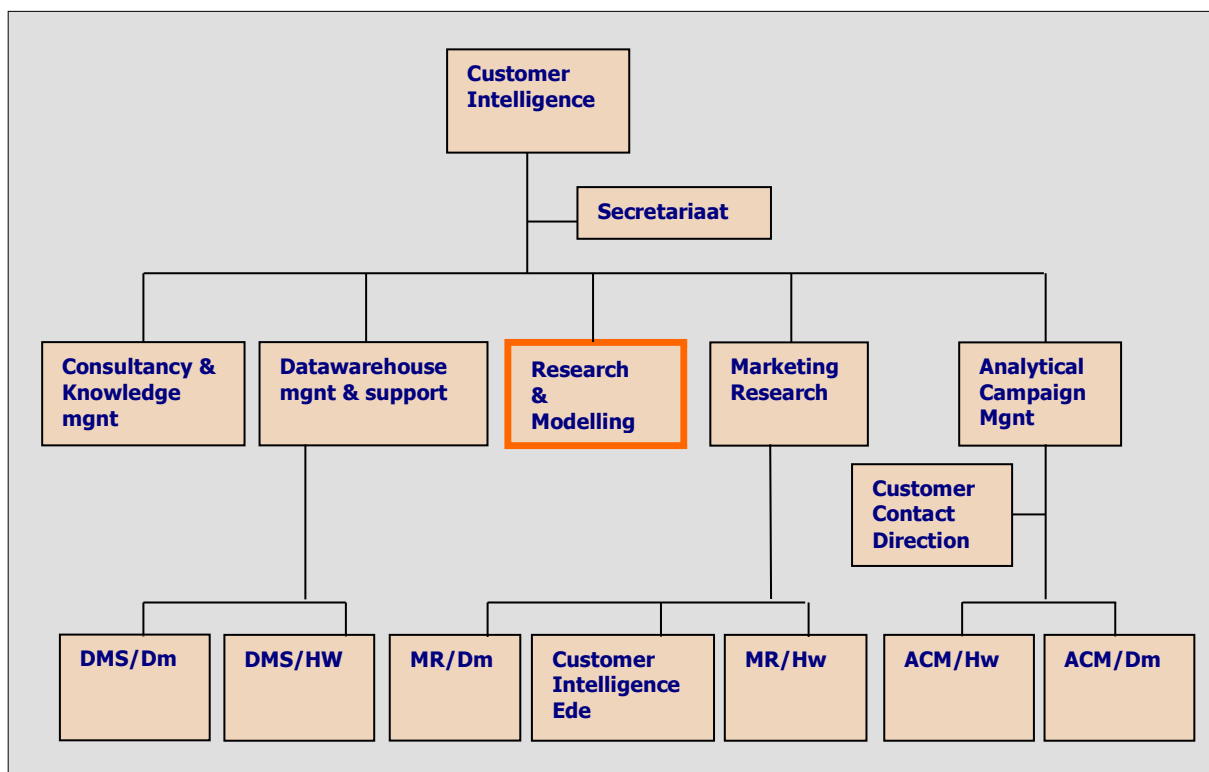


Figuur II: Organogram ING Retail

### Afdeling Customer Intelligence

Customer Intelligence (CI) is een van de onderdelen binnen ING Retail. CI is verantwoordelijk voor het voeren van de klantregie voor alle klanten van Retail, dat wil zeggen voor de labels Postbank, ING Bank en RVS.

Zij doet dit door het maken van slimme selecties voor marktwerkingen (Analytical Campaign Management), door het maken van modellen van klantgedrag of voor scoring in operationele omgevingen (Research & Modelling), in- en extern marktonderzoek (Marketing Research) en het functioneel beheren en verder ontwikkelen van de grote klantenbase van Retail (Datawarehouse Management and Support). Deze vier disciplines komen samen in het Consultancy en kennis centrum van CI (Consultancy & Knowledge Center).



Figuur III: Organogram afdeling Customer Intelligence

### Afdeling Research & Modelling

Research & Modelling staat voor hoogwaardige data-analyse, optimale modellen en state-of-the-art advies over methoden en technieken voor onderzoek, modelling, evaluatie en experiment. Deze afdeling heeft directe invloed op:

- selecties voor marketingacties (m.n. door marketingmodellen)
- acceptatie- en beheersprocessen van kredietproducten (m.n. door risicomodellen en/of scorecards)
- de wijze waarop ING Retail evalueert en experimenteert (m.n. door meetplannen, testopzetten en bijbehorende evaluaties).

Het onderzoek concentreert zich op het eerst genoemde punt; marketingmodellen voor selecties voor marketingacties.



---

# DEEL 1

Een onderzoek naar  
regressiemodellen en beslisbomen

## 1. Inleiding deel 1

Binnen de afdeling Customer Intelligence worden modellen gemaakt die dienen voor het selecteren van klanten voor marketingacties. Postbank gebruikt propensity- en mailmodellen voor het selecteren van de klanten. Een *propensity-model* voorspelt de kans op productafname in een bepaalde periode. Het model wordt gebouwd op basis van data over productafnames in een groep van klanten in het verleden. Een *mailmodel* is een model dat de kans voorspelt dat een klant respondeert op een mailing. Het model wordt gebouwd op basis van data over het reageren op een vergelijkbare mailing in het verleden, bij voorkeur in een random steekproef van klanten die de betreffende of soortgelijke mailing hebben ontvangen. Bij dit soort modellen is de afhankelijke variabele meestal dichotoom van aard. De afhankelijke variabele kan dus maar twee waarden aannemen; wel of niet reageren op een mailing, wel of geen aanschaf van het product in een bepaalde periode.

Voor het bouwen van de mail- en propensity-modellen wordt er bij de Postbank gebruik gemaakt van de technieken CHAID (in SPSS), logistische regressie (in SAS) en Model Builder (in DataDistilleries). Model Builder van DataDistilleries is een nieuwe tool binnen de Postbank (sinds 2000) en maakt gebruik van een voor de Postbank nieuwe modelleertechniek, namelijk decision lists. De tool van DataDistilleries is geautomatiseerd en CHAID en logistische regressie zijn interactieve modelbouwtechnieken. Uit een onderzoek van de Postbank waarin deze drie technieken met elkaar worden vergeleken [Westerlaken (2003)], deed logistische regressie het in een bepaalde situatie beter dan de beslisboomtechnieken CHAID en DataDistilleries (zie hoofdstuk 5). In het onderzoek zijn drie verschillende datasets geanalyseerd; twee van de datasets bestaan uit een grote doelgroep en voldoende respondenten, en één dataset bestaat uit een redelijk grote doelgroep en weinig respondenten. De drie technieken werden toegepast op elk van de drie datasets. Er werden in totaal dus negen modellen gebouwd. In het onderzoek deed logistische regressie het beter bij de marketingacties, waarin er weinig respondenten voorhanden zijn.

Dit deel van het verslag bevat de resultaten van een onderzoek naar (logistische) regressiemodellen en beslisbomen. Er wordt aangegeven in welke situaties de technieken het beste gebruikt kunnen worden. Tevens wordt er een verklaring voor gegeven. Allereerst wordt in hoofdstuk 2 het begrip classificatie uitgelegd. De technieken ‘regressie’ en ‘beslisboom’ worden in respectievelijk hoofdstuk 3 en 4 besproken om ze vervolgens in hoofdstuk 5 met elkaar te vergelijken. De vergelijking van de technieken gebeurt onder andere in geometrische termen. Ook worden er verschillende methoden behandeld voor het meten van de kwaliteit van modellen. Op basis hiervan kunnen de technieken met elkaar vergeleken worden. Verder wordt er in hoofdstuk 5 gekeken naar de resultaten van het onderzoek van de Postbank. In hoofdstuk 6 worden twee vrij nieuwe onderwerpen geïntroduceerd, namelijk Boosting en Bagging. Dit zijn methoden die dienen voor het verbeteren van classificatietechnieken. Uiteindelijk wordt het verslag beëindigd in hoofdstuk 7 met een conclusie.

## 2. Classificatie

Het doel van classificatietechnieken is om te voorspellen tot welke groep (klasse) een bepaald object behoort. De objecten, ook wel instanties genoemd, worden beschreven door een verzameling variabelen. Een *object* is bijvoorbeeld een klant. De klant wordt beschreven door een aantal kenmerken, zoals bijvoorbeeld leeftijd, geslacht, saldo betaalrekening. Stel dat men op basis van deze kenmerken wil voorspellen of de klant gaat reageren op een bepaalde mail. De kenmerken ‘Leeftijd,’ ‘Geslacht’ en ‘Saldo betaalrekening’ zijn de *onafhankelijke* oftewel *verklarende variabelen* en de variabele ‘Reageren’ is de *afhankelijke variabele* oftewel de *klasse* waar de klant toe behoort. De rijen in Tabel 2.1 stellen de objecten oftewel klanten voor en de kolommen zijn de variabelen.

Leeftijd	Geslacht	Saldo betaalrekening	Reageren
21	M	550	Ja
28	V	300	Nee
27	M	210	Nee

Tabel 2.1

Bij classificatie wordt er onderscheid gemaakt tussen ‘*unsupervised learning*’ en ‘*supervised learning*’. In het eerste geval zijn de klassen voor de dataset niet bekend. Het is de bedoeling om na te gaan of er homogene klassen van objecten bestaan binnen een gegeven dataset. Dit wordt ook wel *cluster analyse* genoemd.

In het tweede geval zijn de klassen voor de dataset wel bekend en moeten er bepaalde classificatieregels afgeleid worden van steekproeven die objecten bevatten waarvan de klassen bekend zijn. Vervolgens worden deze regels, die betrekking hebben op de onafhankelijke variabelen, toegepast op nieuwe objecten waarvan de klassen niet bekend zijn. Door middel van de regels kunnen de klassen van deze nieuwe objecten voorspeld worden. Aangezien mijn onderzoek te maken heeft met het selecteren van klanten en de klassen in de data van tevoren bekend zijn, namelijk *wel* of *niet* reageren op een mail (2 klassen), zal ik mij in dit verslag alleen bezig houden met ‘supervised learning’. In het vervolg van dit verslag zal met classificatie dit type bedoeld worden.

### Classificatie in geometrische termen

In geometrische termen kan classificatie als volgt uitgelegd worden: in een multi-dimensionale ruimte stelt ieder object een vector of een punt voor (per verklarende variabele een coördinaat). Binnen de ruimte worden gebieden vastgesteld die elk met een klasse corresponderen. Nieuwe objecten, waarvoor de klassen nog niet bekend zijn, kunnen dan geclassificeerd worden op basis van het gebied waarin ze vallen.<sup>2</sup>

### Training en Test set

Voor het maken van een model is er data nodig met objecten waarvan de klassen bekend zijn. Deze dataset wordt gesplitst in twee datasets, namelijk de training set en de test set. De verhouding is meestal 75% voor de training set en 25% voor de test set. Bij de Postbank is deze verhouding respectievelijk 66% en 33%. De *training set* wordt gebruikt voor het bouwen

<sup>2</sup> Bij de Postbank gaat het om het schatten van de kans dat een klant reageert op een bepaalde mailing. In dit geval worden er twee gebieden vastgesteld: één gebied correspondeert met “Reageren” en het andere gebied correspondeert met “Niet reageren”. De vaststelling van de gebieden hangt in dit geval ook af van het aantal mailings dat gedaan moet worden.

van het model. In de training set zijn zowel de waarden van de verklarende variabelen bekend als de waarde van de klassen van de verschillende objecten. De bedoeling is om classificatieregels te vinden, zodat nieuwe objecten geïdentificeerd kunnen worden. Vervolgens kan het model getest worden op de *test set*; door middel van de classificatieregels worden de objecten van de test set geïdentificeerd. Dit resultaat kan dan vergeleken worden met de werkelijke klassen behorende bij de objecten van de test set. De test set wordt dus gebruikt om de algemeenheid van de classificatieregels te testen; gelden de classificatieregels ook voor nieuwe data? Op deze manier kan de kwaliteit van het model bepaald worden. Er zijn verschillende maten waarin de kwaliteit van het model uitgedrukt kan worden. Het eenvoudigste criterium is het percentage verkeerd geïdentificeerde objecten. Dit is het percentage objecten waarvan de voorspelde en de geobserveerde waarde van de afhankelijke variabele niet overeenkomen. Dit criterium staat bekend als de “*error rate*”. Niet iedere fout hoeft even zwaar te zijn. Het is bijvoorbeeld erger om geen mailing te sturen naar een klant die wel zou reageren dan om een mailing te sturen naar iemand die niet reageert. Bij de Postbank wordt daarom de *lift curve* gebruikt om de kwaliteit van het model te bepalen. In hoofdstuk 5 worden er verschillende criteria behandeld om de kwaliteit van een model te bepalen.

### Overfitting

*Overfitting* vindt plaats wanneer de verkregen classificatieregels zó gespecificeerd zijn dat die regels alleen gelden voor de objecten waarop ze getraind zijn. De regels zijn dan niet algemeen geldig. Een te kleine dataset zou een oorzaak voor overfitting kunnen zijn.

De test set kan gebruikt worden om dit effect te voorkomen.

### Classificatietechnieken bij de Postbank

In de literatuur zijn er verschillende classificatietechnieken te vinden. Deze technieken worden gebruikt om classificatieregels te bepalen; een model te bouwen. Voor het bouwen van mailmodellen en propensity-modellen maakt de Postbank gebruik van de regressietechniek en de beslisboomtechniek. Deze technieken worden in respectievelijk hoofdstuk 3 en 4 behandeld.

Bij de mailmodellen en propensity-modellen gaat het meestal om een afhankelijke variabele die twee waarden kan aannemen; wel of niet reageren op een mailing. Een klant wordt toegewezen aan een van deze twee klassen. De modellen bepalen de kans dat een klant tot een bepaalde klasse behoort. De kans dat een klant reageert op een mailing wordt de *responskans* genoemd. Het is de bedoeling om klanten te identificeren met de hoogste responskansen. De mailing wordt uiteindelijk verdeeld over die klanten die de hoogste responskansen hebben.

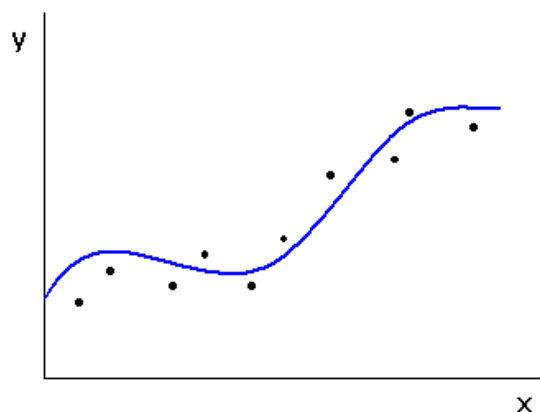
### 3. Regressie

*Regressie* is een statistische methode om te onderzoeken hoe een responsvariabele afhangt van een of meerdere verklarende variabelen. De variabele die men wenst te begrijpen, te verklaren of te voorspellen noemt men de *afhankelijke variabele* of de *respons variabele*. In dit verslag concentreer ik mij op modellen met slechts één afhankelijke variabele.

De variabelen die in het model worden opgenomen omdat er vermoed wordt dat deze een 'effect' (invloed) kunnen hebben op de afhankelijke variabele, noemt men de *onafhankelijke variabele* of de *verklarende variabele*. De onderzoeker zal op grond van voorkennis, vakliteratuur, ervaring, datamining etc. bepalen welke verklarende variabelen gebruikt zullen worden voor verder onderzoek of welke variabelen uiteindelijk in het model opgenomen zullen worden.

#### Regressie in geometrische termen

Zoals in het vorige hoofdstuk al is aangegeven, kunnen observaties (objecten) gezien worden als een aantal punten of vectoren in een multi-dimensionale ruimte. Het is nu de bedoeling om de kromme te vinden die zo goed mogelijk aansluit bij de verklarende variabelen. Deze kromme hoeft niet noodzakelijk exact door de observaties te gaan. Het probleem van regressie is het vinden van de vergelijking van de kromme, ook wel het regressiemodel genoemd. Er bestaan immers heel veel krommen die goed aansluiten bij de observaties. Er is dus een bepaalde maatstaf nodig om te kunnen bepalen welke kromme het beste aansluit bij de observaties. Een voorbeeld van regressie in een twee-dimensionale ruimte wordt gegeven in Figuur 3.1.



*Figuur 3.1: Voorbeeld Regressie in twee dimensionale ruimte; één afhankelijke variabele en één verklarende variabele.*

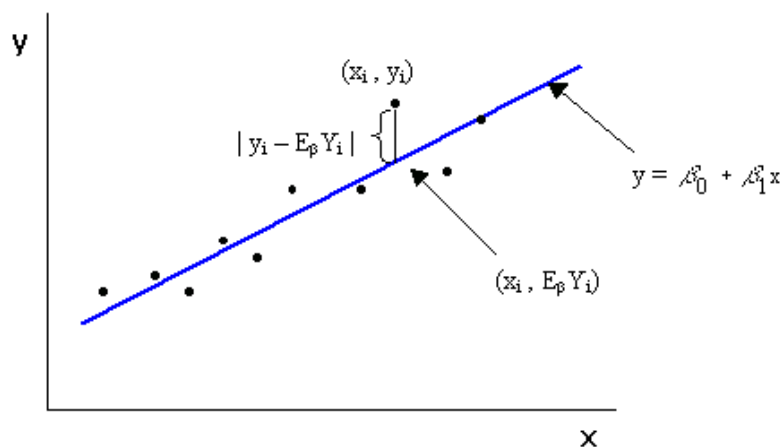
#### 3.1 Lineaire Regressie

Bij lineaire regressie bestaat het volgende verband tussen de onafhankelijke variabele(n) en de afhankelijke variabele:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_m x_{im} + \varepsilon_i \text{ voor } i = 1, 2, \dots, n, \quad (3.1)$$



met  $n$  het aantal observaties. De parameter  $\beta_0$  wordt ook wel het *intercept* genoemd en is de verwachte waarde van  $y_i$  wanneer alle onafhankelijke variabelen gelijk zijn aan nul. De parameters  $\beta_1, \dots, \beta_m$  heten de *regressiecoëfficiënten* en kunnen geïnterpreteerd worden als de verandering in de verwachte waarde van  $y_i$  als de bijbehorende onafhankelijke variabele met één eenheid stijgt én de overige onafhankelijke variabelen constant gehouden worden<sup>3</sup>. De parameter  $\varepsilon_i$  staat voor de *fout* behorende bij observatie  $i$ . De fout  $\varepsilon_i$  is de verticale afstand tussen de observatie  $y_i$  en het punt op de regressiecurve behorende bij die observatie (zie Figuur 3.2). Er wordt verondersteld dat de fouttermen onafhankelijk en normaal verdeeld zijn.



Figuur 3.2: Voorbeeld Lineaire Regressie in twee dimensionale ruimte

In geval van twee variabelen, de afhankelijke variabele en één onafhankelijke variabele wordt de regressiefunctie beschreven door een rechte lijn (zie Figuur 3.2). Hier spreekt men van een enkelvoudige lineaire regressie. Indien er meerdere variabelen aanwezig zijn, spreekt men van meervoudige lineaire regressie en wordt de regressiefunctie beschreven door een hypervlak (= de uitbreiding van het begrip ‘rechte’ voor hogere dimensies).

Sommige niet-lineaire verbanden kunnen ook gemodelleerd worden door middel van de lineaire regressie techniek. De vergelijking  $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_m x^m + \varepsilon$  bijvoorbeeld, kan omgezet worden naar het lineaire model  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m + \varepsilon$  waarbij  $x_1 = x$ ,  $x_2 = x^2$ , ...,  $x_m = x^m$ .

### Parameterschatting

In het geval van lineaire regressie zijn er in de literatuur twee methoden bekend om de waarden van de parameters  $\beta_0, \dots, \beta_m$  te bepalen, gebaseerd op het kleinste kwadratenprincipe en het principe van maximale aannemelijkheid.

#### Kleinste kwadratenprincipe

Beschouw vergelijking 3.1. Laat  $E_\beta Y_i$  de verwachte waarde van de observatie  $y_i$  zijn bij parameterkeuze  $\beta$ . De functie,

<sup>3</sup> Deze interpretatie is natuurlijker in experimentele studies dan in observationele studies.

$$S(\beta) = \sum_{i=1}^n (y_i - E_{\beta} Y_i)^2 \quad (3.2)$$

meet de som van de kwadraten van de afstanden tussen de observaties  $y_i$  en de punten van de kromme ( $E_{\beta} y_i$ ) met overeenkomstige  $x$ -coördinaten, oftewel het verschil tussen de werkelijke waarde van de observaties en de verwachte waarden wanneer  $\beta$  de echte parameter zou zijn. Een voorbeeld hiervan is weergegeven in Figuur 3.2. De parametervector  $\beta$  kan geschat worden door de vector  $\hat{\beta}$  die de functie  $S(\beta)$  minimaliseert. Deze methode staat ook wel bekend als de *kleinste kwadratenmethode*. De gevonden parametervector  $\hat{\beta}$  bepaalt de vergelijking van de beste kromme. Voor de exacte formules voor de berekening van de parameterschatters die hieruit afgeleid kunnen worden, verwijst ik naar [Berry en Feldman (1985)]. Tegenwoordig zijn er voldoende statistische software pakketten beschikbaar om deze schatters snel en eenvoudig te bepalen.

