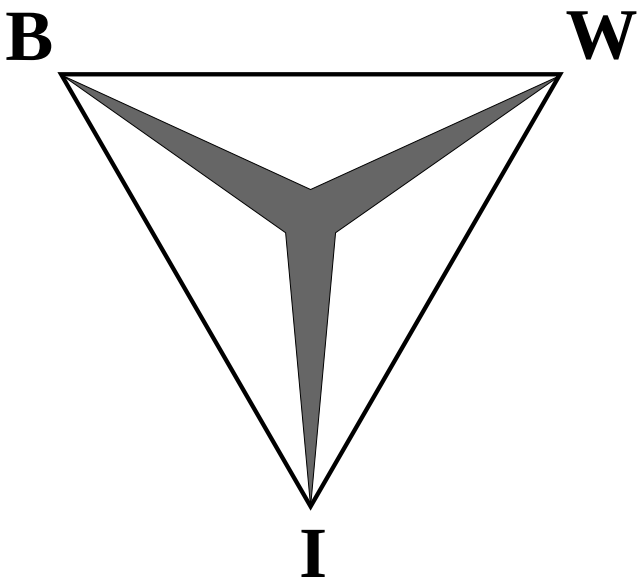
Risico modellering voor IT-projecten

Het kwantificeren van risico met wiskundige technieken



Vrije Universiteit
Faculteit der Exacte Wetenschappen
Divisie Wiskunde en Informatica
De Boelelaan 1081 1081 HV Amsterdam

Joeri van Hoeve, BWI ’96
10 Augustus 2004

Begeleider: Bert Kersten

Voorwoord

In de laatste fase van de studie Bedrijfswiskunde en Informatica (BWI) aan de Vrije Universiteit dient de student een BWI werkstuk te schrijven. Dit werkstuk dient een (literatuur)onderzoek te zijn, waarbij voldoende aandacht moet worden besteed aan de drie belangrijke deelgebieden van de studie. Het onderzochte probleem moet dus zowel een redelijke Bedrijfsgerichte component hebben als wel als een Wiskundige en Informatica component.

Dit werkstuk is in grote mate geïnspireerd door mijn afstudeerstage bij ING Corporate IT[1] te Amsterdam. Mijn onderzoek richtte zich daar op het ontwikkelen van modellen om de kans op budgetoverschrijding, tijdsoverschrijding en onvoldoende opgeleverde functionaliteit te voorspellen van een bepaald IT project. Binnen ING worden projecten, waarbij tenminste 25% van het budget aan IT wordt uitgegeven, als IT projecten beschouwd. In dit onderzoek was ik door de kleine dataset en soort verzamelde projectgegevens (projectkenmerken en -risico's) beperkt in mijn keuze van modelleertechniek. Met dit literatuuronderzoek heb ik dan ook beoogd om een overzicht te geven van meerdere kwantitatieve technieken en aan te geven in welke situaties deze technieken geschikt zijn een bepaald risico van IT projecten te kwantificeren.

Bij deze wil ik graag Bert Kersten bedanken voor zijn vlotte begeleiding en hulp bij het schrijven van dit werkstuk. Dit werkstuk is voor mij de laatste stap in deze studie en ik wil hierbij dan ook met name mijn ouders en broer bedanken, die me in sommige moeilijke periodes in de afgelopen 8 jaar volledig hebben gesteund.

[1] IT is de afkorting van informatie technologie

Samenvatting

Dit werkstuk is bedoeld als een soort handleiding voor het gebruik van voorspelmodellen uit de statistiek en datamining bij het kwantificeren van risico bij IT projecten.

Wij hebben daarvoor een algemene aanpak geïntroduceerd voor het bepalen van een gekwantificeerd risico van een project. Hierbij zal eenduidig moeten worden gedefinieerd, wat onder een *risicovol project*[2] wordt verstaan, zodat deze projecten in bepaalde risicocategorieën kunnen worden ingedeeld. In de volgende stap gebruiken we voorspelmodellen als *logistische regressie* of een *classificatie boom*, die de kans bepalen dat een project tot een bepaalde risicocategorie hoort. De belangrijkste voorwaarde voor het gebruik van deze modellen is een zorgvuldige indeling in risicocategorieën; de modellen zijn zinloos als projecten in meerdere klasses kunnen zitten. Het is verder belangrijk te kijken naar de aanwezige projectkenmerken, die immers de input van het model vormen. Als de kenmerken voornamelijk uit categorieën bestaan, dan betekent dit dat een *classificatie-boom* een beter model is dan *logistische regressie*. Vervolgens moeten we kiezen hoe we afwijking van de gewenste uitkomst willen kwantificeren. De afwijking is altijd een continue waarde en deze kunnen we dus voorspellen met bijvoorbeeld een *lineair model*.