#### *Meest aannemelijke schatter*

Een andere manier om de parametervector  $\beta$  te schatten is gebaseerd op de methode van maximale aannemelijkheid. De resulterende schatter staat bekend onder de engelse term ‘Maximum likelihood estimator.’ Deze schatter wordt verkregen door de log van de aannemelijkheidsfunctie,  $L$ , te maximaliseren. De aannemelijkheidsfunctie hangt af van het onderliggende stochastisch model van de dataset. In [Breen (1996)] vindt u meer informatie over de meest aannemelijke schatter. Indien de fouttermen normaal verdeeld zijn met verwachting gelijk aan 0 en variantie gelijk aan  $\sigma^2$ , dan levert deze schatter hetzelfde resultaat op als het kleinste kwadratenprincipe.

#### **Kwaliteit van het model**

De *determinatiecoëfficiënt*,  $R^2$ , is een maat voor de kwaliteit van een regressiemodel. Deze wordt gedefinieerd als:

$$R^2 = \frac{SST - SSE}{SST} = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2 - \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (3.3)$$

waarbij

- $SST$  := de totale kwadraatsom van de afhankelijke variabele
- $SSE$  := de kwadraatsom van het verschil van de werkelijk gemeten waarden  $Y_i$  en de aan de hand van het model voorspelde waarden  $\hat{Y}_i$
- $n$  := het aantal observaties
- $\bar{Y}$  := het gemiddelde van de geobserveerde data  $Y_i$

$SST$  geeft de fout weer wanneer het gemiddelde van de observaties als voorspelling van de waarnemingen genomen wordt.  $SSE$  geeft de fout weer indien voor de voorspelling van de waarnemingen het regressiemodel gebruikt wordt.  $R^2$  geeft dus het percentage van de reductie van de fouten weer indien, in plaats van het gemiddelde van de waarnemingen, het regressiemodel gebruikt wordt. De waarde van  $R^2$  ligt steeds tussen 0 en 1. Hoe dichter  $R^2$  bij

1 ligt, hoe beter de werkelijke waarden van de afhankelijke variabele benaderd worden door het model. Als  $R^2$  gelijk is aan 0, dan wil dit zeggen dat het model geen enkele toegevoegde waarde heeft. Het model past dan even goed als het model waarin geen enkele verklarende variabele voorkomt.

### 3.2 Niet-Lineaire Regressie

De afhankelijke variabele wordt gegeven als een niet-lineaire functie van de onafhankelijke variabelen,

$$y_i = f(x_i, \beta) + \varepsilon_i \quad \text{voor } i = 1, 2, \dots, n \quad (3.4)$$

waarbij  $f$  een niet-lineaire functie is (niet-lineair in  $\beta$ ).

#### Parameterschatting

Het zoeken naar een minimum van de functie  $S(\beta)$  (3.2) is bij niet-lineaire regressie een stuk lastiger dan bij lineaire regressie. Er bestaat vaak geen expliciete uitdrukking voor de oplossing en er zal gebruik gemaakt moeten worden van een iteratief algoritme dat het minimum benadert. Twee bekende technieken die hiervoor gebruikt kunnen worden zijn de *Newton-Raphson methode* en de *Steepest-descent methode*. Deze technieken zoeken naar nulpunten van de functie (in een minimum van een gladde functie is de eerste afgeleide gelijk aan nul). Ik ga niet verder in op deze technieken. Voor een uitgebreidere uitleg verwijs ik naar [Seber (1989)].

### 3.3 Logistische Regressie

Logistische regressie analyse is geschikt voor een afhankelijke variabele die dichotoom van aard is. De afhankelijke variabele heeft in dit geval dus maar twee mogelijke waarden, zoals bijvoorbeeld ‘succes’ of ‘mislukking’. Deze waarden kunnen we gelijkstellen aan respectievelijk 1 en 0. In het geval van  $n$  onafhankelijke binaire waarnemingen is het aantal successen binomiaal( $n, p$ ) verdeeld waarbij  $p$  de succeskans is.

In het geval van logistische regressie is er informatie aanwezig over verklarende variabelen  $x$  die van invloed kunnen zijn op de succeskans. De vraag is hoe de kans op succes,  $p$ , afhangt van  $x$ .

#### Binomiale verdeling en kansverhoudingen

Het aantal successen in het geval met  $n$  onafhankelijke waarnemingen is binomiaal ( $n, p$ ) verdeeld. De schatter  $\hat{p}$  voor de succeskans wordt gegeven door de fractie successen in de steekproef.

$$\hat{p} = \frac{\# \text{ successen}}{n}$$

Logistische regressie werkt met kansverhoudingen in plaats van fracties. De kansverhouding, ook wel *odds* genoemd, is de verhouding tussen de fracties van de twee mogelijke uitkomsten. Als  $p$  de fractie bij de ene uitkomst is (bijv. succes), dan is  $1 - p$  de fractie bij de andere uitkomst (bijv. mislukking). De ODDS wordt dan gegeven door:

$$ODDS = \frac{p}{1-p} \quad (3.5)$$

Bij logistische regressie willen we de succeskans  $p$  uitdrukken in termen van de verklarende variabelen. De logaritme van de kansverhouding ( $p/(1-p)$ ) wordt gemodelleerd als een lineaire functie van verklarende variabelen. Deze transformatie van de kansverhouding wordt de *log odds* of de *logit* genoemd:

$$Logit(p) = \log(p / (1-p)) \quad (3.6)$$

Het logistisch regressiemodel ziet er als volgt uit:

**Logistisch regressiemodel**

Het statistische model voor logistische regressie is

$$\begin{cases} Y_i \sim \text{Bern}(p_i) \\ p_i = p(x_i) \\ \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \dots + \beta_m x_{im} \end{cases}$$

Hierbij is  $p_i$  een binomiale fractie en  $x_{i1}, \dots, x_{im}$  zijn de verklarende variabelen behorende bij het  $i$ -de object ( $1 \leq i \leq n$ ). De parameters van het logistisch model zijn  $\beta_0, \beta_1, \dots, \beta_m$ .

Figuur 3.3: Logistisch regressiemodel

### De verklarende variabelen

Logistische regressie kan omgaan met zowel categorische als kwantitatieve verklarende variabelen. Om categorische variabelen zoals bijvoorbeeld geslacht in een regressie model te gebruiken, worden *indicator* variabelen gebruikt. Een voorbeeld van een indicatorvariabele is:

$x = 1$  als de klant een man is

$x = 0$  als de klant een vrouw is.

In een logistische regressie analyse kunnen ook onafhankelijke categorische variabelen opgenomen worden met meer dan twee categorieën. In dit geval moeten er net zoveel *dummy-variabelen* aangemaakt worden als er categorieën zijn, die alle op één na in het model opgenomen worden. Dummy variabelen nemen alleen de waarden 0 of 1 aan. Iedere dummy variabele is slechts voor één categorie gelijk aan 1. Voor de overige categorieën is deze variabele gelijk aan 0. Eén categorie wordt gecodeerd met alle dummy-variabelen gelijk aan 0.

### Voorbeeld 3.1

Stel dat een nominale onafhankelijke variabele drie categorieën heeft, namelijk *A*, *B* en *C*. De dummy variabelen zijn dan als volgt:

Categorie	Dummy 1	Dummy 2
A	0	0
B	1	0
C	0	1

Tabel 3.1: Voorbeeld Dummy variabelen

### Parameterschatting

Voor het schatten van de parameters ligt het meer voor de hand om de maximum likelihood-methode te gebruiken, aangezien deze methode in deze situatie betrouwbaardere resultaten oplevert dan de kleinste kwadratenmethode [Menard (1995)]. In de meeste gevallen bestaat er geen expliciete uitdrukking voor de meest aannemelijke schatter. Er is een iteratieve methode nodig om de schatting van de parameters te benaderen. Een bekende methode hiervoor is ‘Fisher scoring’ [McCullagh en Nelder (1989)]. Het softwarepakket SAS maakt voor het schatten van de parameters gebruik van varianten van het Newton-Raphson algoritme en het “iteratively weighted least squares” algoritme.

### Kwaliteit van het model

Voor het meten van de kwaliteit van een logistisch regressiemodel kunnen de *deviantie* (“Deviance”) en de “Pearson chisquared statistic” gebruikt worden. Deze technieken worden behandeld in [McCullagh en Nelder (1989)].

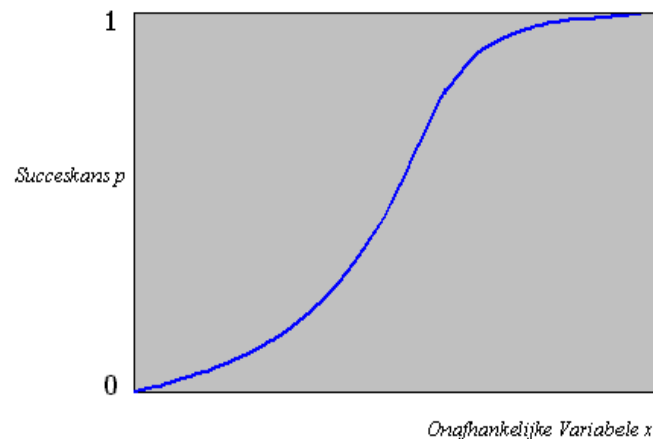
### Interpretatie van het logistisch regressiemodel

Als eenmaal de waarden voor de  $\beta$ ’s geschat zijn, kan de *ODDS ratio* bepaald worden. De ODDS ratio is een verhouding van twee ODDS, bijvoorbeeld de verhouding tussen de kansverhouding dat de man op een mailing reageert en de kansverhouding dat dit voor de vrouw geldt. In dit geval kan de ODDS ratio berekend worden door de transformatie  $e^\beta$ , waarbij  $\beta$  de parameter is behorende bij de verklarende variabele ‘geslacht’. De ODDS ratio kan geïnterpreteerd worden als de verandering in de kansverhouding als de waarde van de bijbehorende verklarende variabele met één eenheid toeneemt, en de overige verklarende variabelen constant gehouden worden. Als de ODDS ratio gelijk is aan 1, dan kan de bijbehorende variabele uit het model verwijderd worden; de twee kansverhoudingen zijn dan gelijk en de variabele heeft dus geen invloed op de voorspelling.

Het logistisch model kan omgezet worden naar een kansmodel. De fracties ‘succes’ en ‘mislukking’ kunnen als volgt berekend worden:

$$p_{succes} = \frac{e^{(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}{e^{(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)} + 1} \quad \text{en} \quad p_{mislukking} = \frac{1}{e^{(\beta_0 + \beta_1 x_1 + \beta_n x_n)} + 1} \quad (3.7)$$

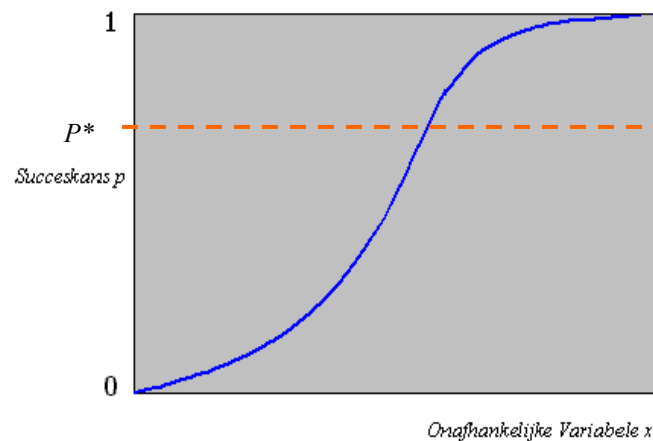
Deze formules geven de relaties weer tussen de kansen ( $p_{succes}$  en  $p_{mislukking} (= 1 - p_{succes})$ ) en de verklarende variabelen. In onderstaand Figuur 3.4 is een voorbeeld gegeven van een enkelvoudige logistische regressie. Deze logistische curve is een *S*-vormige curve. De curve beschrijft de relatie tussen de verklarende variabele en de succeskans  $p_{succes}$ .



Figuur 3.4: Voorbeeld Logistische Regressie in twee-dimensionale ruimte

### De stap naar de praktijk

Als eenmaal de vergelijking voor de succeskans ( $p_{succes}$ ) is gevonden, kan per klant de geschatte succeskans bepaald worden door het invullen van de waarden van de verklarende variabele behorende bij die klant. Vervolgens worden de succesansen geordend in afnemende grootte. Op basis van de grootte van de mailing wordt bepaald welke klanten een mailing toegestuurd krijgen. Stel dat de mailing 100.000 groot is, dan worden de eerste 100.000 klanten uit de geordende lijst geselecteerd voor een mailing. De kans dat deze klanten reageren is immers naar schatting het grootst.



Figuur 3.5: Het opsplitsen in twee groepen

Bekijk Figuur 3.5. De groep wordt in tweeën opgesplitst<sup>4</sup>: klanten met een succeskans groter dan  $p^*$  krijgen een mailing toegestuurd.

<sup>4</sup> Voor een bepaalde marketingactie wordt er meerdere keren per jaar een mailing verstuurd. De groep wordt door de modelbouwer op de afdeling CI/R&M daarom in meerdere groepen opgesplitst.

**Logistische regressie in geometrische termen**

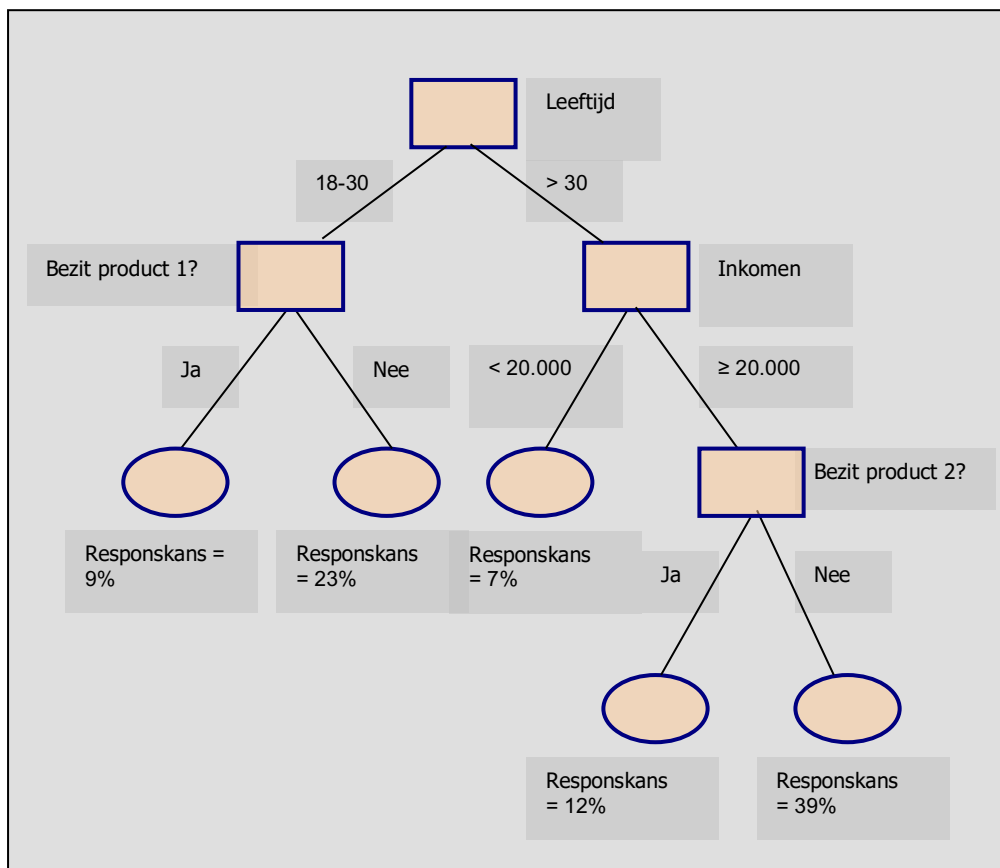
Logistische regressie is een lineaire methode; alle informatie omtrent de verklarende variabelen wordt samengevat in één enkele lineaire combinatie van deze variabelen. De ruimte der verklarende variabelen wordt opgesplitst in twee half-ruimten, die elk corresponderen met één waarde van de te voorspellen binaire variabele.

Bij logistische regressie is het dus de bedoeling om de lineaire vergelijking te vinden die de ruimte het beste opsplijst in twee half-ruimten. Iedere ruimte stelt dan een klasse voor. De objecten kunnen geclassificeerd worden op basis van de half-ruimte waarin ze vallen.

In het geval van de postbank wordt de ruimte der verklarende variabelen in twee half-ruimten opgesplitst op basis van het aantal mailings dat gedaan moet worden. De respondenten waarvan de kans op responderen het grootst is, krijgen een mailing toegestuurd. De lineaire vergelijking die de ruimte opsplijst in twee half-ruimten hangt dus uiteindelijk af van de grootte van de mailing.

## 4. Beslisbomen

Een andere techniek om een bepaald object te classificeren of om succeskansen te schatten, is de beslisboomtechniek. Dit wordt gedaan op basis van de waarde van de karakteristieken van dat object; de waarden van de verklarende variabelen. Een eenvoudig voorbeeld van een beslisboom wordt weergegeven in Figuur 4.1. In een beslisboom stelt iedere knoop een test voor op de waarde van een verklarende variabele. De terminale knopen leggen de klassen van een object vast. Wanneer de klasse van een object bepaald moet worden, zal dit object de boom binnenkomen via de wortel; dit is de knoop boven aan de boom. Afhankelijk van de waarden van de verklarende variabelen worden de verschillende knopen doorlopen. Uiteindelijk belandt het object bij een terminale knoop van de boom. Deze knoop geeft de kans weer dat het object tot een bepaalde klasse behoort.



Figuur 4.1: Voorbeeld beslisboom

### De stap naar de praktijk

Als eenmaal de beslisboom is geconstrueerd, dan wordt afhankelijk van de grootte van de mailing een selectie gemaakt. Allereerst worden de klanten geselecteerd die na het doorlopen van de boom eindigen in de knoop met de hoogste kans op responderen.

Indien de grootte van de mailing kleiner is dan het aantal klanten behorende bij die knoop, wordt er een willekeurige selectie gedaan uit die klanten. Indien de grootte van de mailing groter is, dan wordt de rest van de mailing verdeeld over de klanten behorende bij de knoop (knoop) met de daarop volgende hoogste responskans(en).

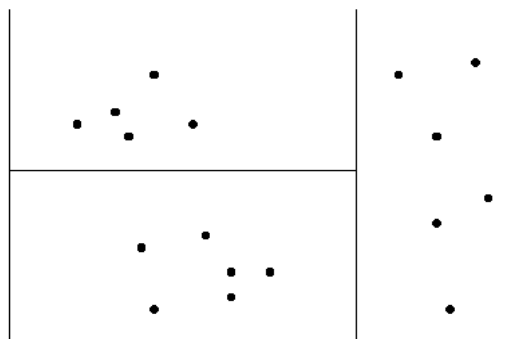


Stel dat een mailing 100.000 groot is en dat de knoop met de hoogste responskans 80.000 klanten bevat. De 80.000 klanten worden dan geselecteerd voor de mailing. De rest van de mailing (20.000) wordt dan verdeeld over de klanten die horen bij de knoop met de daarop volgende hoogste responskans. Indien deze knoop meer dan 20.000 klanten bevat, worden er willekeurig 20.000 van deze klanten geselecteerd voor de mailing.

### Beslisboomtechniek in geometrische termen

De beslisboomtechniek is een niet-lineaire techniek, die de ruimte van de verklarende variabelen opsplijst in meerdere rechthoeken (meer precies hyper-rechthoeken) [Croux en Lemmens (2003)]. Dit gebeurt volgens een slim opgesteld algoritme en leidt tot een optimale verdeling van de ruimte in verschillende rechthoeken. Elke rechthoek wordt toegewezen aan een bepaalde klasse (of de kans op een bepaald klasse). De beslisboomtechniek kan gezien worden als het tekenen van rechthoeken rondom de punten, objecten die tot één groep behoren. Ieder punt in een rechthoek behoort dan tot dezelfde klasse. Dit is in tegenstelling tot statistische classificatietechnieken zoals lineaire en logistische regressie, waarbij de data in klassen wordt verdeeld door het trekken van een lijn of het construeren van een hypervlak door de data ruimte.