De beschreven modellen in dit werkstuk kunnen in theorie dus gebruikt worden om elk risico te kwantificeren. De voorbeelden, die we in dit werkstuk hebben gegeven waren gericht op IT projecten. De algemene aanpak van risicomodellering, die we beschreven hebben is natuurlijk ook geschikt voor andere niet-IT projecten. Een belangrijke constatering met betrekking tot risicomodellering bij IT projecten is het feit, dat zeer geschikte technieken als *logistische regressie* of *rough data models* nog niet worden toegepast. Het zal dus interessant zijn om deze technieken in de praktijk te gaan toepassen. Een andere uitdaging is het formuleren van eenduidigere definities van ongewenste uitkomsten van IT projecten, zodat de uiteindelijke risicovoorspellingen nog zinvoller worden.

[2] Een risicovol project wordt ook wel opgevat als een project met ongewenste uitkomsten.

Inhoudsopgave

Hoofdstuk 1

Inleiding

Informatie technologie (IT) is voor grote multinationale organisaties een van de grootste kostenposten en tegelijk ook een van de belangrijkste produktie factoren. Zo'n tien jaar geleden werd IT voornamelijk beschouwd als administratieve ondersteuning, terwijl IT tegenwoordig een belangrijke rol speelt in de hele waardeketen van een bedrijf. Als we bijvoorbeeld kijken naar IT uitgaven van grote mondiaal opererende banken, dan zien we dat deze kosten zeer hoog zijn. Volgens [14] worden deze kosten voor Nederlandse banken op 20 tot 22% van de totale operationele kosten geschat. De belangrijkste bedrijfsprocessen, zoals bijvoorbeeld het automatische betalingsverkeer en het beheer van financiële produkten worden helemaal of voor een groot deel ondersteund door IT. Het grote strategische belang van IT onderstreept het nut van risicobeheersing bij investeringen in IT projecten binnen een organisatie.

Deze risicobeheersing is hard nodig, omdat *de meeste directeuren IT uitgaven beschouwen als een zwart gat: ongeacht hoeveel aandacht ook er wordt besteed aan IT, een duidelijke rechtvaardiging voor de uiteindelijke opbrengsten is niet aanwezig* [24, p. 2]. In 1996 nam de Amerikaanse overheid haar maatregelen en kwam zij met de Clinger Cohen Act[1]. Deze wet verplicht overheidsinstellingen tot een portfolio benadering van hun IT projecten. Deze benadering houdt in dat er een balans moet worden gevonden tussen het verwachte risico en de verwachte opbrengst van een of meerdere projecten. Een belangrijke voorwaarde voor deze portfolio benadering is het bepalen van het risico van ieder individueel project. De doelstelling van dit werkstuk is om een overzicht te geven van enkele beschikbare kwantitatieve voorspelmodellen, die bruikbaar zijn om het risico in te schatten van een IT project.

[1] Clinger Cohen Act: Deze wet uit 1996 geeft richtlijnen aan de DOD's (afdelingen van defensie) en aan andere overheidsinstellingen over de aanpak van acquisitie en management van IT

Hoofdstuk 2

Vraagstelling

De hoofdvraag in dit werkstuk luidt als volgt:

- Welke voorspelmodellen zijn geschikt om het risico van een IT project te kwantificeren?

Alvorens we deze vraag kunnen gaan beantwoorden zullen we allereerst moeten bepalen wat verstaan wordt onder een *risico*vol *IT* project. Vooral de term *IT* wordt heel algemeen gebruikt en daarom kunnen we onze probleemstelling aan de hand van de volgende deelvragen beantwoorden:

- Wat verstaan we onder een IT project?
- Wat verstaan we onder een risicovol project?
- Welke voorspelmodellen worden (in de literatuur) toegepast op IT projecten?