Een voorbeeld van de beslisboomtechniek in een twee-dimensionale ruimte wordt gegeven in onderstaand figuur. In het geval van de Postbank correspondeert ieder gebied met een bepaalde responskans. Klanten die in gebieden vallen met de grootste succeskans worden geselecteerd voor de mailing.



Figuur 4.2: Voorbeeld beslisboomtechniek in twee dimensionale ruimte

Beslisbomen verdelen de objecten in disjuncte verzamelingen. De verzamelingen worden elk beschreven door een bepaalde *beslisregel* gebaseerd op een of meerdere verklarende variabelen. Ieder pad dat doorlopen wordt in de boom stelt een beslisregel voor. Beslisbomen classificeren een object dus op basis van beslisregels die uitgedrukt zijn in termen van de onafhankelijke variabelen. Wanneer men verwacht dat er onderliggende beslisregels bestaan, zou de beslisboomtechniek gekozen kunnen worden. Het hoeft echter niet zo te zijn dat de beslisboom dan de beste techniek is om de objecten te classificeren.

### Het bouwen van de boom

Voor het opsplitsen van de ruimte van verklarende variabelen in meerdere rechthoeken, worden er algoritmes gebruikt. Er zijn verschillende algoritmes ontwikkeld. Deze algoritmes

zijn gebaseerd op het testen van de waarden van de onafhankelijke variabelen bij iedere knoop. Ieder algoritme gebruikt een andere test om te controleren of er op basis van de waarde van de variabele een significant verschil bestaat, waardoor de objecten opgesplitst kunnen worden. In dit verslag worden de algoritmes van de technieken CHAID en DataDistilleries besproken. Andere bekende beslisboomtechnieken zijn ID3, C4.5 en CART. Voor meer informatie over deze twee technieken verwijst ik naar [Berry en Linoff (1997)].

## 4.1 CHAID

CHAID staat voor Chi Squared Automatic Interaction Detector en is een methode voor het classificeren van categorische data. Deze methode bouwt bomen die twee of meer vertakkingen aan één knoop hebben. Het doel van deze methode is het opsplitsen van een verzameling objecten in verschillende subgroepen. Dit gebeurt op basis van de verklarende variabelen. CHAID analyseert de waarden van de verklarende variabelen ten opzichte van de waarden van de afhankelijke variabele voor het vinden van significante verschillen. Voor iedere verklarende variabele worden de objecten op een zodanige manier opgesplitst, dat de verkregen subgroepen (segmenten) significant van elkaar verschillen met betrekking tot de afhankelijke variabele. De segmenten overlappen elkaar niet; ieder object kan dus maar in precies één segment voorkomen.

### Componenten van CHAID-analyse

De componenten die nodig zijn om een CHAID analyse uit te voeren zijn:

- Een of meer verklarende variabelen. Deze variabelen moeten categorisch zijn of daarnaar getransformeerd worden.
- Een afhankelijke variabele
- Significantie niveau ( $\alpha$ ) voor de toets om te bepalen welke objecten samengevoegd worden en welke niet.
- Stopcriterium om het proces van opsplitsen te beëindigen. Hierdoor wordt geprobeerd om te stoppen met het groeien van de boom, voordat overfitting plaats vindt. Een stopcriterium kan bijvoorbeeld het aantal subgroepen zijn of de minimale groepsgrootte van een subgroep of gewoon “met de hand stoppen” (methode die bij de Postbank gebruikt wordt).

### Algoritme

Beschouw een afhankelijke variabele met  $d \geq 2$  categorieën en een verklarende variabele met  $c$  categorieën waarbij  $c \geq 2$ . De data kunnen gebruikt worden voor het vullen van een  $c \times d$  – kruistabel<sup>5</sup>. De eerste stap in het algoritme is om het aantal rijen in de kruistabel te verminderen door het samenvoegen van categorieën die niet significant van elkaar verschillen. Het resultaat is dan een  $j \times d$  - tabel ( $j = 2, 3, \dots$ , of  $c$ ) die rijen (subgroepen) bevat die het meest van elkaar verschillen met betrekking tot de waarden van de afhankelijke variabele.

---

<sup>5</sup>  $k \times r$  – kruistabel: de mogelijke waarden van de eerste en de tweede variabele wordt in  $k$ , respectievelijk,  $r$  disjuncte categorieën of intervallen opgedeeld, waarna een tabel gemaakt wordt bestaande uit  $k$  rijen en  $r$  kolommen, zodanig dat op de  $i$ -de rij in de  $j$ -de kolom de frequentie komt te staan van het aantal objecten in de data waarvan de eerste variabele in categorie  $i$  zit en de tweede in categorie  $j$ .

Beschouw ter illustratie de volgende kruistabel:

	Reageren	Niet reageren	Totaal
Regio1	88	21	109
Regio2	97	34	131
Regio3	35	42	77
Regio4	101	15	116
Regio5	90	8	98
Totaal	415	120	535

Tabel 4.1: Voorbeeld kruistabel

Het aantal klanten dat gereageerd heeft op een mailing bedraagt dus 415, waarvan 88 uit Regio1, 97 uit Regio2, 35 uit Regio3, 101 uit Regio4 en 90 uit Regio5. Het totaal aantal verzonden mailings bedraagt 535. De vraag is of de kans op reageren gelijk is voor de verschillende regio's. De volgende nulhypothese wordt dan per paar getoetst:

$H_0$  : de kans op reageren is voor beide regio's gelijk.

Voor  $k \times r$  – tabellen is de toetsingsgroottheid als volgt gedefinieerd:

$$T = \sum_{i=1}^k \sum_{j=1}^r \frac{(N_{ij} - n\hat{p}_{ij})^2}{n\hat{p}_{ij}} \quad (4.1)$$

met

- $N_{ij}$  := het aantal objecten behorende bij de  $i$ -de rij en de  $j$ -de kolom van de kruistabel
- $n$  := het totaal aantal objecten in de tabel ( $n = \sum_i \sum_j N_{ij}$ ).
- $\hat{p}_{ij}$  := de geschatte kans op plaats  $(i,j)$  in de kruistabel onder de nulhypothese,

$$\hat{p}_{ij} = \frac{N_{i.} N_{.j}}{n^2}$$

$T$  heeft onder de nulhypothese voor 'grote steekproefomvang' **bij benadering** een  $\chi^2_{k-1}$  -verdeling [Everitt, (1977)]. De nulhypothese wordt verworpen voor grote waarden van de toetsingsgroottheid  $t$ ;  $t > \chi^2_{\alpha, k-1}$  bij een betrouwbaarheidsdrempel  $\alpha$ . Indien de nulhypothese wordt verworpen bestaat er een significant verschil tussen de categorieën met een bepaalde foutkans ( $p$ -waarde) en worden de categorieën niet samengevoegd.

Voor iedere  $j \times d$  – tabel wordt een  $p$ -waarde berekend. De eerste stap in het CHAID-algoritme is het maken van een  $j \times d$  - kruistabel voor iedere verklarende variabele. Vervolgens worden de objecten (hier: klanten) in subgroepen opgesplitst op basis van de verklarende variabele met de kleinste  $p$ -waarde. Voor iedere gevonden subgroep wordt het proces herhaald totdat er aan een stopcriterium wordt voldaan. De stappen van CHAID worden in Figuur 4.4 weergegeven.

**Algoritme CHAID**

1. Maak voor iedere verklarende variabele een  $c \times d$  -kruistabel met alle categorieën van de verklarende variabelen en de categorieën van de afhankelijke variabele. Voer voor elk van de verklarende variabelen de volgende stappen uit voor het maken van een  $j \times d$  -kruistabel ( $j = 2, 3, \dots, c$ ) :

*Samenvoegen:*

- a. Voer voor ieder paar categorieën de  $\chi^2$ -toets uit en beschouw het paar met categorieën die het minst significant van elkaar verschillen. Dit is het paar met de grootste  $p$ -waarde.

Als deze  $p$ -waarde groter is dan het significantieniveau  $\alpha$ , voeg deze categorieën samen tot één categorie. Herhaal deze stap voor de nieuwe categorieën, totdat er geen  $p$ -waarde te vinden is die groter is dan  $\alpha$ .

- b. Beschouw de nieuwe categorieën die zijn opgebouwd uit drie of meer van de originele categorieën. Bepaal de meest significante opsplitsing binnen deze categorie. Als de  $p$ -waarde kleiner is dan het significantieniveau  $\alpha$ , maak deze opsplitsing en ga terug naar stap a.

Resultaat: optimale partitie voor iedere verklarende variabele met betrekking tot de waarden van de afhankelijke variabele.

2. *Opsplitsen:* De objecten worden opgesplitst in subgroepen o.b.v. de best verklarende variabele uit stap 1. Dit is de variabele met de kleinste  $p$ -waarde die hoort bij de  $\chi^2$ -toets behorende bij de uiteindelijke partitie.
3. *Stoppen:* Voor iedere gevonden subgroep uit stap 2 wordt het proces herhaald totdat er aan een stopcriterium wordt voldaan.

Figuur 4.4: Algoritme CHAID

Het resultaat is een boomstructuur, waarin de subgroepen opgenomen zijn. De boom kan dan gebruikt worden voor het classificeren van nieuwe objecten. Let wel dat de statistische toets die gebruikt wordt bij het algoritme slechts een benadering van de  $\chi^2$ -toets is. Een grote hoeveelheid data is daarom nodig voor het verkrijgen van betrouwbare resultaten.

## 4.2 DataDistilleries

In 2000 heeft CI een nieuwe tool gekocht, namelijk Model Builder van DataDistilleries (DD). In de tool wordt de techniek “*decision list*” toegepast. Decision List zoekt naar groepen van klanten waarvan de kans op een bepaalde gebeurtenis, bijvoorbeeld de kans op responderen, significant afwijkt van het gemiddelde van deze kans voor de gehele steekproef (“*extreme deviant behaviour*”). Er worden modelsegmenten gegenereerd totdat de nieuw gegenereerde modelsegmenten niet meer beter zijn dan gemiddeld. Een modelsegment kan bijvoorbeeld zijn: Man & Getrouwd & Inkomen  $\geq 5000$ . Zodra een segment oftewel regel gevonden is, worden de klanten die aan deze regel voldoen, weggestreept uit de data en vormen de resterende klanten de “*remainder*”. Vervolgens wordt in de “*remainder*” weer naar een groep klanten gezocht met “*extreme deviant behaviour*” en wordt deze weer weggestreept van de populatie en vormen de resterende klanten de (nieuwe) “*remainder*”. Het proces wordt

herhaald totdat aan een stopcriterium wordt voldaan. De lijst van regels die op deze manier wordt gevonden vormt het model. In Figuur 4.5 wordt het algoritme van DD weergegeven.<sup>6</sup>

Significantie wordt getoetst met behulp van de  $t$ -toets. Er wordt tweezijdig getoetst op geen verschil tussen de gemiddelde kansen van de “remainder” en de nieuwe groep van klanten.

#### **Algoritme DataDistilleries**

1. Genereer een modelsegment, een groep klanten die voldoen aan een bepaalde regel.
2. Verwijder de gevonden groep klanten bij stap 1 uit de steekproef. De resterende groep klanten vormt de “remainder”.
3. Herhaal stap 1 en 2 totdat aan een stopcriterium wordt voldaan.

*Figuur 4.5: Algoritme DataDistilleries*

<sup>6</sup> Door gebrek aan informatie over het algoritme van DD kon ik mij hierin niet verder verdiepen.

## 5. Regressie versus Beslisboom

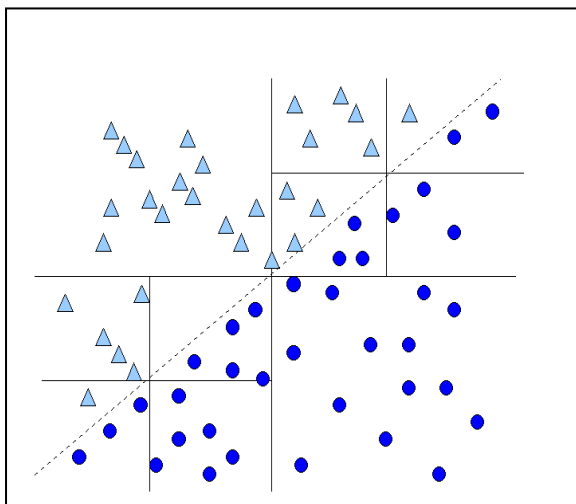
In de vorige hoofdstukken werden de technieken Regressie en Beslisbomen besproken. Daarin werden de technieken logistische regressie, CHAID en DataDistilleries uitgelegd. Deze technieken worden onder andere gebruikt bij het bouwen van mail-modellen en propensity-modellen voor de Postbank.

In dit hoofdstuk worden de technieken Regressie en Beslisbomen met elkaar vergeleken.

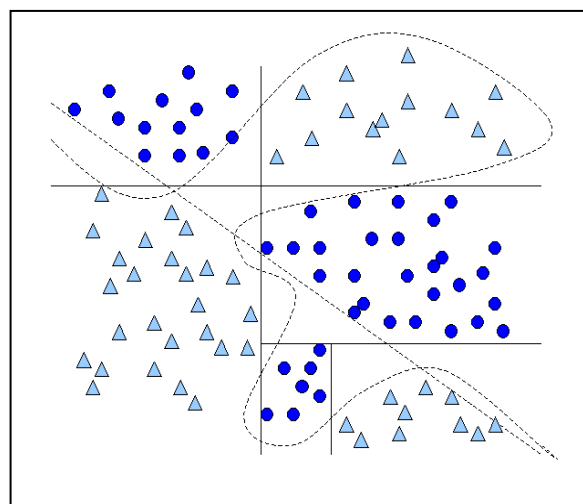
### Regressie versus Beslisbomen in geometrische termen

De beslisboomtechniek kan gezien worden als het tekenen van rechthoeken (meer precies hyper-rechthoeken) rondom de punten, objecten die tot één groep behoren<sup>7</sup>. Ieder punt in een rechthoek behoort dan tot dezelfde klasse. Dit is in tegenstelling tot de regressietechniek zoals lineaire en logistische regressie, waarbij de data in klassen wordt verdeeld door het trekken van een lijn (of tekenen van een hypervlak) door de data ruimte. Bij logistische regressie wordt de ruimte van verklarende variabelen opgesplitst in twee halfruimten, elk corresponderend met één waarde van de te voorspellen binaire/dichotome variabele. In Figuur 3.1 en Figuur 4.2 worden voorbeelden gegeven van respectievelijk de beslisboomtechniek en de regressietechniek in een twee-dimensionale ruimte.

In sommige situaties is het beter om de ruimte in twee half-ruimten te verdelen in plaats van in rechthoeken. Een duidelijk voorbeeld hiervan wordt gegeven in Figuur 5.1. Links boven en rechts onder zijn makkelijk te verdelen in rechthoeken, maar voor rechts boven en links onder is dit knap lastig. Deze gebieden moeten in heel veel kleine rechthoeken verdeeld worden om grenzen te kunnen leggen tussen de verschillende groepen. In Figuur 5.2 is echter te zien dat het trekken van rechthoeken een betere methode is om de objecten in klassen te verdelen. In dit geval is er geen simpele curve te vinden die de verschillende groepen opsplijst.



Figuur 5.1: Ruimte is beter te verdelen door middel van een enkele lijn



Figuur 5.2: Ruimte verdelen in rechthoeken is beter dan het trekken van een enkele lijn of een curve

Technieken die een enkele lijn gebruiken om klassen te onderscheiden, zoals regressie, zijn zwak in situaties waarin er meerdere manieren zijn voor een object om tot een bepaalde klasse

<sup>7</sup> De zijvlakken van de (hyper)rechthoeken staan loodrecht op de coördinaat-assen.

te behoren [Berry en Linoff (1997)]. Neem als voorbeeld de “creditcard”. Twee verschillende klanten met bijvoorbeeld totaal verschillende uitgavenpatronen kunnen beide als kredietwaardig geclassificeerd worden. De ene klant maakt bijvoorbeeld lage transactiekosten, maar omdat zijn saldo steeds hoog is, zal hij altijd kunnen voldoen aan zijn verplichtingen en wordt hij als kredietwaardig geclassificeerd. De andere klant maakt iedere maand het totaal bedrag van zijn rekening in één keer over, maar wordt als kredietwaardig geclassificeerd vanwege de grote transacties die hij maakt. Indien er meerdere manieren zijn voor een object om tot een bepaalde klasse te behoren, zullen de groepen verspreid zijn in de ruimte, zoals in Figuur 5.2 het geval is.

In de praktijk is gebleken dat de beslisboomtechniek beter weet om te gaan met interactietermen<sup>8</sup> dan de logistische regressietechniek. Dat de beslisboomtechniek de ruimte opsplijt in meerdere (hyper)rechthoeken, en hierdoor flexibeler is in situaties waarin er meerdere manieren zijn voor een object om tot een bepaalde klasse te behoren, zou een verklaring hiervoor kunnen zijn. Maar dit is slechts een mogelijke verklaring. Er zal verder onderzoek hiernaar gedaan moeten worden om deze uitspraak te kunnen doen.

Vanwege de vele rechthoeken in de dataruimte hebben beslisbomen daarentegen vaker te maken met het “*grensgeval-probleem*”. Hiermee wordt bedoeld dat een object nog net binnen een bepaald gebied valt. Door een kleine verandering aan de grensligging valt dit object onder een ander gebied. Omdat bij de beslisboomtechniek de data wordt opgesplitst door het tekenen van meerdere rechthoeken, heeft deze techniek veel vaker te maken met dit probleem dan bij de regressietechniek, waar er maar één grens getrokken wordt.

## 5.1 Vergelijken van classificatietechnieken

Voor het vergelijken van classificatietechnieken kunnen diverse criteria toegepast worden. Het meest voor de hand liggende is de *juistheid/kwaliteit* van het model; hoe goed sluit het model aan bij de werkelijke data. Andere belangrijke criteria zijn *robuustheid* en *gevoeligheid*; hoe goed kan de techniek tegen veranderingen in de training-data. Een ander criterium is *interpreteerbaarheid* van het model. Ook *implementeerbaarheid* van de techniek is een belangrijk criterium; de mate waarin de techniek binnen de organisatie efficiënt kan worden ingezet. Verder kan nog gekeken worden naar de *typen variabelen* waarmee de techniek om kan gaan.

### 5.1.1 Juistheid van het model

Voor het meten van de juistheid van het model zijn er verschillende methoden beschikbaar. De eenvoudigste methode is het bepalen van het percentage verkeerd geclassificeerde objecten; de “error rate”. In hoofdstuk 3 werden er ook verschillende methoden behandeld, namelijk  $R^2$ , “Deviance” en “Pearson chisquared statistic”. Deze methoden zijn niet geschikt voor alle situaties, omdat iedere foute classificatie (of correcte classificatie) even zwaar wordt behandeld. Vooral in de marketing is er een verschil tussen foute classificaties (of correcte classificaties). Bijvoorbeeld, het classificeren van een klant als een niet-respondent, terwijl hij

---

<sup>8</sup> Interactietermen worden gebruikt wanneer er interactie is tussen de verschillende termen en deze interactie invloed heeft op de waarde van de afhankelijke variabele.

dat wel is, telt zwaarder dan een klant classificeren als een respondent terwijl hij dat niet is. In het laatste geval verlies je alleen de kosten van de mailing, terwijl je in het eerste geval de winst van het product misloopt. Een geschikte methode in zo een situatie is de *lift curve*. Postbank maakt ook gebruik van de liftcurve.

Een andere methode voor het meten van de kwaliteit van een model in zo een situatie is de *ROC-curve*

### *Confusion matrix*

Beschouw de 2-klasse situatie, waarbij de afhankelijke variabele de waarde 'ja' of 'nee' kan aannemen. De vier mogelijke uitkomsten bij een enkele voorspelling zijn weergegeven in Tabel 5.2. Deze tabel staat bekend als de "*confusion matrix*".

- Correct positief: het object is correct geclassificeerd als 'ja'
- Correct negatief: het object is correct geclassificeerd als 'nee'
- Vals positief: het object is geclassificeerd als 'ja' terwijl het in werkelijkheid 'nee' is.
- Vals negatief: het object is geclassificeerd als 'nee' terwijl het in werkelijkheid 'ja' is.

De twee typen fouten (Vals) wegen niet even zwaar en hebben verschillende kosten. Dit geldt ook voor de twee typen correcte classificaties (Correct); een klant die geclassificeerd is als een respondent en het ook daadwerkelijk is, levert meer op dan een klant die geclassificeerd is als een niet-respondent en dat ook niet is.

Voorspelling	<i>Ja</i>	<i>Nee</i>
Werkelijk		
<i>Ja</i>	Correct Positief	Vals Negatief
<i>Nee</i>	Vals Positief	Correct Negatief

Tabel 5.2: *Confusion matrix voor 2-klassen probleem*

### De lift curve

In de marketing wordt veelal de lift curve gebruikt om de kwaliteit van een model te bepalen. Veronderstel dat op basis van voorgaande experimenten bekend is dat het aantal respondenten dat resulteert uit een mailing van één miljoen groot ongeveer gelijk is aan 1000. Het succespercentage is dus ongeveer gelijk aan 0.1%. Neem aan dat het model 100.000 klanten selecteert, waarvan het responspercentage gelijk is aan 0.4% (400 respondenten). Dit is dus een toename van een factor 4 ( $0.4 / 0.1$ ). De toename in het percentage respondenten wordt de *lift factor* genoemd.

### *Construeren van de lift curve*

Veronderstel dat je een dataset tot je beschikking hebt waarvan de classificaties bekend zijn. De dataset bestaat uit 100.000 klanten waarvan 20.000 respondenten zijn. Het totale succespercentage is dus gelijk aan 20%. Het model wordt vervolgens uitgevoerd op de test set en levert als resultaat de geschatte responskans van de klanten. Bekijk vervolgens de klanten

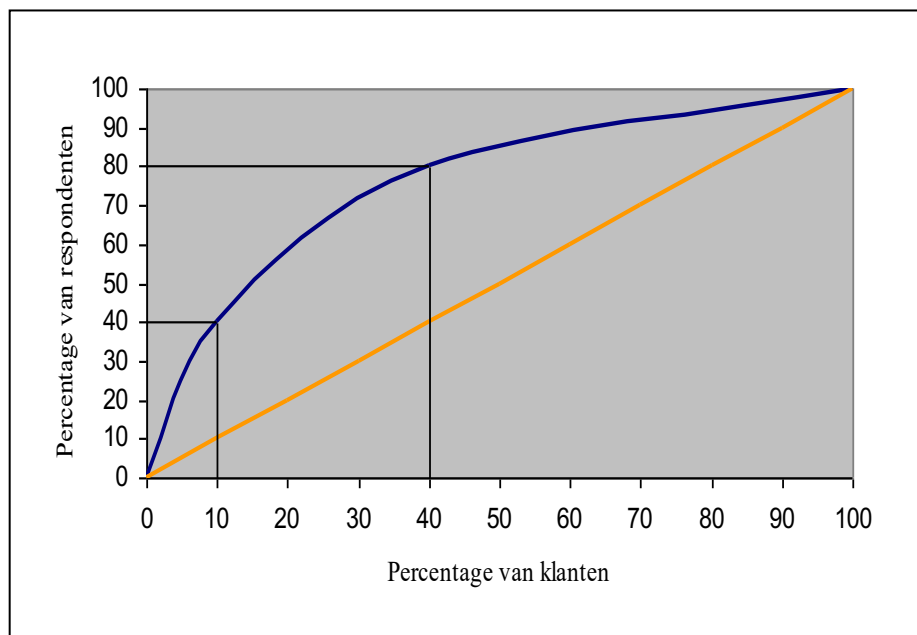


in de test set en sorteer deze in afnemende grootte op basis van de geschatte responskans. Indien je een mailing wilt doen van 10.000 groot, dan bestaat de selectie uit de eerste 10.000 klanten uit de gesorteerde lijst. Deze hebben immers de grootste kans op responderen. Vervolgens kan de lift factor berekend worden:

$$\text{Lift factor} = \frac{\text{succespercentage van de selectie}}{\text{totale succespercentage}}$$

Stel dat blijkt dat 8000 klanten van de 10.000 correct geclassificeerd zijn als respondent. Het succespercentage van de selectie is dan gelijk aan 80%. Dit levert een lift factor gelijk aan 4 op ( $80/20 = 4$ ). Herhaal dit voor verschillende selectiegroottes. Op deze manier kan een lift curve geconstrueerd worden zoals in Figuur 5.3.

Op de horizontale as staan de verschillende selectiegroottes als percentage van het totaal aantal klanten (hier: 100.000). De verticale as geeft het aantal werkelijke respondenten weer als percentage van het totaal aantal respondenten (hier: 20.000). In dit voorbeeld levert 10% van het totaal aantal klanten (een selectie van 10.000) een responspercentage van 40% ( $= (8000/20.000) \times 100\%$ ) op. De diagonale lijn geeft de verwachte respondenten weer als de klanten willekeurig geselecteerd worden. De curve correspondeert met het resultaat als de selecties op basis van het model gekozen worden.



Figuur 5.3: Voorbeeld lift curve

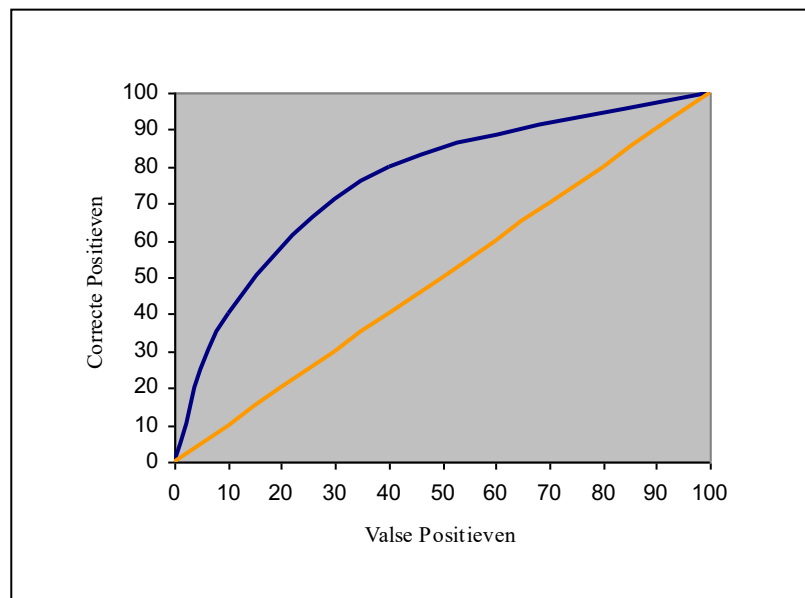
Je wilt bij een zo klein mogelijke selectiegrootte een zo groot mogelijk percentage respondenten bereiken. Dit komt overeen met een hoge lift curve; hoe hoger de lift curve, hoe beter het model. De ideale situatie in het besproken voorbeeld is een responspercentage van 100% bij een selectie van 20%; 20.000 klanten reageren bij een mailing van 20.000 groot.

#### De ROC-curve

De ROC-curve (Receiver Operating Characteristics) geeft de relatie weer tussen het aantal respondenten van de selectie als percentage van het totaal aantal respondenten van de gehele

steekproefgrootte, en het aantal niet-respondenten van de selectie als percentage van het totaal aantal niet-respondenten van de gehele steekproefgrootte. Met andere woorden, er wordt een verband weergegeven tussen de correcte positieven en de valse positieven. Een voorbeeld wordt gegeven in Figuur 5.4.

De verticale as komt overeen met de verticale as van de lift curve. Op de horizontale as staat het percentage valse positieven. In geval het percentage respondenten (positieven) in de steekproefgrootte heel klein is (bijvoorbeeld 0.1 %), dan lijken een lift curve en een ROC-curve heel erg op elkaar. Het verschil tussen selectiegrootte en het aantal niet-respondenten (negatieven) is dan immers verwaarloosbaar.



Figuur 5.4.: Voorbeeld ROC-curve

Net als bij de lift curve is het doel hier om de oppervlakte onder de curve zo groot mogelijk te krijgen. Je wilt immers dat het percentage correcte classificaties als respondent zo groot mogelijk is.

Het gebied onder de ROC-curve is ook een geschikte manier om modellen met elkaar te vergelijken. Een model dat willekeurig classificeert heeft oppervlakte die gelijk is aan 0.5 en in de ideale situatie is de oppervlakte gelijk aan 1. In de ideale situatie is het percentage correcte positieven gelijk aan 100% en het percentage valse positieven gelijk aan 0%.

Beschouw het voorbeeld van de constructie van de lift curve. De steekproef is 100.000 groot. In geval de selectie 10.000 groot is, is het aantal respondenten 8000. Het percentage correcte positieven is dus in dit geval gelijk aan 40% ( $8000/20.000 \times 100\%$ ). 2000 zijn verkeerd geassocieerd als respondent. Het percentage valse positieven is in dit geval gelijk aan ( $2000/80.000 \times 100\%$ ) = 2.5%.

### 5.1.2 Robuustheid en gevoeligheid

### *Robuustheid*

De beslisboomtechniek is robuuster dan de regressietechniek. Indien er een grote verandering in de dataset plaats vindt, zal dit meer effect hebben op de regressievergelijking. Robuustheid heeft te maken met de gevoeligheid voor uitbijters. De regressietechniek werkt naar een gemiddelde toe. Indien de training-dataset uitbijters bevat, zullen deze heel veel invloed hebben op de regressiecurve (of vlak). De beslisboomtechniek kan beter omgaan met uitbijters.

Dit geval kan vergeleken worden met het gemiddelde (regressie) versus de mediaan (beslisboom).

### *Gevoeligheid*

De beslisboomtechniek is een onstabiel proces als het gaat om een kleine verandering in de training-dataset. Deze kan er makkelijk voor zorgen dat op een bepaald punt in de boom een andere verklarende variabele geselecteerd wordt om de data op te splitsen. Dit heeft effect op de verdere vertakking van de boom. De verandering kan dus resulteren in een heel andere boom. Dit wil niet zeggen dat het een groot effect heeft op de classificatie. Het risico is alleen groter dat het een effect zou hebben op de classificatie.

Dit probleem gaat samen met het grensgeval-probleem.

De regressietechniek is een stabielere techniek. Het model verandert heel weinig als er een kleine verandering in de training-dataset plaatsvindt.

### **5.1.3 Interpreteerbaarheid**

Beslisbomen geven een duidelijk beeld welke variabelen het belangrijkst zijn voor classificatie. De variabele die het beste onderscheid maakt, correspondeert met de eerste knoop die onder aan de wortel wordt geplaatst. Een beslisboom is tevens eenvoudig verklaarbaar vanwege het gebruik van beslisregels. De beslisbomen zijn makkelijk te vertalen naar een taal begrijpelijk voor de mens.

De regressietechniek is in vergelijking tot de beslisboomtechniek minder eenvoudig te verklaren en minder makkelijk te vertalen naar een begrijpelijke taal voor de mens.

### **5.1.4 Implementeerbaarheid binnen de Postbank**

Door de tool van Datadistilleries (DD) zijn beslisboomtechnieken binnen de Postbank goed te implementeren. Deze tool zorgt ervoor dat de resultaten van de beslisboommodellen meteen op de juiste plaats terechtkomen voor verder gebruik; het proces is volledig geautomatiseerd. In geval van regressie zijn er meer handelingen nodig om de resultaten op de juiste plaats te krijgen. In deel 2 van dit verslag wordt dit proces uitgebreid beschreven. Het gevolg van de meerdere “handmatige” handelingen die nodig zijn bij regressie is, dat voor een regressiemodel al snel twee dagen werk nodig is. Een beslisboom is dus goedkoper te produceren.

### **5.1.5 Typen variabelen**

Beslisbomen kunnen omgaan met zowel categorische als continue verklarende variabelen. In geval van categorische variabelen wordt er voor iedere categorie of een groep categorieën een tak genomen. Bij continue variabelen wordt er gesplitst door een of meerdere getallen te nemen in het interval van de waarden.

Beslisbomen zijn minder geschikt voor classificatie situaties waarbij de doelvariabele continu is, omdat deze dan zeer veel terminale knopen moet bevatten om goed te kunnen werken. Dit zal leiden tot overfitting, omdat iedere waarde in zijn eigen rechthoek wordt geplaatst. Wanneer in zo een situatie toch voor de beslisboomtechniek wordt gekozen, wordt de doelvariabele normaliter gediscrèteerd; er wordt per eindknoop een gemiddelde bepaald voor de schatting van de waarde van de doelvariabele. Dit zorgt er helaas ervoor dat de schatting minder zuiver is.

Regressie kan omgaan met zowel continue als categorische verklarende variabelen. Om categorische verklarende variabelen op te nemen in een regressiemodel kunnen indicator- of dummy variabelen gebruikt worden (zie paragraaf 3.3).

Regressie kan goed omgaan met een continue afhankelijke variabele. Indien de afhankelijke variabele twee categorieën heeft, kan logistische regressie toegepast worden. Als er meer dan twee categorieën zijn, kan er geprobeerd worden om de afhankelijke variabele te hercoderen naar twee categorieën. Een andere mogelijkheid is om meerdere logistische regressie analyses uit te voeren, waarbij je steeds twee categorieën met elkaar vergelijkt. Bijvoorbeeld bij een variabele met drie categorieën  $a$ ,  $b$  en  $c$  kun je drie logistische regressie analyses uitvoeren: één voor de kans  $a$  versus  $b$ ; één voor de kans  $a$  versus  $c$ ; en één voor de kans  $b$  versus  $c$ . Het nadeel hiervan is dat de resultaten van de verschillende analyses meestal niet dezelfde uitkomsten geven. De beste oplossing is dan ook om een zogenaamde multinomiale logistische regressie analyse toe te passen. Deze techniek is een uitbreiding van de gewone logistische regressie analyse en wordt hier verder niet behandeld. Voor de geïnteresseerde lezer verwijs ik naar [Lammers, Pelzer en Hendrickx (1998)].

## 5.2 Onderzoek afdeling R&M

In het onderzoek van de afdeling R&M [Westerlaken (2003)] werden de technieken, Logistische Regressie, CHAID en DD met elkaar vergeleken. Hiervoor werden met elk van de technieken modellen gebouwd. De kwaliteit van de verkregen modellen werd onder andere vergeleken op basis van:

1. de *liftcurve* van de modellen. Deze kan voor ieder model bepaald worden, waardoor vergelijking van de verschillende technieken mogelijk is.
2. de meest significante *variabelen* die met de verschillende technieken naar voren komen

Voor het eerste product is met iedere techniek een *propensity model* gebouwd; de modellen voorspellen de kans op productafname in een bepaalde periode. Voor de andere twee producten zijn *mailmodellen* gebouwd; de modellen voorspellen de kans dat een klant respondeert op een mailing. In totaal zijn er dus negen modellen gebouwd.

Uit het onderzoek is onder andere gebleken dat logistische regressie het beter doet dan de andere technieken in geval er weinig respondenten zijn. Op basis hiervan werd geconcludeerd dat logistische regressie beter omgaat met kleine aantallen respondenten dan CHAID en DD. We bekijken de resultaten van het onderzoek.



### 5.2.1 Resultaten onderzoek R&M

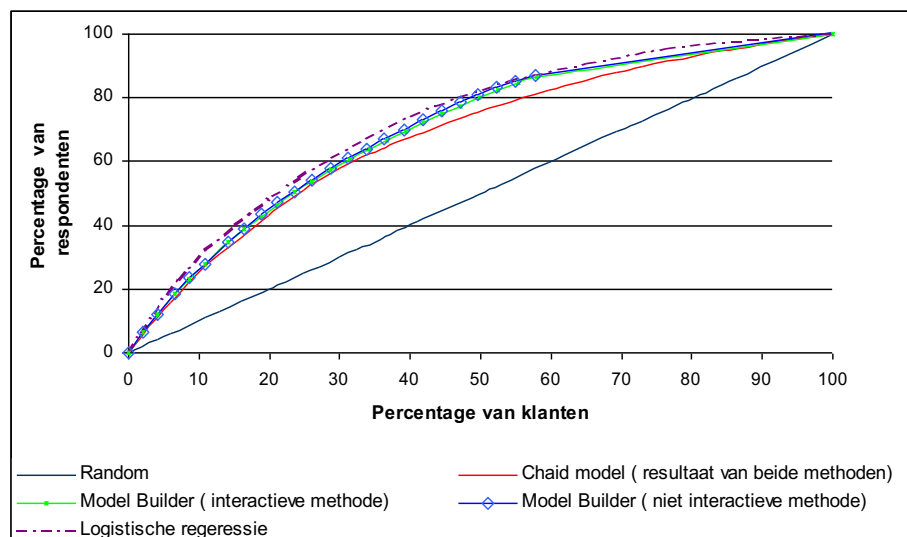
#### Product 1

Voor de bouw van deze modellen is er in de database gekeken welke klanten, over een periode van drie maanden, het product hebben afgenomen. De gegevens behorende bij deze modellen staan weergegeven in Tabel 5.3. Het gaat hier om een grote doelgroep en voldoende respondenten. Het aantal variabelen dat uit de logistische regressie voortkomt is groot. Ook DD bevat een groot aantal variabelen. CHAID bevat opvallend minder variabelen.

Product 1	
Soort model	Propensity model
# respondenten	78.144 (5.86%)
# non-respondenten	1.256.456 (94.14%)
Totaal	1.334.600
# variabelen CHAID	6
# variabelen DD	15
# variabelen Log. Reg	> 50

Tabel 5.3: Gegevens modellen Product 1

De liftcurve voor de drie verkregen modellen behorende bij dit product wordt weergegeven in Grafiek 5.1. De liftcurves lopen praktisch over elkaar heen. CHAID presteert wat minder goed, maar het verschil is niet groot. De modellen/technieken presteren dus ongeveer even goed.



Grafiek 5.1: Lift curve Chaid, Decision List en Logistische regressie voor Product 1

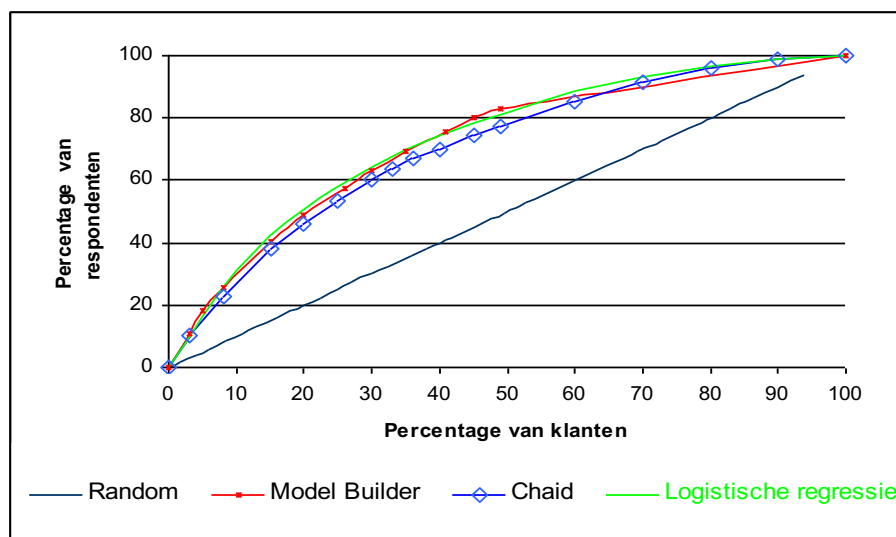
## Product 2

De gegevens behorende bij deze modellen staan weergegeven in Tabel 5.4. Het gaat hier om een redelijk grote doelgroep met voldoende respondenten. Het aantal variabelen dat uit de logistische regressie voortkomt is groot. DD bevat ook veel variabelen, terwijl het aantal variabelen bij CHAID ook hier een stuk minder is.

Product 2	
Soort model	Mail model
# respondenten	4.827 (10.02 %)
# non-respondenten	43.333 (89.98 %)
Totale doelgroep	48.160
# variabelen CHAID	6
# variabelen DD	10
# variabelen Log. Reg	15?

Tabel 5.4: Gegevens modellen Product 2

De liftcurve voor de drie verkregen modellen behorende bij dit product wordt weergegeven in Grafiek 5.2. Zoals te zien is er geen groot verschil op te merken tussen de verschillende technieken. De curve van CHAID ligt weer iets onder de andere curven, maar dit verschil is klein.