De eerste twee deelvragen zullen aan bod komen in het volgende hoofdstuk om zo de lezer wat meer inzicht te geven in IT projecten en de bijbehorende risico's. In Hoofdstuk 3 beginnen we met een algemene uitleg van het hele modelleer proces en introduceren we ook een aantal wiskundige technieken. De derde deelvraag komt aan bod in Hoofdstuk 4 en zal ingaan op enkele gepubliceerde voorspelmodellen, die toegepast worden op IT projecten. Tenslotte besluiten we met de conclusies en aanbevelingen in Hoofdstuk 5.

Hoofdstuk 3

IT projectrisico's

In dit hoofdstuk zullen we ingaan op wat we precies verstaan onder een *risicovol IT project*. We geven allereerst een algemene definitie van *IT project* en gaan ook in op de definitie van een *risico*. Vervolgens behandelen we in Sectie 2.2 specifieke voorbeelden van risicovolle IT projecten.

3.1 Algemene definities

We willen allereerst komen tot een algemene definitie van een IT project en daartoe gaan we apart in op de termen *IT* en *project*. Een project wordt gedefinieerd in [16, p. 89]:

- Een *project* is een verzameling van ongeroutineerde activiteiten en hun onderlinge relaties, die bedoeld zijn om een specifiek doel te bereiken.

We zien dat in plaats van de term IT de termen ICT[1] en IS[2] ook veelvuldig in de literatuur verschijnen. We zien de volgende definities voor *IT (ICT)* en *IS* in respectievelijk [29] en [30].

1. *IT of ICT* is de technologie, die benodigd is voor het verwerken van informatie. In het bijzonder het gebruik van computers en computer software voor het converteren, opslaan, bewerken, verzenden en ontvangen van informatie.

2. *IS* is een combinatie van computer hardware, software en communicatie technology, die is ontworpen om informatie te verwerken voor een of meerdere gerelateerde bedrijfsprocessen.

De definitie voor IS (2) gaat wat dieper in op de belangrijke technologie componenten en benadrukt duidelijk de verschillende componenten van deze technologieën en ook het uiteindelijke doel van de informatieverwerking. De definitie voor IT/ICT (1) beschrijft het informatieverwerkingsproces wat uitgebreider en dus komen we nu zelf tot een *vrije* definitie van een IT project:

[1] Informatie en Communicatie Technologie
[2] Informatie Systeem

- Een *IT project* is een verzameling van ongeroutineerde activiteiten en hun onderlinge relatie om uiteindelijk een combinatie van computer hardware, software en communicatie technology te gebruiken voor het converteren, opslaan, bewerken, verzenden en ontvangen van informatie ten behoeve van een of meerdere gerelateerde bedrijfsprocessen.

In het woordenboek wordt *risico* gedefinieerd als:

- *gevaar voor schade of verlies*
- *de gevaarlijke of kwade kans of kansen die zich bij iets voordoen*

Deze definities zijn niet eenduidig. We zien namelijk dat een risico zowel opgevat kan worden als een "kans op een gevaar" als op het "gevaar" zelf. Deze dubbelzinnige eigenschap in het woord risico wordt ook nog eens geïllustreerd aan de hand van Boehm, die in [4] de zogenoemde *risk exposure* (RE) introduceert. Deze RE wordt gedefinieerd door de volgende relatie:

$$RE = P(UO) * L(UO) \tag{3.1}$$

De P(UO) staat voor de kans op een onbevredigende uitkomst van het project en L(UO) staat voor het verlies voor de betrokken partij bij een onbevredigende uitkomst van het project. De risk exposure is dus duidelijk een voorbeeld van een gekwantificeerd risico. De absolute hoeveelheid risico bij een risicovol project wordt weergegeven door L(UO) De term verlies is een beetje verwarrend, omdat dit doet vermoeden dat de L(UO) altijd in geld wordt uitgedrukt, terwijl dit bijvoorbeeld ook kan staan voor de tijdsoverschrijding, We kunnen deze L(UO) dan ook beter beschouwen als een bepaalde negatieve afwijking van de gewenste situatie. Verder komen er bij een project vaak meerdere ongewenste situaties voor. We passen daarom de formule (3.1) aan om te komen tot $RE_i(A)$ die het totale gekwantificeerde risico (met A als maat) weergeeft van project i, waarbij we k onafhankelijk van elkaar optredende ongewenste situaties hebben:

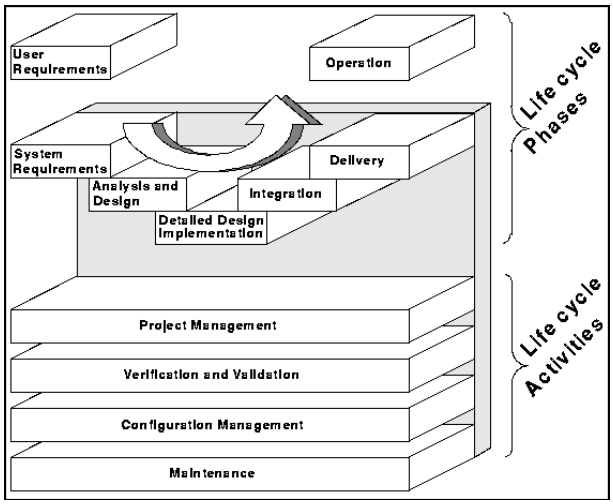
$$RE_i(A) = \sum_{j=1}^{k} (P_i(UO_j) * A_i(UO_j)) \tag{3.2}$$

De twee componenten van ons uiteindelijke risico, $P_i(UO_j)$ en $A_i(UO_j)$ moeten afzonderlijk worden bepaald aan de hand van projectkenmerken van elke afzonderlijk project. We zullen onze vraagstelling om te kijken welke voorspelmodellen geschikt zijn om het risico van een IT project te kwantificeren uit Hoofdstuk 2 dan ook in twee delen moeten opsplitsen:

- Welke voorspelmodellen zijn geschikt om de kans op een of meerdere ongewenste uitkomsten van een IT project te bepalen?
- Welke voorspelmodellen zijn geschikt om de afwijking van een of meerdere gewenste uitkomsten van een IT project te bepalen, gegeven dat het feit dat een ongewenste uitkomst optreedt?

3.2 Risicovolle IT projecten

We gaan nu in op enkele voorbeelden van risicovolle IT projecten en merken op dat IT projecten grofweg te verdelen zijn in *software*[3] en *infrastructuur*[4] projecten. We gaan nu vooral in op *software* projecten, omdat deze projecten het meest risicovol zijn van karakter volgens [14]. We kunnen een software project het best illustreren aan de hand van de zogenoemde software levencyclus. Figuur 3.1 uit [15] geeft een goed overzicht van deze cyclus met de bijbehorende fases en activiteiten.



Figuur 3.1: Overzicht van fases en activiteiten in de software levenscyclus.

We zullen niet ingaan op alle technische aspecten en risico's in deze fases. Voor een gedetaileerde beschrijving van elke fase verwijzen we de lezer dan ook graag naar de meer specifieke literatuur (o.a. [23], [2], [19]). Het belangrijkste is dat we duidelijk zien, dat er een kop en staart aan het project zit: Aan de hand van de *gebruikerseisen* kan het projectplan gemaakt worden en daarna start de ontwikkelingscyclus[5]. Na oplevering naar tevredenheid van de gebruiker gaat de software in gebruik en in deze *operationele fase* wordt de software onderhouden. Dit onderhoud is geen nieuw project, maar een activiteit zoals het projectmanagement. Als de gebruikerseisen veranderen gedurende de operationele fase en het software produkt wordt aangepast, dan spreken we van een nieuw project (software vernieuwing). De software projecten kunnen dus als individuele projecten worden

[3] Hiermee bedoelen we op maat gemaakte software oplossingen en geen licenties voor standaard softwarepakketten.

[4] De IT infrastructuur wordt in [5] opgevat als de funderingen voor IT van een bedrijf en bestaat uit gekoppelde en standaard IT diensten, waaronder hardware, netwerken, opslag netwerken, middleware en databases.