Grafiek 5.2: Lift curve Chaid, Decision List en Logistische regressie voor Product 2

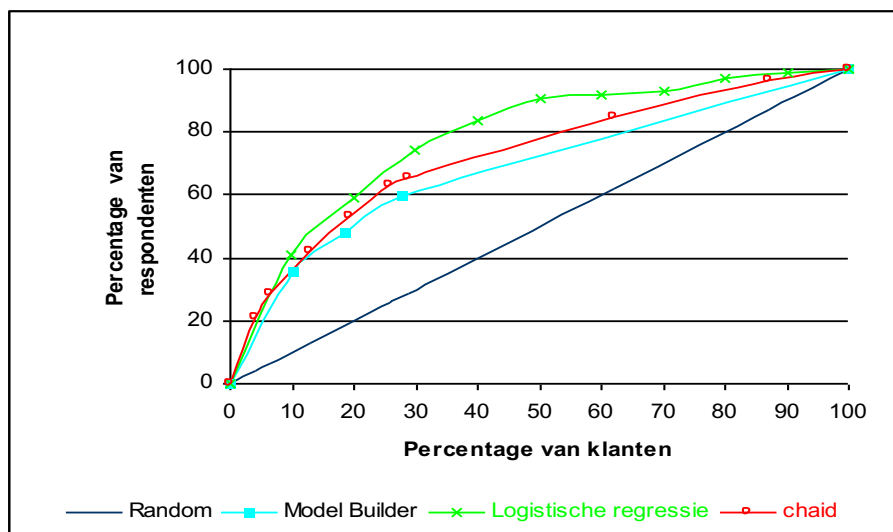
### Product 3

De modellen zijn gebaseerd op een mailing die in November 2001 verzonden is. In Tabel 5.5. staan de gegevens behorende bij deze modellen weergegeven. De doelgroep is hier redelijk groot, maar het aantal respondenten is klein. Logistische regressie levert een groot aantal variabelen op. DD en CHAID daarentegen leveren beide weinig variabelen op.

Product 3	
Soort model	Mail model
# respondenten	269 (0.56 %)
# non-respondenten	48.206 (99.44 %)
Totale doelgroep	48.475
# variabelen CHAID	5
# variabelen DD	5
# variabelen Log. Reg	15?

Tabel 5.5: Gegevens modellen Product 3

De liftcurve voor de drie verkregen modellen behorende bij dit product wordt weergegeven in Grafiek 5.3. Het is duidelijk te zien dat logistische regressie een betere liftcurve heeft dan CHAID en DD. In dit geval levert DD de laagste curve op.



Grafiek 5.3: Lift curve Chaid, Decision List en Logistische regressie voor Product 3



### 5.3 Discussie

#### Onderzoek afdeling R&M

In dit onderzoek werden de drie technieken logistische regressie, CHAID en DD met elkaar vergeleken. Het is gebleken dat logistische regressie het beter doet bij een marketingactie waarin er weinig respondenten voorhanden zijn (Product 3). Wat verder opvalt, is dat het aantal variabelen dat de technieken CHAID en DD in dat geval opleveren, klein is. De mogelijke reden hiervoor is, dat door het kleine aantal respondenten er maar weinig onderscheidende subgroepen gevonden zullen worden. De boom kan zich dus niet verder vertakken en overige variabelen kunnen niet gebruikt worden voor verdere opsplitsing van de data. De informatie van de overige variabelen gaat hierdoor verloren. Logistische regressie kan wel meerdere variabelen gebruiken, zoals te zien is bij Product 3, en krijgt hierdoor meer informatie om een beter model te bouwen.

De mogelijke reden waarom logistische regressie het beter doet in een situatie met een klein aantal respondenten is waarschijnlijk dat logistische regressie meer informatie tot zich opneemt.

#### Algemeen

In onderstaand tabel wordt er een overzicht gegeven van de verschillen tussen de regressietechniek en de beslisboomtechniek.

Regressie	Beslisboom
Opsplitsing data ruimte in twee halfruimten	Opsplitsing data ruimte in meerdere (hyper) rechthoeken
Geschikt voor niet-rechthoekige gebieden	Problemen met niet-rechthoekige gebieden
Minder goed als er meerdere manieren zijn voor een object om tot een bepaalde klasse te behoren	Flexibel in situaties waarin er meerder manieren zijn voor een object om tot een bepaalde klasse te behoren
Moeite met interactie-termen	Flexibel met interactie-termen
“Grensgeval-probleem” komt minder vaak voor	“Grensgeval-probleem” komt vaak voor
Stabiel m.b.t. gevoeligheid	Onstabiel m.b.t. gevoeligheid
Niet goed te interpreteren	Goed te interpreteren
Niet goed te implementeren	Goed te implementeren
Geschikt voor situaties waarbij de doelvariabele continu is	Minder geschikt voor situaties waarbij de doelvariabele continu is

Tabel 5.6: Regressie VS Beslisboom

In onderstaand tabel wordt een vergelijking gedaan tussen de twee technieken aan de hand van de verschillende criteria die in paragraaf 5.1 zijn genoemd:

Criterion	Regressie	Beslisboom
Robuustheid	-	+
Gevoeligheid	+	-
Interpreteerbaarheid	-	+
Implementeerbaarheid (bij de Postbank)	-	+
Typen variabelen	+	±

Tabel 5.7: Regressie VS Beslisboom

De beslisboomtechniek is wat gevoeligheid betreft zeer zwak. Er bestaan echter technieken die de stabiliteit mogelijk kunnen verbeteren. Deze technieken worden in hoofdstuk 6 behandeld.

Zoals werd aangegeven, is een beslisboom voor de Postbank goedkoper dan een regressiemodel. De implementeerbaarheid van logistische regressie binnen de Postbank is minder goed dan die van de beslisboomtechniek. Deel 2 van dit verslag beschrijft een mogelijke oplossing hiervoor.

Logistische regressie weet beter om te gaan met continue afhankelijke variabelen en scoort daarom hoger bij het criterium “Typen variabelen”. In geval van mailmodellen en propensity-modellen gaat het om dichotome afhankelijke variabelen. Zowel de beslisboomtechniek als de logistische regressietechniek kunnen hier goed mee omgaan. In geval van dichotome afhankelijke variabelen scoren de technieken even hoog.

## 6. Boosting en Bagging

Tijdens mijn onderzoek naar regressiemodellen en beslisbomen ben ik technieken tegengekomen, die dienen voor het verbeteren van de prestatie van classificatietechnieken. Het gaat om de technieken Boosting en Bagging die op dit moment erg in de schijnwerpers staan en het in de praktijk uitstekend schijnen te doen [Borra en Di Ciaccio (2002)].

Het idee achter Boosting en Bagging is dat beslissingen niet meer genomen worden op basis van één model, maar op basis van een combinatie van meerdere modellen van hetzelfde type, bijvoorbeeld beslisbomen. Het idee kan vergeleken worden met een vergadering waarbij er verschillende experts aanwezig zijn. Verschillende meningen van verschillende experts worden in beschouwing genomen en op basis van een overeenstemming of een meerderheidsstemming (majority vote) wordt er een beslissing genomen. Bij boosting en bagging stelt één model één expert voor en het doel is om de verschillende modellen te combineren om tot betrouwbare beslissingen te komen. Andere technieken die ook gebruik maken van combinaties van verschillende modellen zijn “Stacking” en “error-correcting output codes”. Voor informatie over deze twee laatste technieken, verwijst ik naar [Witten en Frank (1999)].

### 6.1 Boosting

Boosting vindt zijn oorsprong in de literatuur van “Artificial Intelligence” [Schapire (1990), Freund (1995), Freund en Schapire (1997)]. Het is een iteratieve methode; er wordt begonnen met een basismodel en van daaruit worden er steeds nieuwe modellen ontwikkeld. De bedoeling bij boosting is dat ieder model leert van zijn voorgaande model. Dit gebeurt op basis van het toekennen van gewichten aan objecten.

Het basismodel wordt gebouwd op een training dataset waarvan de objecten gelijke gewichten hebben. Op basis van de prestatie van het basismodel (op de training data) worden er nieuwe gewichten toegekend aan de objecten. Een object dat in het model verkeerd geïdentificeerd is, krijgt een hoger gewicht toegekend en een object dat correct geïdentificeerd is, krijgt een lager gewicht toegekend; de criteriumfunctie wordt aangepast (bijv. de aannemelijkheidsfunctie bij logistische regressie). Vervolgens wordt er op de nieuwe ‘gewogen’ training dataset een model gebouwd en worden er aan de hand van de prestatie (error) van dat model nieuwe gewichten toegekend aan de objecten. Het proces herhaalt zich totdat er aan een stopcriterium is voldaan (bijv. het aantal modellen dat gegenereerd moet worden of als er geen verbetering meer ontstaat in de prestatie van de modellen).

Door het toekennen van gewichten aan objecten kan de classificatietechniek zich concentreren op de objecten die moeilijk te classificeren zijn, namelijk die met het hoogste gewicht.

Als de modellen zijn gebouwd, kan een voorspelling gedaan worden door middel van meerderheidsstemming. Ieder model krijgt een gewicht toegekend dat gebaseerd is op de prestatie (error) van het model. Een model dat goed gepresteerd heeft, krijgt een hoog gewicht toegekend en een model dat slecht gepresteerd heeft, krijgt een laag gewicht toegekend. Om uiteindelijk een voorspelling te doen, worden de gewichten van alle modellen die verwijzen naar een bepaalde klasse gesommeerd en de klasse waarvan het totaal het hoogste is, wordt gekozen als de klasse voor dat object. In Voorbeeld 6.1 wordt dit nader uitgelegd.

Het idee bij boosting is dus dat nieuwe modellen gecorrigeerd worden door ze te laten leren van de fouten of misclassificaties die de voorgaande modellen hebben gemaakt. Ieder model wordt beïnvloed door zijn voorgaande model.

Boosting is een methode om de zuiverheid van de schatting te verhogen en kan op iedere classificatietechniek toegepast worden [Ridgeway (2002)].

### Voorbeeld 6.1

Stel dat er 5 modellen zijn met als gewichten (deze worden bepaald aan de hand van de prestatie (error) van het model):

Model 1: 0.2  
 Model 2: 0.3  
 Model 3: 0.45  
 Model 4: 0.5  
 Model 5: 0.55

De verschillende modellen leveren het volgende als resultaat:

Object	Model 1	Model 2	Model 3	Model 4	Model 5	Klasse
1	Wel	Niet	Wel	Niet	Niet	Niet
2	Niet	Niet	Wel	Wel	Wel	Wel
3	Niet	Wel	Wel	Wel	Niet	Wel
4	Wel	Niet	Niet	Wel	Niet	Niet
5	Niet	Niet	Niet	Wel	Wel	Wel
6	Wel	Wel	Wel	Wel	Niet	Wel
7	Niet	Wel	Wel	Wel	Wel	Wel

Tabel 6.1: Voorbeeld Classificatie

Beschouw object 1; de som van de gewichten van de modellen die naar ‘Wel’ verwijzen is gelijk aan 0.65 (= 0.2 + 0.45) en de som van de gewichten van de modellen die naar ‘Niet’ verwijzen is gelijk aan 1.35 (= 0.3 + 0.5 + 0.55). De uiteindelijke klasse voor object 1 wordt dus “Wel”. Op deze manier worden de klassen van de overige objecten bepaald.

### 6.1.1 Adaboost

Er zijn verschillende boosting methodes ontwikkeld. De methode die het meest wordt gebruikt is AdaBoost, ontwikkeld door Freund en Schapire [Freund en Schapire (1996)]. Een beschrijving van deze methode is als volgt:

- laat  $(x_1, y_1), \dots, (x_N, y_N)$  de training data zijn met  $x_i$  een vector van verklarende variabelen en  $y_i = -1$  of  $1$  (bijv.  $1 =$  responderen en  $-1 =$  niet responderen).
- De functie  $F(x)$  wordt gedefinieerd als  $F(x) = \sum_1^M c_m f_m(x)$ ,  
 waarbij
  - $M$  het aantal modellen is,
  - iedere  $f_m(x)$  een model voorstelt,  $f_m(x) \in \{-1, 1\} \forall x$ ,
  - en  $c_m$  constanten zijn.
- De uiteindelijke voorspelling is  $\text{sign}(F(x))$ .

De AdaBoost procedure fit modellen  $f_m(x)$  op ‘gewogen’ training datasets. Het AdaBoost algoritme is in onderstaand figuur weergegeven.

**AdaBoost**

1. Start met gewichten  $w_i = 1/N$ ,  $i = 1, \dots, N$  waarbij  $N$  het aantal objecten in de training data set is.
2. Herhaal voor  $m = 1, 2, \dots, M$ :
  - Fit het model  $f_m(x) \in \{-1, 1\}$  door gebruik te maken van gewichten  $w_i$  op de training data.
  - Bereken
 
$$\text{err}_m = E_w [1_{(y \neq f_m(x))}] = \sum_{i=1}^N w_i \cdot 1_{(y \neq f_m(x))}$$
  - Bereken  $c_m = \log((1 - \text{err}_m) / \text{err}_m)$ .
  - Laat  $w_i = w_i \exp[c_m 1_{(y \neq f_m(x_i))}]$ ,  $i = 1, 2, \dots, N$  en normaliseer vervolgens zodat  $\sum_i w_i := 1$ .
3. Sign  $\left[ \sum_i^M c_m f_m(x) \right]$

Figuur 6.1: Algoritme voor AdaBoost

$E_w$  staat voor het gewogen gemiddelde over de training-dataset. Bij iedere iteratie worden de gewichten van de verkeerd geclassificeerde objecten verhoogd met een factor die afhangt van de error van het model op de gewogen data.

De invloed van het uiteindelijke resultaat hangt af van de waarde van  $c_m$  (het gewicht van het model  $m$ ). Uit het algoritme kan worden afgeleid dat voor kleine  $\text{err}_m$ ,  $c_m$  groot is en dus een grote invloed heeft op het uiteindelijke resultaat. Voor  $\text{err}_m$  gelijk aan 0, is  $c_m$  niet gedefinieerd. Als  $\text{err}_m$  gelijk is aan 0.5, dan is  $c_m$  gelijk aan 0 en blijven de gewichten onveranderd. Is  $\text{err}_m$  groter dan 0.5, dan is  $c_m$  negatief en heeft dus een negatief effect op het resultaat. Dit betekent dat wanneer een model een error groter dan 0.5 heeft, de tegenovergestelde waarden als resultaat van  $f_m(x)$  in beschouwing worden genomen.

In praktijk is gebleken dat AdaBoost resistent is voor overfitting. De reden hiervoor is nog niet bekend [Friedman, Hastie en Tibshirani (2000)].

## 6.2 Bagging

Bootstrap AGGREGatING methode om een classificatieregels te verbeteren werd geïntroduceerd door [Breiman (1996)], en is geïnspireerd door het meer gekende bootstrap principe. In geval van Bagging wordt de classificatietechniek getraind op verschillende training datasets, waardoor er verschillende modellen ontstaan. De training datasets worden gemaakt aan de hand van de resampling methode. De datasets zijn van dezelfde grootte en

worden willekeurig met teruglegging gekozen. Er worden  $M$  training-datasets van gelijke grootte gecreëerd, door willekeurig objecten uit de training dataset te trekken, mét teruglegging. Op deze  $M$  datasets worden er  $M$  modellen gebouwd. De verschillende modellen leveren verschillende uitkomsten op en op basis van meerderheidsstemming (majority vote) wordt er een keuze gemaakt. In geval van numeriek voorspellen wordt het gemiddelde berekend van de verschillende uitkomsten. Dus Bagging verenigt verschillende uitkomsten van modellen die zijn gebaseerd op verschillende training-datasets, tot één voorspelling. In Voorbeeld 6.2 wordt de classificatie nader uitgelegd.

Bagging kan goed gebruikt worden voor instabiele classificatietechnieken zoals beslisbomen; kleine verandering in de training data genereert een heel ander model, vooral als de training sets klein zijn.

### Voorbeeld 6.2

Objecten	Uitkomst model 1	Uitkomst model 2	Uitkomst model 3	Uitkomst model 4	Uitkomst model 5	Uitkomst model 6	Klasse
Jan	Wel	Wel	Wel	Niet	Niet	Wel	Wel
Klaas	Niet	Wel	Niet	Niet	Wel	Niet	Niet
Piet	Niet	Niet	Wel	Niet	Wel	Niet	Niet
Katrien	Wel	Wel	Niet	Wel	Niet	Wel	Wel

Tabel 6.2: Voorbeeld Majority Vote Bagging

## 6.3 Boosting versus Bagging

In deze paragraaf wordt er nog kort ingegaan op de overeenkomsten van en de verschillen tussen bagging en boosting.

### Verschillen

- Bagging verschilt van Boosting in het genereren van de verschillende modellen. Bij Bagging krijgt ieder model hetzelfde gewicht, terwijl bij Boosting het toekennen van gewichten gebruikt wordt om succesvollere modellen meer invloed te geven.
- Boosting is sequentieel; ieder model wordt beïnvloed door zijn voorgaande model(len). Dit is in tegenstelling tot Bagging, waar de modellen onafhankelijk van elkaar worden gebouwd.
- Bagging reduceert de variantie, terwijl Boosting de zuiverheid van de schatter verbetert. (Voor meer informatie hierover verwijst ik naar [Ridgeway (2002)]).
- Boosting werkt voor stabiele én niet-stabiele classificatietechnieken [Witten en Frank (1999)].
- Bagging werkt voor niet-stabiele classificatietechnieken [Witten en Frank (1999)] en heeft geen toegevoegde waarde bij stabiele classificatietechnieken.

### Overeenkomsten

- Bagging en Boosting maken allebei gebruik van majority vote en numeriek voorspellen. Vanwege het toekennen van gewichten aan de modellen van boosting, werkt Majority vote iets anders (zie paragraaf 6.1).
- In beide gevallen zijn de modellen die gecombineerd worden van *hetzelfde* type.

## 6.4 Boosting en Bagging voor de Postbank

De vraag is nu hoe boosting en bagging toegepast zouden kunnen worden bij het selecteren van klanten voor een marketingactie van de Postbank. Omdat de Postbank voor dit soort selecties voornamelijk gebruik maakt van logistische regressiemodellen en beslisbomen, zal ik boosting en bagging voor deze twee classificatietechnieken bekijken. Bij boosting wordt uitgegaan van het AdaBoost algoritme behandeld in paragraaf 6.2.1 (Figuur 6.1)

### 6.4.1 Boosting

#### Boosting bij Logistische regressie

Bouw, zeg, 10 logistische regressiemodellen gebaseerd op 10 versies ‘gewogen’ training-datasets:

Je begint hiervoor met een training dataset waarvoor de gewichten van de objecten gelijk zijn (Algoritme stap 1).

Een logistisch regressiemodel geeft als resultaat de responskansen van de verschillende klanten binnen de training dataset. Voor het bepalen van de error van het model (Algoritme stap 2) zal de groep van klanten eerst opgesplitst moeten worden in twee groepen, respondenten en niet-respondenten (zie paragraaf 3.3 “De stap naar de praktijk”). Als de error is bepaald, kunnen de gewichten van de objecten aangepast worden om vervolgens een nieuw model te bouwen.

De uiteindelijke klasse kan op verschillende manieren bepaald worden. Hieronder geef ik een paar alternatieven.

#### Alternatief 1

Beschouw per model de geschatte responskansen van de klanten. Na het runnen van de modellen, ontstaat er voor ieder van de 10 modellen een lijst met responskansen voor de klanten in de dataset. Omdat aan ieder model een gewicht is toegekend zal voor de uiteindelijke responskans het gewogen gemiddelde bepaald moeten worden.

#### *Voorbeeld 6.3*

Beschouw een klant uit de dataset. Stel dat de responskansen voor deze klant en de gewichten van de verschillende modellen als volgt zijn:

Model	1	2	3	4	5	6	7	8	9	10
Gewicht	0.3	0.4	0.35	0.5	0.45	0.5	0.6	0.55	0.65	0.6
Responskans	0.9	0.8	0.7	0.85	0.6	0.4	0.75	0.95	0.8	0.8

Tabel 6.3: Alternatief 1: Boosting bij logistische regressie bij de Postbank

De uiteindelijke responskans kan dan als volgt berekend worden:

$$\text{Responskans} = \sum_{i=1}^M \left( \frac{\text{gewicht model } i}{\text{som van de modelgewichten}} \right) * \text{responskans model } i$$

waarbij  $M$  het aantal modellen is.

Voor deze klant is de responskans dus gelijk aan:

Responskans =

$$(0.3/4.9 * 0.9) + (0.4/4.9 * 0.8) + (0.35/4.9 * 0.7) + (0.5/4.9 * 0.85) + (0.45/4.9 * 0.6) + (0.5/4.9 * 0.4) + (0.6/4.9 * 0.75) + (0.55/4.9 * 0.95) + (0.65/4.9 * 0.8) + (0.6/4.9 * 0.8) = \underline{0.78}$$

De responskans kan op deze manier voor iedere klant bepaald worden. En vervolgens kan de totale groep weer opgesplitst worden in twee groepen (respondent en niet-respondent) op de manier zoals besproken in paragraaf 3.3 (“stap naar de praktijk”).

### Alternatief 2

Beschouw per model de klasse waarnaar de klanten zijn verwezen. Na het runnen van de modellen ontstaat er voor ieder van de 10 modellen een lijst met klassen voor de klanten in de dataset. Omdat aan ieder model een gewicht is toegekend zal voor de uiteindelijke klasse het gewogen gemiddelde bepaald moeten worden.

#### *Voorbeeld 6.4*

Beschouw een klant uit de dataset. Stel dat de klassen voor deze klant en de gewichten van de verschillende modellen als volgt zijn:

Model	1	2	3	4	5	6	7	8	9	10
<b>Gewicht</b>	0.3	0.4	0.35	0.5	0.45	0.5	0.6	0.55	0.65	0.6
<b>Klasse</b>	Wel	Wel	Niet	Wel	Niet	Niet	Niet	Wel	Wel	Wel

*Tabel 6.4: Alternatief 2: Boosting bij logistische regressie bij de Postbank*

De som van de gewichten van de modellen die naar “Wel” verwijzen is gelijk aan 3 (= 0.3 + 0.4 + 0.5 + 0.55 + 0.65 + 0.6). De som van de gewichten van de modellen die naar “Niet” verwijzen is gelijk aan 1.9 (= 0.35 + 0.45 + 0.5 + 0.6).

De klant wordt dus uiteindelijk geclassificeerd als respondent.

### Alternatief 3

Stel dat de dataset 100.000 groot is en er moet een mailing gedaan worden van 10.000 groot. Rangschik de klanten per model op basis van hun geschatte responskans. Degene met de hoogste responskans krijgt rangnummer 1 en degene met de laagste responskans krijgt rangnummer 100.000.

Na het runnen van de modellen ontstaat er voor ieder van de 10 modellen een lijst met rangnummers voor de klanten.

#### *Voorbeeld 6.5*

Beschouw een klant uit de dataset. Stel dat de rangnummers voor deze klant en de gewichten van de verschillende modellen als volgt zijn:

Model	1	2	3	4	5	6	7	8	9	10
<b>Gewicht</b>	0.3	0.4	0.35	0.5	0.45	0.5	0.6	0.55	0.65	0.6
<b>Rangnummer</b>	223	1178	1355	1054	75.000	88.522	3358	110	1588	1200

*Tabel 6.5: Alternatief 3: Boosting bij logistische regressie bij de Postbank*



$$\text{Gewogen gemiddelde rangnr.} = \sum_{i=1}^M \left( \frac{\text{gewicht model } i}{\text{som van de modelgewichten}} \right) (\text{rang}_i)$$

waarbij  $M$  het aantal modellen is.

Gewogen gemiddelde Rangnummer =

$$(0.3/4.9 * 223) + (0.4/4.9 * 1178) + (0.35/4.9 * 1355) + (0.5/4.9 * 1054) + (0.45/4.9 * 75.000) + (0.5/4.9 * 88.522) + (0.6/4.9 * 3358) + (0.55/4.9 * 110) + (0.65/4.9 * 1588) + (0.6/4.9 * 1200) = \underline{17015.89}$$

Het rangnummer kan op deze manier voor iedere klant bepaald worden. Selecteer vervolgens de 10.000 klanten met het laagste rangnummer. Deze klanten krijgen dan een mailing toegestuurd.

### **Boosting bij beslisbomen**

Bouw, zeg, 10 beslisbomen gebaseerd op 10 versies gewogen training-datasets volgens het Adaboost algoritme.

Net als bij een logistisch regressiemodel geeft een beslisboom als resultaat responskansen. Iedere eindknoop geeft de kans op respons weer voor klanten die in die knoop belanden. Voor het bepalen van de error van het model en de verschillende gewichten (Algoritme stap 2) zal de groep ook hier eerst opgesplitst moeten worden in twee groepen (zie hoofdstuk 4 “stap naar de praktijk”).

#### Alternatief 1

Beschouw een klant die de verschillende bomen doorloopt. De klant belandt bij iedere boom in een eindknoop die de responskansen weergeeft. In totaal zijn er dus 10 verschillende responskansen voor deze klant. Voor de uiteindelijke responskansen wordt het gewogen gemiddelde genomen. Deze wordt op dezelfde manier berekend als bij logistische regressie.

De responskansen wordt voor iedere klant op deze manier bepaald. Als eenmaal de uiteindelijke responskansen berekend zijn voor de klanten, kan de groep opgesplitst worden in twee groepen zoals beschreven in hoofdstuk 4 (“De stap naar de praktijk”).

#### Alternatief 2

Beschouw een klant die de verschillende bomen doorloopt. De klant krijgt per boom een klasse toegewezen. De uiteindelijke klasse wordt op dezelfde manier bepaald als bij logistische regressie (Alternatief 2).

#### Alternatief 3

Rangschik de klanten. Stel dat er 10 eindknoten in de boom zijn. Afhankelijk van de eindknoop waarin de klant uiteindelijk belandt, krijgt deze een rangnummer toegewezen. In dit geval zullen een aantal klanten hetzelfde rangnummer hebben; er is sprake van ‘ties’ in de rangen. Hier wordt het uiteindelijke rangnummer anders bepaald dan bij logistische regressie. Stel dat er in de eindknoop met de hoogste geschatte responskansen  $m$  klanten zitten. De rang van ieder van de klanten in die knoop is dan gelijk aan het gemiddelde van de rangen 1 tot en met  $m$  ( $= \frac{1}{2}(m+1)$ ).

## 6.4.2 Bagging

### Bagging bij logistische regressie

Er kan aangetoond worden dat voor classificatietechnieken die lineair zijn, zoals logistische regressie, bagging geen verbetering geeft. Bagging bouwt een aantal modellen door de classificatietechniek te trainen op verschillende training datasets. Deze training datasets worden gemaakt aan de hand van de resampling methode.

Zoals eerder werd opgemerkt in dit verslag is logistische regressie een vrij stabiel proces. Een kleine verandering in de training dataset zal geen grote verandering opleveren in model. Indien bagging wordt toegepast op logistische regressie, zullen de verkregen modellen niet veel van elkaar verschillen. Het bouwen van meerdere modellen levert geen toegevoegde waarde op.

### Bagging bij beslisbomen

Bouw, zeg, 10 bomen op verschillende training datasets die zijn verkregen door middel van resampling. Vervolgens bepaalt iedere boom de responskans voor een bepaalde klant. Omdat in geval van Bagging alle bomen gelijke gewichten hebben, is de uiteindelijke responskans voor een bepaalde klant gelijk aan het gemiddelde van de responskansen die verkregen zijn door de verschillende bomen.

Voor de uiteindelijke selectie zou men in dit geval ook gebruik kunnen maken van een variant van alternatief 3 van de vorige paragraaf, waarbij gebruik gemaakt wordt van rangnummers.

## 6.4.3 Boosting en Bagging met SAS Enterprise Miner

De technieken Boosting en Bagging zitten in een applicatie, “Enterprise Miner”, die in SAS gebruikt wordt. Deze applicatie wordt helaas niet gebruikt op de afdeling Customer Intelligence, maar wel binnen de ING bank. Het is mogelijk om deze applicatie ook op de afdeling Customer Intelligence te krijgen.

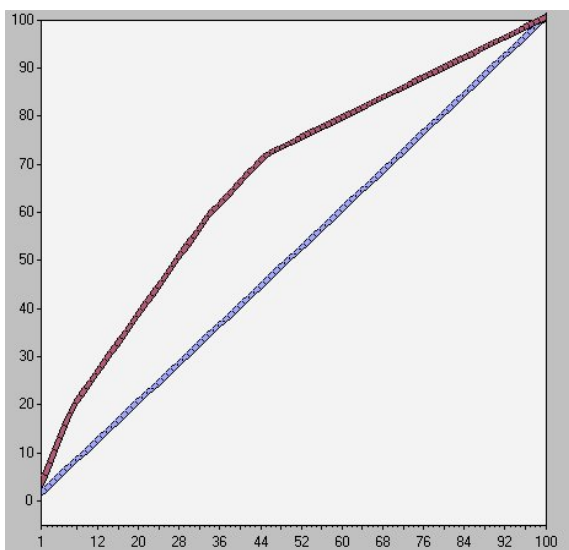
Ik heb zelf geëxperimenteerd met Boosting en Bagging in Enterprise Miner. Hiervoor heb ik een dataset gebruikt met klantnummers waarvan de klasse (0 of 1) bekend is. De dataset is fictief, in die zin dat het niet gaat om een echte marketingactie. De dataset bevat wel echte variabelen uit de klantendatabase en echte klantnummers.

De klassen van de klanten zijn bepaald met een (verzonnen) formule<sup>9</sup>.

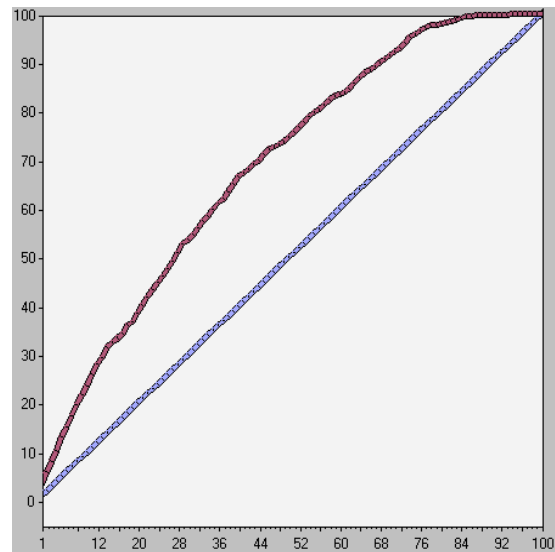
In Enterprise Miner heb ik een boom gebouwd met het CHAID algoritme. Vervolgens heb ik Bagging en Boosting toegepast. De lifcurves zijn weergegeven in onderstaande figuren. Zoals u ziet verschillen de resultaten niet zoveel van elkaar. Bagging en Boosting leveren een iets hogere liftcurve op dan het geval waarin er slechts een enkele boom gebouwd wordt. Het verschil is echter heel klein.

---

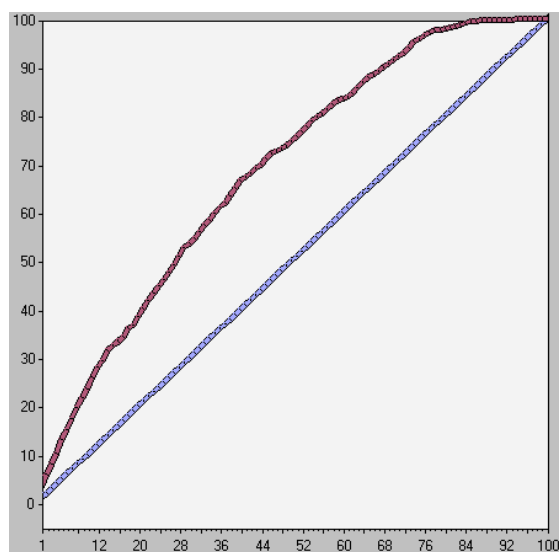
<sup>9</sup> Deze formule heeft mijn begeleider bij ING opgesteld.



*Figuur 6.2 :Liftcurve enkele boom. Een mailing van 44% levert een responspercentage van  $\pm 72\%$*



*Figuur 6.3 :Liftcurve Bagging. Een mailing van 44% levert een responspercentage van  $\pm 75\%$*



*Figuur 6.4 :Liftcurve Boosting. Een mailing van 44% levert een responspercentage van  $\pm 73\%$*

Omdat Bagging en Boosting in dit geval geen succes opleveren, wil niet zeggen dat het Bagging en Boosting niet aan te raden zijn. Er zijn namelijk nog enkele punten waarmee rekening gehouden moet worden bij de resultaten van dit experiment. Het eerste punt is dat ik geen ervaring heb met modelbouw in Enterprise Miner. Dit is een applicatie die ook nog niet op de afdeling CI gebruikt wordt. Verder heb ik geen ervaring met de variabelen. Om een goed model te kunnen bouwen, is kennis van de variabelen nodig. Dit, om onder andere te weten hoe om te gaan met 'missing values'. Een ander punt is dat ik dit experiment in een vrij

korte tijd heb gedaan. Die tijd was niet voldoende omdat ik nog moest leren hoe Enterprise Miner precies werkt.

## 7. Conclusies en Aanbevelingen

### 7.1 Conclusies

In dit onderzoek is er een vergelijking gemaakt tussen de (logistische) regressietechniek en de beslisboomtechniek. Deze technieken worden gebruikt voor het bouwen van modellen die dienen voor het selecteren van klanten bij marketingacties van de Postbank. Het doel van het onderzoek was om aan te geven in welke situaties een regressiemodel beter werkt dan een beslisboom (of andersom), alsmede een verklaring hiervoor.

#### Onderzoek R&M

In eerder onderzoek [Westerlaken, 2003] deed regressie het in een bepaalde situatie beter dan de beslisboom. Het ging hier om een grote doelgroep met weinig respondenten. De reden waarom logistische regressie het beter doet in deze situatie zou kunnen zijn, dat logistische regressie meer informatie tot zich opneemt. Het aantal variabelen dat CHAID en DD gebruiken voor het bouwen van het model is klein. Dit, omdat er vanwege het kleine aantal respondenten weinig onderscheidende subgroepen gevonden worden. De boom kan zich niet verder vertakken en overige variabelen kunnen niet gebruikt worden voor verdere opsplitsing. Het gevolg hiervan is dat hun bijdrage aan informatie verloren gaat. Logistische regressie kan wel meerdere variabelen opnemen en beschikt hierdoor over meer informatie om te classificeren.

#### Algemene richtlijnen

##### Wanneer beslisboomtechniek?

- De beslisboomtechniek is flexibeler in situaties waarin er meerdere manieren zijn voor een object om tot een bepaalde klasse te behoren (zie blz.23 Figuur 5.1).
- Verder kan voor een beslisboom gekozen worden indien de interpreteerbaarheid van belang is. Beslisbomen bestaan uit beslisregels die makkelijk vertaald kunnen worden naar een taal berijpelijk voor de mens.
- De beslisboomtechniek is robuust in geval van uitbijters.
- Een ander voordeel van de beslisboom-techniek is dat deze goed te implementeren is bij de Postbank. Het proces is volledig geautomatiseerd.

##### Problemen bij beslisboomtechniek

- De beslisboomtechniek is een onstabiel proces. Een kleine verandering in de training dataset kan zorgen voor een grote verandering in het model.
- Doordat de beslisboomtechniek de ruimte in meerdere (hyper)rechthoeken opsplijt, komt bij deze techniek het grensgevalprobleem vaker voor.

##### Wanneer regressietechniek?

- De regressietechniek is een stabielere techniek. Een kleine verandering in de training dataset heeft heel weinig effect op het model.

##### Problemen bij regressietechniek

- Een nadeel van de regressietechniek is dat deze niet goed te implementeren is bij de Postbank. Het proces is handmatig.
- De regressietechniek is gevoelig voor uitbijters.

## 7.2 Aanbevelingen

De gegeven reden waarom logistische regressie het beter doet dan de beslisboom in een situatie met een klein aantal respondenten, is slechts een mogelijke verklaring en is niet bewezen. Om een algemene uitspraak te doen dat het inderdaad zo is dat logistische regressie beter werkt in een dergelijke situatie, is verder onderzoek nodig dat gebaseerd is op meerdere gelijksoortige situaties.

Bagging en Boosting zijn technieken om het stabiliteitsprobleem bij de beslisboomtechniek te reduceren. Deze technieken schijnen het in de praktijk uitstekend te doen. Het verdient daarom aanbeveling om te experimenteren met deze technieken. Deze zitten in een applicatie, “Enterprise Miner”, die in SAS gebruikt wordt. Helaas wordt deze applicatie niet gebruikt op de afdeling Customer Intelligence, maar wel binnen de ING bank. Het is mogelijk om deze applicatie ook op de afdeling Customer Intelligence te krijgen.

In deel 2 van dit verslag wordt een tool besproken. Deze tool is een oplossing voor het implementatieprobleem van de regressietechniek bij de Postbank.

De tool zorgt ervoor dat de resultaten van de regressiemodellen, iedere maand automatisch gedraaid worden en meteen op de juiste plaats terechtkomen voor verdere verwerking. Tevens kunnen nieuwe modellen eenvoudig toegevoegd worden aan de tool en kunnen oude modellen eenvoudig verwijderd worden. Het proces is met behulp van deze tool volledig geautomatiseerd. De kans op fouten is hiermee gereduceerd en de onderzoeker heeft meer tijd voor onderzoekswerk.

In dit onderzoek ging het alleen om dichotome afhankelijke variabelen. De beslisboomtechniek en de logistische regressietechniek werken in dit geval beide goed. Gaat het echter om continue afhankelijke variabelen, zoals bijvoorbeeld winstbepaling, dan levert de regressietechniek een zuivere schatting op voor de te voorspellen waarde.

Indien men een zuivere schatting wil hebben van de te voorspellen waarde, is de regressietechniek een betere techniek dan de beslisboom. Is men tevreden met een gemiddelde geschatte waarde, kan de beslisboom ook gebruikt worden.

## Referenties

- [Berry en Linoff (1997)] Michael J.A. Berry en Gordon Linoff. *Data Mining Techniques, For marketing, sales, and Customer support*. Wiley Computer publishing.
- [Berry en Feldman (1985)] William D. Berry en Stanley Feldman. *Multiple regression in practice*. Sage publications.
- [Breen (1985)] Richard Breen. *Regression Models: Censored, Sample Selected, or Truncated Data*, 17-23. Sage Publications.
- [Breiman (1996)] Leo Breiman. *Bagging Predictors*. *Machine Learning*, 24, 123-140.
- [Borra en di Ciaccio (2002)] Simone Borra en Agostino Di Ciaccio. *Improving nonparametric regression methods by bagging and boosting*. *Computational Statistics & Data analysis* 38 (2002) 407-420.
- [Croux en Lemmens (2003)] Christophe Croux en Aurélie Lemmens. *Bagging van statistische classificatieregels*. Artikel, *Business In-Zicht*, nummer 12, maart 2003.
- [Everitt (1977)] Brian S. Everitt. *The Analysis of Contingency Tables*. Chapman and Hall, London.
- [Freund (1995)] Freund, Y. *Boosting a weak learning algorithm by majority*. *Inform. And Comput.* 121, 256-285.
- [Freund en Schapire (1996)] Freund, Y. en Schapire R.E. *Experiments with a new boosting algorithm*. In Saitta, L., editor, *Proc. Thirteenth International Conference on Machine Learning*, Bari, Italy. San Fransisco: Morgan Kaufmann, 148-156.
- [Freund en Schapire (1997)] *A decision-theoretic generalization of online learning and an application to boosting*. *J. Comput. System Sciences* 55.
- [Friedman, Hastie en Tibshirani (2000)] Jerome Friedman, Trevor Hastie en Robert Tibshirani. *Special Invited Paper. Additive logistic regression: a statistical view of boosting*. *The Annals of Statistics* 2000, Vol. 28, No. 2, 337-407.
- [Luger en Stubblefield (1993)] George F. Luger en William A. Stubblefield. *Artificial Intelligence, Structures and strategies for complex problem solving, 2nd edition*. The Benjamin/Cummings Publishing Company, Inc.
- [McCullagh en Nelder (1989)] McCullagh, P., en Nelder, J.A., *Generalized Linear Models* (2nd edition), Chapman and Hall, London.
- [Menard (1995)] Scott Menard. *Applied logistisc regression analysis*, 12-19. Sage Publications.
- [Ridgeway (2002)] Greg Ridgeway. *Looking for Lumps: boosting and bagging for density estimation*. *Computational Statistics & Data analysis* 38 (2002) 379-392.
- [de Rijke (1996)] Angenita de Rijke. *Op zoek naar de respondent*. Verslag stage bij de Postbank voor de studie Bedrijfs-wiskunde & Informatica aan de Vrije Universiteit te Amsterdam.
- [Schapire (1990)] Schapire, R. E. The strength of weak learnability. *Machine Learning* 5, 197-227.

[Seber (1989)] Seber, G.A.F., en Wild, C.J., *Nonlinear Regression*, Wiley, New York.

[Witten en Frank (1999)] Ian H. Witten en Eibe Frank. *Data Mining, Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers.

[Westerlaken (2003)] Claudia Westerlaken. *Vergelijking van modelleertechnieken*. Artikel, ING Nederland / Retail / Customer Intelligence / Research & Modelling.





---

# DEEL 2

Een tool voor het maandelijks  
uitscoren van regressiemodellen

## 1. Inleiding deel 2

In deel 1 van het verslag werden de regressietechniek en de beslisboomtechniek uitgelegd. Deze technieken worden op de afdeling Research & Modelling (R&M) gebruikt om modellen te bouwen die de kans voorspellen dat een klant op een bepaalde mailing zal reageren (de responskans).

Het resultaat van zo een model is een lijst met relatienummers van klanten en de bijbehorende geschatte responskans. De modellen worden gemaakt op de afdeling R&M en het resultaat van de modellen wordt vervolgens verder verwerkt op de afdeling Analytical Campaign Management (ACM) (zie paragraaf 1.2 voor de organisatiestructuur binnen afdeling Customer Intelligence). Hierbij kan gedacht worden aan het koppelen van naam- en adresgegevens aan de relatienummers van de klanten.