[5] We noemen het een cyclus, omdat de fases soms ook meerdere keren kunnen worden doorlopen.

beoordeeld, omdat er een duidelijk begin en eind is aan het project. Deze afbakening is nodig om te kunnen vaststellen of een individueel project risicovol is (zowel qua kosten of verwachte opbrengsten).

Wij illustereren het gebrek aan een duidelijk begin en een eind bij *infrastructuur* projecten aan de hand van een citaat uit [13, p. 244]; *infrastructuur projecten zijn beoordeeld als losse projecten onrendabel , maar creëren wel mogelijkheden voor rendabele vervolg investeringen.* Deze opvatting wordt gedeeld door een onderzoek van Weill [25], die vier categorieën van IT investeringen geeft en daarbij constateert, dat in de meest succesvolle bedrijven de meeste kosten uitgaan naar de *infrastructuur* projecten om vervolgens te investeren in de drie overige soorten *software* projecten, die zorgen voor de opbrengsten.

Wij weten nu dus dat we ons hoofdzakelijk moeten richten op software projecten en we zullen dan ook ingaan op een softwarerisico aan de hand van enkele definities van enkele onbevredigende uitkomsten van een software project. In [4] onderscheidt men verschillende onbevredigende uitkomsten van een project vanuit het perspectief van vier deelnemers (de klant, ontwikkelaar, gebruiker en onderhoudspersoon):

- Voor klanten en ontwikkelaars zijn overschrijdingen van budget en het tijdsschema het meest onbevredigend.
- In de ogen van gebruikers zullen projecten gefaald zijn als er een produkt wordt opgeleverd met verkeerde functionaliteit of met tekortkomingen in de interface.
- Een slechte kwaliteit van de software zal voor onderhoudspersonen een zeer ongewenst uitkomst van een project zijn.

De Standish Group is een onderzoeksbedrijf, dat door middel van intensieve interviews en enquetes met grote bedrijven (top 500 Fortune lijst) o.a. een top tien publiceert van de succesfactoren van software-ontwikkelings projecten. Zij onderscheiden de volgende typen van uitkomsten van projecten in [22]:

- Succesvol: Het project is afgerond binnen de geplande tijd en het geplande budget en heeft alle kenmerken en functies zoals van te voren was afgesproken
- Uitgedaagd: Het project is afgerond en operationeel, maar heeft het geplande budget en de originele tijdsschatting overschreden. Ook zijn er minder kenmerken en functies opgeleverd dan afgesproken was
- Mislukt: Het project is gestopt voordat het af was of is nooit geimplementeerd.

De twee laatste definities zijn duidelijk voorbeelden van ongewenste uitkomsten vanuit het zicht van een projectmanager. We kunnen ook vanuit een financiële manager kijken en zien dan dat [10] de Netto Contante Waarde (NCW) methode noemt als voornaamste indicator van projectsucces. Bij deze methode wordt op basis van verwachte kosten en baten in de toekomst de huidige waarde van het project berekend. Een project wordt in [10, p.255] als succesvol beschouwd, *als het is opgeleverd met de hoogst haalbare NCW en met de gewenste kwaliteit*. We kunnen nu deze definitie omschrijven naar bijvoorbeeld een zeer ongewenste uitkomst van een project vanuit het financiële oogpunt:

- een project is opgeleverd tegen een lagere NCW dan de hoogsthaalbare en met een mindere kwaliteit dan gewenst.

We zien dus duidelijk dat er genoeg verschillende ongewenste uitkomsten zijn. Wij zijn hier niet uitputtend in geweest en hebben ons met name gericht op de wat algemenere projectmanagement risico's. In ieder geval hebben we laten zien dat het met name van belang is vanuit welk gezichtspunt in een bedrijf naar een project wordt gekeken. Verder zien we ook dat de meeste definities van ongewenste uitkomsten zich beperken tot de oplevering van het project. Deze fase is natuurlijk een heel belangrijke mijlpaal in een software project en het is mooi als het project tot zover succesvol is met betrekking tot de NCW. Als we de hoogsthaalbare NCW echter werkelijk willen halen, dan zullen we in de lange operationele fase van een project niet alleen naar ongewenste situaties moeten kijken vanuit het gezichtspunt van de software onderhouders, maar ook vanuit het financiële oogpunt.