Zoals werd vermeld in deel 1, is de implementeerbaarheid van regressiemodellen binnen de Postbank minder goed dan die van beslisbomen. Het uitscoren van beslisboommodellen is volledig geautomatiseerd; de resultaten van de modellen komen meteen op de juiste plaats terecht, zodat deze verder verwerkt kunnen worden op de afdeling ACM. Dit proces bestaat in geval van regressiemodellen uit meerdere “handmatige” handelingen. Het gevolg hiervan is dat voor een regressiemodel al gauw twee dagen nodig zijn om de resultaten op de juiste plaats te krijgen. Het proces is tevens afhankelijk van de onderzoeker die het model heeft gebouwd en de onderzoeker is hierdoor tijd kwijt aan productiewerk. Ook is de kans op het maken van fouten groter bij een handmatig proces. Deze factoren zijn onwenselijk.

Dit deel van het verslag gaat nader in op dit probleem van regressiemodellen bij de Postbank. In hoofdstuk 2 worden de stappen van het handmatig proces beschreven. Er wordt gekeken naar mogelijke oplossingen voor dit probleem in hoofdstuk 3. Vervolgens behandelt hoofdstuk 4 de gekozen aanpak van het probleem. Dit hoofdstuk beschrijft de ontwikkelde programmatuur waarmee logistische regressiemodellen binnen de Postbank automatisch uitgescoord kunnen worden.

## 2. Het handmatig proces

Beslisboommodellen kunnen met behulp van DataDistilleries (DD) automatisch uitgescoord worden. Dit wil zeggen dat als de modellen gedraaid worden, de resultaten meteen op de algemene schijf geplaatst worden. Deze schijf is toegankelijk voor de gehele afdeling Customer Intelligence. De afdeling ACM heeft dan meteen de resultaten tot haar beschikking voor verdere verwerking.

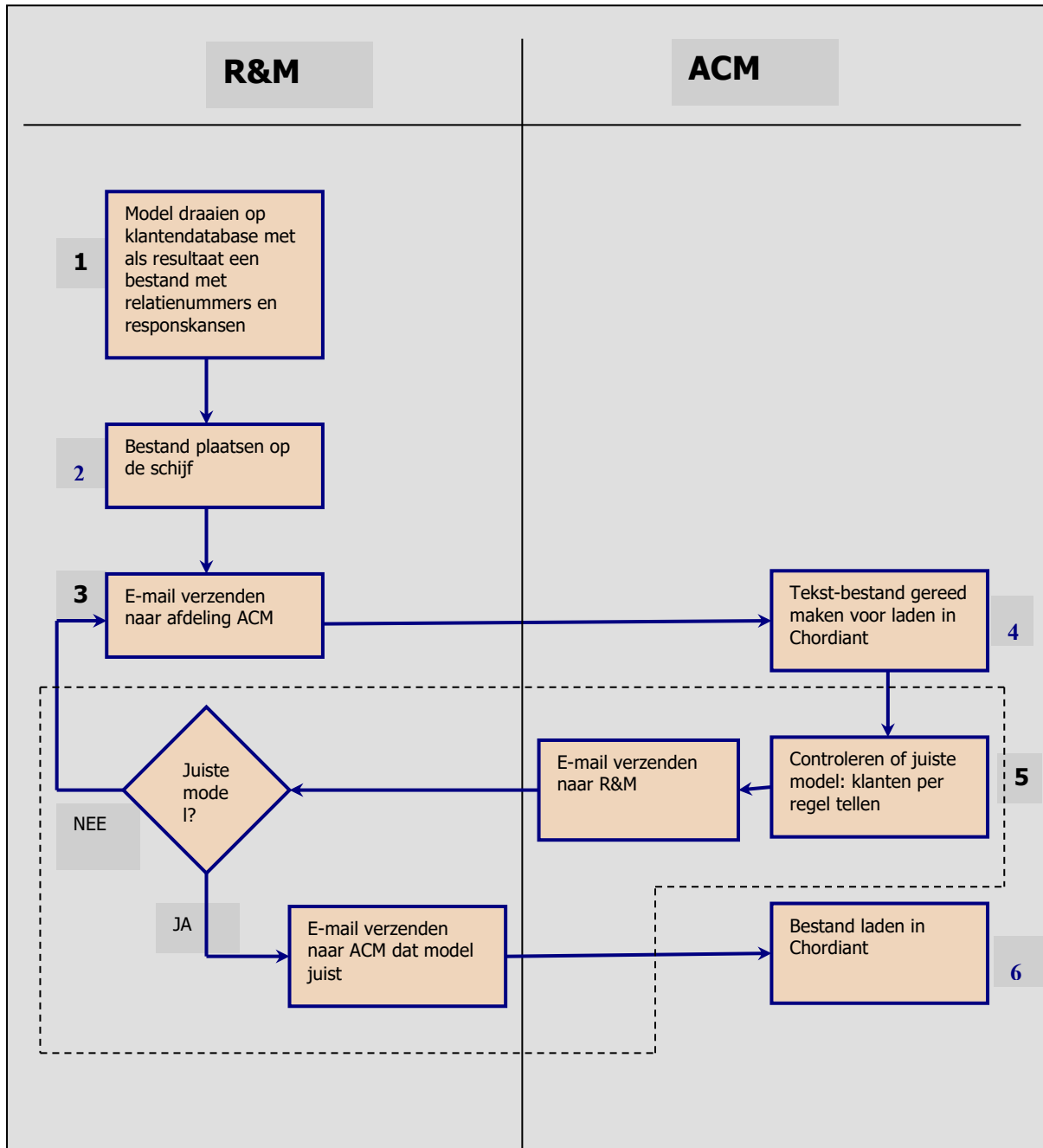
Regressiemodellen kunnen echter niet met DD uitgescoord worden. In plaats hiervan was er in het verleden een programma geschreven dat draaide op “het Mainframe”<sup>10</sup>. Dit programma zorgde er tevens voor dat regressiemodellen iedere maand automatisch uitgescoord werden. Omdat het Mainframe niet meer beschikbaar is, moeten de regressiemodellen tegenwoordig “handmatig” worden uitgescoord. Dit proces gebeurt in de volgende 6 stappen:

1. Het model wordt toegepast op (een deel van) de klantendatabase met als resultaat een bestand met de relatienummers van de klanten en de bijbehorende responskansen.
2. Het bestand moet vervolgens op de algemene schijf geplaatst worden.
3. Er moet een e-mail verzonden worden naar de afdeling ACM met als bericht dat het bestand op de interne schijf staat.
4. Vervolgens moet er op de afdeling ACM een programma gedraaid worden om dit bestand in Oracle of Chordiant te kunnen laden. Oracle en Chordiant zijn programma's waar onder andere data gefilterd kan worden, gegevens zoals naam en adres van verschillende klanten opgevraagd kunnen worden en dergelijke.
5. Het is wel eens voorgekomen dat op de afdeling ACM een verkeerd model werd geladen in Chordiant/Oracle. Er is daarom een controle-stap nodig, om dit incident te voorkomen.
6. Het bestand moet tenslotte geladen worden in Chordiant / Oracle.

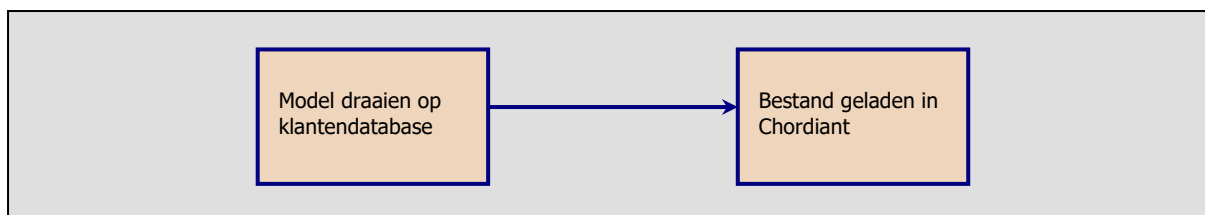
Het zou beter zijn voor de doorlooptijd als na het draaien van een regressiemodel, het bestand met de resultaten meteen op de algemene schijf staat en geladen is in Chordiant of Oracle. De huidige situatie en de gewenste situatie staan in de volgende figuren weergegeven.

---

<sup>10</sup> Mainframe is een computersysteem waarmee er gewerkt werd. Dit systeem is vanaf 2004 niet meer beschikbaar.



Figuur 2.1: Handmatig proces



Figuur 2.2: Gewenste situatie

### 3. Oplossingsanalyse

Het programma dat op het Mainframe gebruikt werd om de modellen maandelijks uit te scoren is gemaakt in SAS-code en is in het bezit van de IT-afdeling.

Het nieuwe systeem “Orion” weet ook met SAS-code om te gaan. Een oplossing voor het implementatieprobleem zou dus kunnen zijn om de SAS-code te kopiëren naar Orion.

Enkele problemen komen hierbij kijken:

- Men is niet helemaal tevreden met het programma:
  - Voor het toevoegen van het model is de IT-afdeling nodig. Het een en ander zoals de naam van het model, nieuwe variabelen enz. wordt dan in het programma erbij getypt. Dit wil je vermijden, want een tyfoutje kan voor een opstopping zorgen binnen het productieproces. Tevens is het inplannen van capaciteit bij de IT-afdeling problematisch; dit veroorzaakt lange wachttijden.
  - Binnen het programma worden er enkele voorselecties gedaan, zoals het selecteren van een leeftijdgroep. Deze selecties kunnen aan de onderzoeker overgelaten worden.
- Het is binnen CI niet precies bekend hoe het maandelijks automatisch uitscoren van regressiemodellen in zijn werk ging.
- De resultaten van de gedraaide modellen kwamen maandelijks automatisch in de Oracle/Chordiant-omgeving te staan, welke toegankelijk is voor de CI-afdeling. Hoe dit precies gedaan werd is niet bekend.

Er kunnen dus 3 stappen onderscheiden worden:

1. Het toevoegen of verwijderen van modellen.
2. Het maandelijks draaien van de modellen.
3. Het plaatsen van de resultaten in de Oracle/Chordiant-omgeving.

In onderstaand figuur is weergegeven hoe deze stappen uitgevoerd werden in de oude situatie op het Mainframe, hoe dit in de huidige situatie wordt aangepakt, en hoe men wenst deze stappen uit te voeren:

	<b>Toevoegen/Verwijderen van modellen</b>	<b>Maandelijks draaien van modellen</b>	<b>Plaatsen van resultaten in Oracle/Chordiant-omgeving</b>
<b>Oude situatie</b>	Uitgevoerd door IT-afdeling	Proces onbekend	Proces onbekend
<b>Huidige situatie</b>	N.V.T	Handmatig op aanvraag (niet maandelijks)	Handmatig
<b>Gewenste situatie</b>	R&M	Automatisch	Automatisch

Figuur 3.1: Oude situatie versus Gewenste situatie

#### Toevoegen/Verwijderen van modellen

In het verleden zijn een aantal problemen ontstaan bij het toevoegen/verwijderen van modellen, welke een opstopping in het doorloopp proces veroorzaakte. Het toevoegen/verwijderen van modellen werd op een andere afdeling, de IT-afdeling gedaan.

Mocht er iets mis gaan, dan is er informatieoverdracht nodig van de R&M-afdeling naar de IT-afdeling. Hierbij is er een kans op miscommunicatie.

Omdat het toevoegen/verwijderen van modellen dicht tegen de kerntaak van de afdeling R&M ligt, wenst R&M deze taak op zichzelf te nemen.

#### Maandelijks draaien van modellen

Er bestaat een tool genaamd PAC die ervoor kan zorgen dat SAS-code maandelijks automatisch gedraaid wordt. Voor het maandelijks automatisch draaien van de modellen zou deze tool dus erg handig zijn. Het probleem is echter dat deze tool op een heel andere afdeling en plaats (Diemen) staat.

#### Plaatsen van resultaten in Oracle/Chordiant-omgeving

Voor het plaatsen van de resultaten in de Oracle/Chordiant-omgeving is toestemming nodig van de IT-afdeling. Omdat de modellen iedere maand gedraaid zullen worden, zullen er iedere maand resultaten opgeleverd worden. Het is natuurlijk niet wenselijk om iedere maand toestemming te vragen voor het plaatsen van deze resultaten in de Oracle/Chordiant-omgeving.

Indien men eenmalig toestemming wil vragen en toch iedere maand nieuwe resultaten wil opleveren, zullen deze resultaten in een datastructuur geplaatst moeten worden die steeds hetzelfde is. Er wordt dan eenmalig toestemming gevraagd voor die datastructuur. De data in de datastructuur kunnen dan wel steeds veranderd worden.

## 4. Aanpak probleem

Het toevoegen en verwijderen van modellen zal gedaan worden met behulp van een tool die gebruikt zal worden op de afdeling R&M. De tool zal een modellenlijst opleveren in de vorm van een tekst-bestand. Dit bestand dient als input voor een programma dat ervoor zal zorgen dat de modellen die in de lijst staan, iedere maand automatisch uitgescoord worden. Hieronder staat een schematische weergave van de aanpak van het probleem gegeven.



Figuur 4.1: Aanpak probleem

### 4.1. De Tool

De tool wordt gemaakt in Excel/Visual Basic. Hiervoor is gekozen omdat deze op alle computers binnen de afdeling CI gebruikt kan worden. Er hoeft dan geen nieuwe programmatuur geïnstalleerd te worden.<sup>11</sup>

#### 4.1.1 Definitie van eisen

Voordat begonnen wordt met het bouwen van de tool, moet eerst met de toekomstige gebruikers nagegaan worden aan welke eisen de tool moet voldoen; welke functies moet de tool beslist hebben, welke functies kunnen handig zijn en welke functies zal de tool niet hebben. Het volgende MoSCoW lijstje resulteerde:

<b>MoSCoW</b>	
<u>Must have</u>	
▪	Model toevoegen/verwijderen
▪	Help functie
<u>Should have</u>	
▪	Gebruiker toevoegen/verwijderen
▪	Wachtwoord en gebruikersnaam
▪	Wachtwoord wijzigen
▪	Gegevens modellen opvragen
▪	Check of model goed is
<u>Could have</u>	
▪	Monitoren
▪	Lijst opvragen van ingelogde gebruikers
<u>Won't have</u>	
▪	Functie om modellen te draaien

Figuur 4.2: MoSCoW-lijstje

<sup>11</sup> Om nieuwe programmatuur te mogen installeren moet deze eerst getest en gekeurd worden. Het kan daarom een hele tijd duren voordat de nieuwe programmatuur gebruikt kan worden.



#### 4.1.2 Gebruikers en functies

Er worden twee soorten gebruikers onderscheiden:

- *Systeembeheerder*: de beheerder van de tool
- *Gebruiker*: degenen die de tool gebruiken voor de modellen

De tool ondersteunt drie soorten functies:

- Algemene functies
- Functies voor de gebruiker van de tool
- Functies voor de systeembeheerder van de tool

De algemene functies kunnen gebruikt worden door de systeembeheerder én de gebruikers. De functies voor de systeembeheerder kunnen alleen door de systeembeheerder gebruikt worden. En de functies voor de gebruiker kunnen alleen door de gebruikers gebruikt worden.

#### Algemene functies

Login	
Beschrijving:	Om het hoofdmenu / systeembeheerdersmenu binnen te komen, zal de gebruiker twee velden moeten invullen, namelijk een gebruikersnaam en een bijbehorend wachtwoord. Of men in het hoofdmenu of het systeemmenu terechtkomt, hangt af van de gebruikersnaam en het wachtwoord.
Input:	Gebruikersnaam en bijbehorend wachtwoord.
Verwerking:	De gebruikersnaam wordt in de gebruikerslijst opgezocht en kijkt of het ingevuld wachtwoord bij de gebruiker hoort.
Output:	<ul style="list-style-type: none"><li>▪ Als de gebruikersnaam niet in de lijst voorkomt of als het wachtwoord niet bij de gebruikersnaam hoort, dan wordt er een foutmelding gegeven.</li><li>▪ Als alles correct is, komt de gebruiker, afhankelijk van de opgegeven gebruikersnaam en het bijbehorend wachtwoord, óf in het hoofdmenu, óf in het systeembeheerdersmenu terecht.</li></ul>

Logout	
Beschrijving:	Om het hoofdmenu of het systeembeheerdersmenu te verlaten, zal de gebruiker moeten uitloggen.
Input:	Klik op de log out knop.
Verwerking:	Hoofdmenu / Systeembeheerdersmenu wordt gesloten.
Output:	Het hoofdmenu wordt gesloten en het login scherm komt te zien.

## Functies voor gebruiker

<b>Model toevoegen</b>	
Beschrijving:	Deze optie wordt gebruikt indien de gebruiker een model wil toevoegen aan de modellenlijst om maandelijks automatisch uitgescoord te worden.
Input:	De naam en plaats van het model (bestand).
Verwerking:	<ul style="list-style-type: none"> <li>▪ De naam en plaats van het model (bestand) worden opgeslagen in de modellenlijst.</li> <li>▪ De nieuwe modellenlijst wordt weggeschreven naar tekst-bestand.</li> </ul>
Output:	De gebruiker krijgt een melding dat het model is toegevoegd aan de lijst

<b>Model verwijderen</b>	
Beschrijving:	Deze optie wordt gebruikt indien de gebruiker het niet meer nodig vindt dat een bepaald model maandelijks automatisch uitgescoord wordt. Het model wordt dan met deze functie verwijderd uit de modellenlijst.
Input:	De naam en plaats van het model.
Verwerking:	<ul style="list-style-type: none"> <li>▪ De naam en plaats van het model (bestand) worden verwijderd uit de modellenlijst.</li> <li>▪ De nieuwe modellenlijst wordt weggeschreven naar tekst-bestand.</li> </ul>
Output:	De gebruiker krijgt een melding dat het model is verwijderd uit de lijst

<b>Modellenlijst opvragen</b>	
Beschrijving:	Deze optie wordt gebruikt indien de gebruiker gegevens van een bepaald model wil opvragen. Deze gegevens zijn: <ul style="list-style-type: none"> <li>▪ Naam</li> <li>▪ Plaats</li> <li>▪ Datum dat model is toegevoegd in de lijst</li> <li>▪ Naam van gebruiker door wie het model is toegevoegd</li> </ul>
Input:	De naam van het model (wordt geselecteerd uit de lijst)
Verwerking:	De gegevens worden gekoppeld aan het opgegeven model
Output:	Gegevens van het model wordt weergegeven.

Help-functie	
Beschrijving:	Deze functie wordt gebruikt indien men informatie wil hebben over de functies (voor gebruikers) binnen de tool.
Input:	Een onderwerp waarover men informatie wil (het onderwerp wordt geselecteerd uit een lijst).
Verwerking:	De tekst behorende bij het opgegeven onderwerp wordt opgezocht.
Output:	De tekst behorende bij het opgegeven onderwerp wordt weergegeven.

### Functies voor systeembeheerder

Gebruiker toevoegen	
Beschrijving:	De systeembeheerder gebruikt deze optie om een nieuwe gebruiker te creëren. Er zijn 5 velden die hiervoor om in te vullen. Het invullen van de velden “gebruikersnaam”, “wachtwoord” en “Bevestig wachtwoord” is verplicht. De overige velden “Naam” en “Locatie” mogen ook ingevuld worden, maar zijn niet verplicht.
Input:	Gebruikersnaam, Wachtwoord en Bevestig wachtwoord (Naam en Locatie).
Verwerking:	<ul style="list-style-type: none"> <li>▪ Er wordt nagegaan of gebruikersnaam al voorkomt in de lijst.</li> <li>▪ Er wordt nagegaan of het wachtwoord overeenkomt met de bevestiging.</li> <li>▪ De gebruiker wordt opgeslagen in de gebruikerslijst.</li> </ul>
Output:	<ul style="list-style-type: none"> <li>▪ Indien de gebruikersnaam al voorkomt in de lijst wordt er een foutmelding gegeven</li> <li>▪ Indien het wachtwoord niet overeenkomt met de bevestiging wordt er een foutmelding gegeven.</li> <li>▪ Indien alles correct is, krijgt de systeembeheerder een melding dat de gebruiker is toegevoegd.</li> </ul>

Gebruiker verwijderen	
Beschrijving:	De systeembeheerder gebruikt deze functie om een bestaande gebruiker te verwijderen uit de gebruikerslijst. De betreffende gebruiker heeft dan geen toegang meer tot de tool.
Input:	Gebruikersnaam (deze wordt geselecteerd uit een lijst)
Verwerking:	De gebruiker met de opgegeven gebruikersnaam wordt verwijderd uit de gebruikerslijst.
Output:	De systeembeheerder krijgt een melding dat de gebruiker is verwijderd.

Gebruikersgegevens opvragen	
Beschrijving:	Deze optie wordt gebruikt indien de systeembeheerder gegevens van een gebruiker wil opvragen. Deze gegevens zijn: <ul style="list-style-type: none"><li>▪ Naam</li><li>▪ Locatie</li><li>▪ Gebruikersnaam</li><li>▪ Wachtwoord</li></ul>
Input:	De gebruikersnaam (wordt geselecteerd uit de lijst).
Verwerking:	De gegevens worden gekoppeld aan de opgegeven gebruikersnaam.
Output:	Gegevens van de gebruiker wordt weergegeven.

Help-functie	
Beschrijving:	Deze functie wordt gebruikt indien de systeembeheerder informatie wil hebben over de functies (voor de systeembeheerder) binnen de tool.
Input:	Een onderwerp waarover men informatie wil (het onderwerp wordt geselecteerd uit een lijst).
Verwerking:	De tekst behorende bij het opgegeven onderwerp wordt opgezocht.
Output:	De tekst behorende bij het opgegeven onderwerp wordt weergegeven.

### 4.1.3 GUI's

Als de eisen / wensen zijn vastgesteld en nagedacht is over de verschillende functies, kan begonnen worden met het uiterlijk van de tool. Hieronder volgen de schermen (GUI's) in de tool.

Bij het opstarten van de tool komt allereerst het login scherm te zien, zodat de gebruiker kan inloggen.





Verander wachtwoord

Gebruikersnaam:

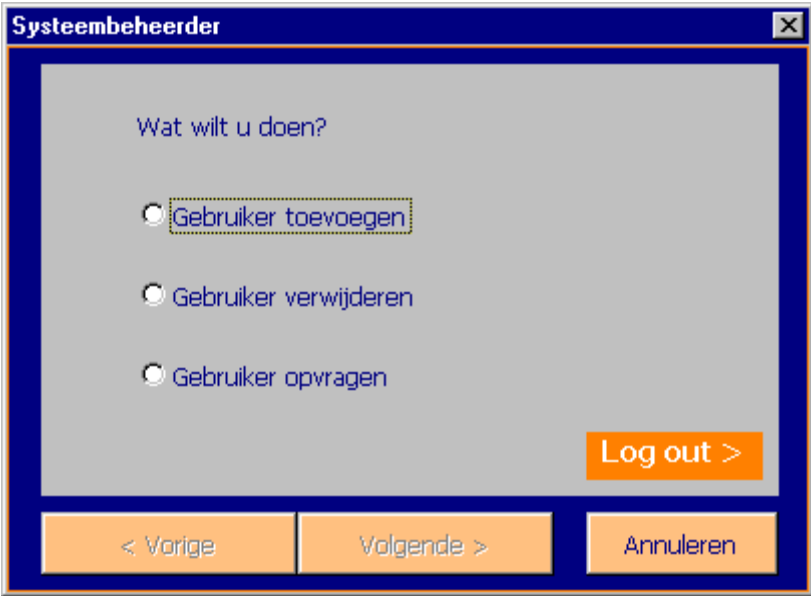
Oud wachtwoord:

Nieuw wachtwoord:

Bevestig nieuw wachtwoord:

OK Annuleren Help

Indien de systeembeheerder inlogt, komt het volgende scherm te zien:



Systeembeheerder

Wat wilt u doen?

Gebruiker toevoegen

Gebruiker verwijderen

Gebruiker opvragen

Log out >

< Vorige Volgende > Annuleren

De systeembeheerder kan vervolgens een gebruiker toevoegen aan het systeem, een gebruiker verwijderen of de gegevens van een gebruiker opvragen.

Om een gebruiker toe te voegen moeten alle tekstvelden ingevuld worden, behalve de tekstveld "Afdeling", deze is niet verplicht.



**Gebruiker toevoegen**

Voer gegevens van nieuwe gebruiker in: \*) Niet verplicht

Naam:

Afdeling:\*

Gebruikersnaam:

Wachtwoord:

Bevestig wachtwoord:

Log out >

< Vorige      Toevoegen      Help

Een gebruiker kan verwijderd worden door deze te selecteren in de lijst “Selecteer gebruiker(s)” en deze te plaatsen in de lijst “Verwijder gebruiker(s)”. Vervolgens moet op de knop “Verwijderen” geklikt worden om de gebruikers in de lijst “Verwijder gebruiker(s)” te verwijderen. Indien niet geklikt wordt op de knop “Verwijderen” worden de gebruikers teruggezet in de gebruikerslijst.



**Gebruiker verwijderen**

Selecteer gebruiker(s):      Verwijder gebruiker(s):

ysmkarg  
ykgarg

>>

<<

Log out >

< Vorige      Verwijderen      Help

In de GUI “Gebruiker Opvragen” kan de systeembeheerder de gegevens van de gebruiker opvragen waaronder ook het wachtwoord.

**Gebruiker opvragen**

Selecteer gebruiker: ysmkarg  
ykarg

Gegevens gebruiker:  
Naam: Yonina  
Afdeling: R&M  
Wachtwoord: yyyyyy

Geef gegevens weer

Log out >

< Vorige      Volgende >      Help

Indien men onder een andere gebruikersnaam inlogt, komt men in het hoofdmenu, waarbij er gekozen kan worden tussen model toevoegen, model verwijderen, gegevens van model opvragen of monitoren.

**Hoofdmenu**

Wat wilt u doen?

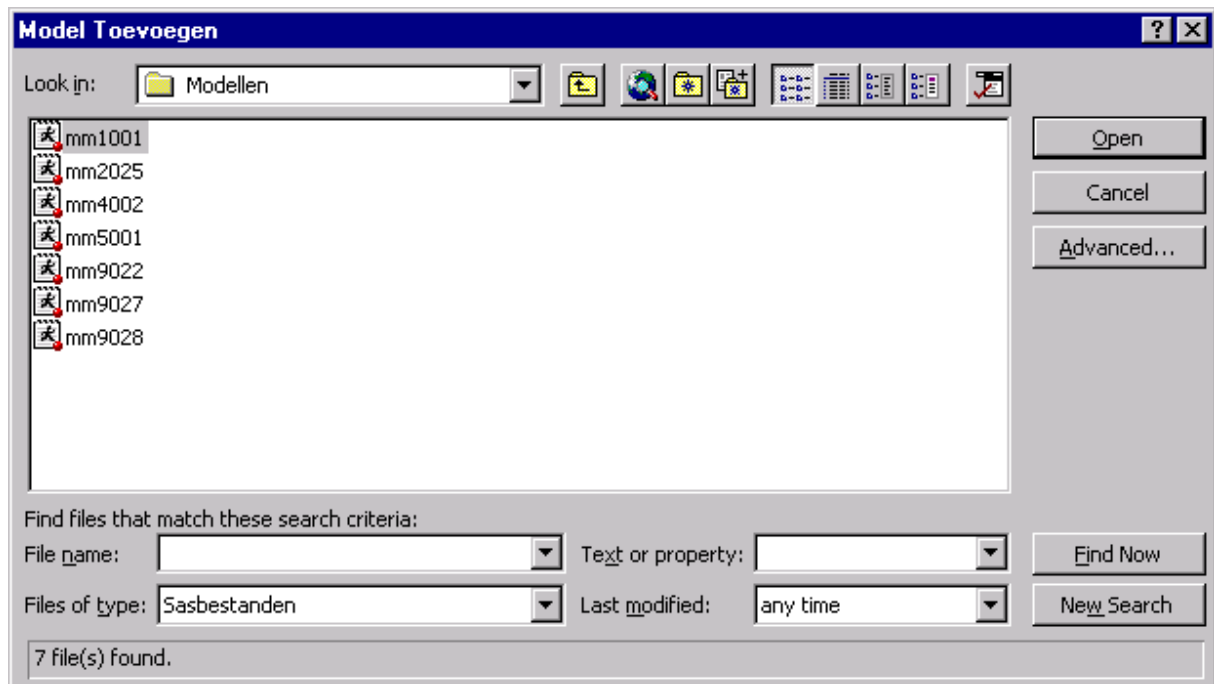
Model toevoegen  
 Model verwijderen  
 Modellenlijst opvragen

ING  
Retail/ Customer intelligence/  
Research & Modelling

Log out >

Hoofdmenu      OK      Help

Bij het toevoegen van een model komt het bekende “Open bestand”-scherm te zien. Hieruit kan dan het bestand geselecteerd worden die toegevoegd moet worden aan de lijst.

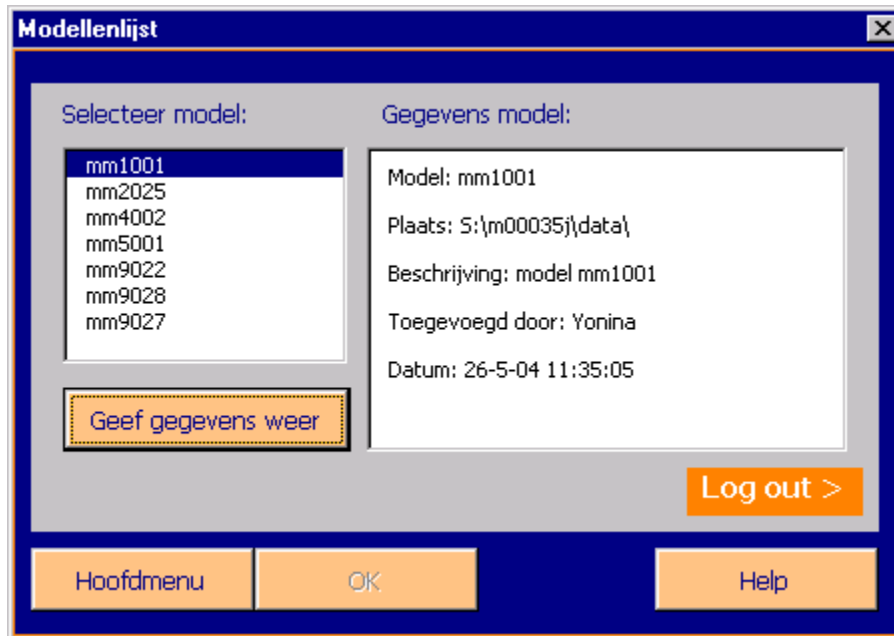


Een model kan verwijderd worden door deze te selecteren in de lijst “Selecteer model(len)” en deze te plaatsen in de lijst “Verwijder model(len)”. Vervolgens moet op de knop “Verwijderen” geklikt worden om de modellen in de lijst “Verwijder model(len)” te verwijderen. Indien niet geklikt wordt op de knop “Verwijderen” worden de modellen teruggezet in de modellenlijst.





De gegevens van de modellen kunnen opgevraagd worden in de GUI “Modellenlijst”. Hier kan onder andere nagegaan worden door wie en wanneer een bepaald model is toegevoegd aan het systeem.



#### 4.1.4 Enkele opmerkingen

- De tool draait alleen onder Windows.
- Voor het gebruik van de tool is Microsoft Excel nodig. Versie 97 of hoger.
- Uit voorzorgsmaatregelen ondersteunt het systeem slechts één gebruiker te gelijk.
- Het systeem bevat aanvankelijk 0 gebruikers. De eerste gebruiker van het systeem wordt meteen de systeembeheerder.

#### 4.2. Het tekstbestand

De tool levert een tekstbestand op. Het tekstbestand bestaat uit een aantal regels. Op iedere regel staan de volgende gegevens:

- Naam van het model
- Naam van het model
- Plaats van het model
- Beschrijving van het model

De gegevens worden door een komma van elkaar gescheiden. Een voorbeeld van het uiterlijk van het tekstbestand:

```
mm9027.sas,mm9027,S:\m00035\data\,mailing product A  
mm1001.sas,mm1001,S:\m00035\data\,mailing product B  
mm2025.sas,mm2025,S:\m00035\data\,mailing product C  
mm4002.sas,mm4002,S:\m00035\data\,mailing product X  
mm5001.sas,mm5001,S:\m00035\data\,mailing product Y  
mm9022.sas,mm9022,S:\m00035\data\,mailing product Z
```

### 4.3. Het programma

Het programma is geschreven in SAS-code en zal in Diemen gedraaid worden onder de tool PAC. Er is gekozen voor SAS-code omdat de regressiemodellen in SAS-code zijn gemaakt. Ondanks dat de tool PAC op een andere afdeling en plaats staat, is het vanwege het netwerksysteem mogelijk dat de resultaten op de algemene schijf van Customer Intelligence geplaatst worden.

De hoofdtaken van het programma zijn:

1. Het uitscoren van de modellen
2. Het omzetten van de resultaten naar een geschikte datastructuur voor Chordiant/Oracle.

#### *Het uitscoren van de modellen*

In het tekstbestand wordt gelezen welke modellen uitgescoord moeten worden. De modellen worden één voor één uitgescoord.

#### *Omzetten van de resultaten naar geschikte datastructuur*

Omdat bekend is dat de datastructuur van Datadistilleries geschikt is voor de omgeving van Chordiant/Oracle, is gekozen voor een soortgelijk structuur.

De resultaten worden geplaatst in Oracle-tabellen. Deze kunnen door zowel Chordiant als Oracle geopend worden.

### De datastructuur

Het resultaat van het uitscoren van modellen wordt weergegeven in drie tabellen:

- Action
- Segment
- Score.

#### Action

Deze tabel bevat de acties (modellen) die zijn uitgevoerd. Deze tabel bestaat uit vier kolommen:

- *ID* – identiteit van de actie
- *Action* – Naam van de actie
- *Beschrijving* – Beschrijving van de actie
- *Timestamp* – Datum en tijd van uitvoering van de actie

Kolommen	Type	Mogelijke lengte
ID	Numeriek	1 – 2
Action	Standaard tekst	6
Beschrijving	Standaard tekst	0-100
Timestamp	Standaard tekst	8

Tabel 4.1: Beschrijving Action-tabel

### Segment

Deze tabel bestaat uit drie kolommen:

- *ID* – identiteit van het segment
- *Action* – naam van de actie waartoe het segment behoort
- *Regel* – De regel behorende bij de actie

Kolommen	Type	Mogelijke lengte
ID	Numeriek	1 - 4
ActionID	Standaard tekst	6
Regel	Numeriek	1 - 2

Tabel 4.2: Beschrijving Segment-tabel

### Score

Deze tabel bestaat uit twee kolommen:

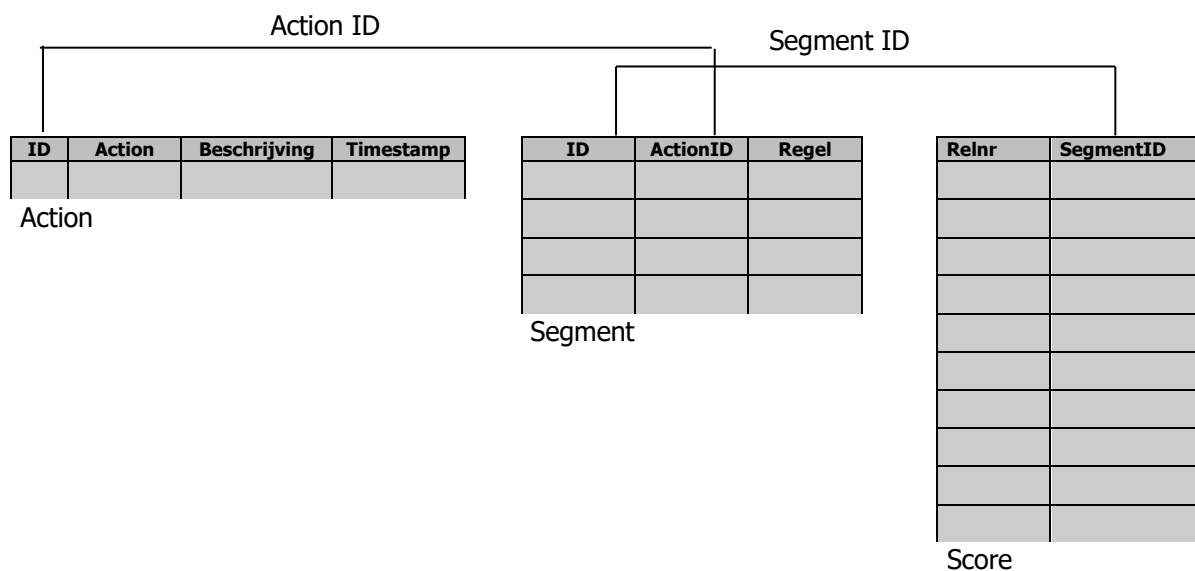
- *Relnr* – relatienummer van de klant
- *SegmentID* – Het segment waartoe het relatienummer behoort.

Kolommen	Type	Mogelijke lengte
Relnr	Numeriek	1 - 20
SegmentID	Numeriek	1 - 4

Tabel 4.3: Beschrijving Score-tabel

### Verband tussen de tabellen

Het verband tussen de tabellen wordt weergegeven in onderstaand figuur.



### Voorbeeld

Stel dat de modellen MM1001, MM2001 en MM301 uitgescoord worden en dat de resultaten als volgt zijn:

MM1001

Relatiennr.	Score
1	0
2	1
3	0

MM2001

Relatiennr.	Score
1	0
2	0
3	1

MM3001

Relatiennr.	Score
1	0
2	1
3	1

De Action-tabel wordt dan als volgt:

ID	Action	Beschrijving	Tijd
1	MM1001	Mailing product X	
2	MM2001	Mailing product Y	
3	MM3001	Mailing product Z	

In dit voorbeeld heeft ieder model twee soorten regels: 1 of 0. De segment-tabel wordt dan:

SegmentID	ActionID	Regel
1	1	0
2	1	1
3	2	0
4	2	1
5	3	0
6	3	1

Vervolgens kan de score-tabel opgesteld worden. Per regel wordt er nagegaan welke klant hieraan voldoet. In dit voorbeeld zijn er in totaal 6 regels:

1. Bij ActionID = 1 hoort het model MM1001.  
Klanten van model MM1001 met score 0 → relatiennr. 1 en 3
2. Bij ActionID = 1 hoort het model MM1001.  
Klanten van model MM1001 met score 1 → relatiennr. 2
3. Bij ActionID = 2 hoort het model MM2001.  
Klanten van model MM2001 met score 0 → relatiennr. 1 en 2
4. Bij ActionID = 2 hoort het model MM2001.  
Klanten van model MM2001 met score 1 → relatiennr. 3
5. Bij ActionID = 3 hoort het model MM3001.

Klanten van model M3001 met score 0 → relatiernr. 1

6. Bij ActionID = 3 hoort het model MM3001.  
 Klanten van model M3001 met score 1 → relatiernr. 2 en 3

Relatiernr.	SegmentID
1	1
3	1
2	2
1	3
2	3
3	4
1	5
2	6
3	6

### Stappen Sas-programma

#### 1. Init

- Verwijder de oude SAS-tabellen (Action, Segment en Score)
- Verwijder de oude Oracle-tabellen (Action, Segment en Score)
- Importeer het tekstbestand

#### 2. Uitscoren

De modellen worden één voor één uitgescoord. Ieder model levert een tabel op bestaande uit 2 kolommen: relatienummer en Score.

#### 3. Datastructuur

- Maak Action-tabel
- Maak Segment-tabel
- Maak Score-tabel

#### 4. Oracle-tabellen

Zet de tabellen, Action, Segment en Score om naar Oracle-tabellen.

Figuur 4.3. Stappen Sas-programma

## 4.4. Eisen van het model

Er zijn enkele eisen waaraan een model moet voldoen voordat het aan het systeem toegevoegd mag worden:

- Een model kan pas toegevoegd worden als het in een bepaalde directory zit. Alle modellen die aan het systeem worden toegevoegd moeten namelijk in één directory zitten. Hiervoor is gekozen, zodat er een beter overzicht is van de modellen.
- De naam van de aangemaakte dataset moet hetzelfde zijn als de naam van het model (zonder de extensie). Voorbeeld:
  - Naam model: mm0001.sas
  - Naam dataset: mm0001
- De naam van het model bestaat uit 6 tekens, waarvan de eerste 2 gelijk zijn aan “mm” en de overige 4 cijfers zijn. Geldige namen zijn dus bijvoorbeeld:
  - mm0000
  - mm1010
  - mm2341
- De regels die worden opgeleverd door het model zijn numeriek.
- Het is niet nodig om de maandstand te bepalen. Wel moet de variabele “maandstand” meegegeven worden.
  - TijdID = &maandstand
- Het model levert een tabel op bestaande uit twee kolommen: een kolom met relatienummers en een kolom met de scores. De naam van de kolom met relatienummers is *Relnr* (verplicht) en de naam van de kolom met de scores is variabel.