Hoofdstuk 4

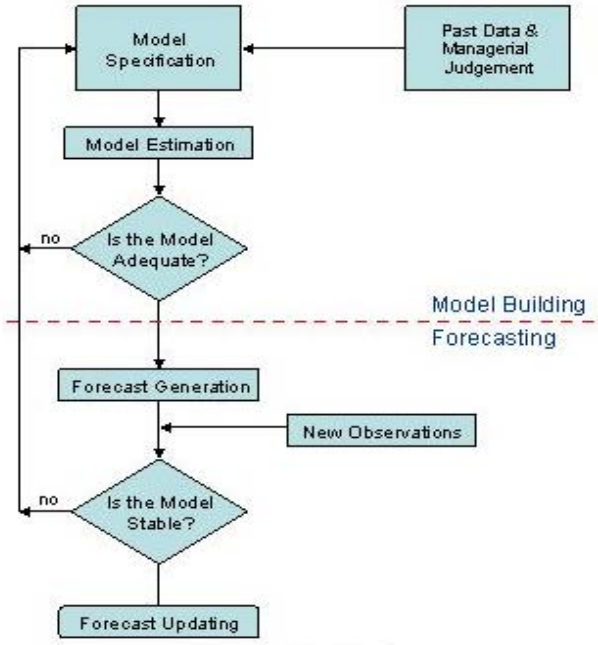
Voorspelmodellen

Dit hoofdstuk introduceert een aantal modelleertechnieken op het gebied van de statistiek en datamining, die geschikt zijn om een voorspellend risicomodel mee te ontwikkelen. We hebben ons in dit werkstuk beperkt tot technieken die relaties of patronen in de data beschrijven, waarbij we ook uitermate geïnteresseerd zijn in een expliciete representatie van deze patronen. We hebben daarom een eventueel zeer bruikbare techniek als *neurale netwerken* buiten beschouwing gelaten, omdat deze voorspellingen maken op basis van een veel moeilijker te interpreteren model. Dit is niet wenselijk bij risico voorspelmodellen. We zijn namelijk niet alleen geïnteresseerd in de voorspelde risico's, maar ook in de onderliggende oorzaken van deze risico's, zodat preventieve maatregelen kunnen worden genomen. In sectie 4.1 geven wij informatie over de modelbouw en bijbehorende statistische termen, die daaropvolgende secties 4.2-4.6 inzichtelijker maken maken. In deze secties zullen we enkele modelleertechnieken introduceren.

4.1 Enige achtergrondinformatie

In Figuur 4.1 uit [1] wordt op de volgende pagina een mooi raamwerk gegeven, waarin een duidelijk overzicht wordt geven van verschillende fases en acties, die moeten leiden tot een degelijk voorspelmodel. We zien, dat de eerste fase bestaat uit het hebben van data. In ons geval verstaan we onder data de projectkenmerken en het geobserveerde projectrisico. Bij het wiskundig modelleren worden deze projectkenmerken de *verklarende* variabelen genoemd; het te voorspellen projectrisico wordt ook wel de *te verklaren of respons*variabele genoemd. Deze beide variabelen kunnen vervolgens worden onderverdeeld volgens [12] in twee soorten variabelen; een *continue variabele* representeert quantitatieve data met een continue verzameling waarden en een *categorische variabele* representeert kwalitatieve data en is discreet. Dit betekent dat alleen bepaalde vaststaande numerieke of niet numerieke gegevens kunnen worden aangenomen. Deze beide verzamelingen zijn niet waterdicht. Als we kijken naar een variabele, die het aantal fouten weergeeft , dan kunnen we deze variabele niet opvatten als continu (1.5 fout is niet mogelijk);

de variabele is ook niet geen categoriaal, want bestaat niet uit een vaststaand aantal waarden. Wij zullen deze variabele gewoon opvatten als een continue variabele.



Figuur 4.1: Overzicht van fases en acties bij het bouwen van een voorspelmodel

Na een gedegen samenvatting[1] van de projectgegevens kunnen we het model *specificeren*. Hierna moeten we projectgegevens indelen in een trainings- en een testset, waarbij idealiter de verdeling van het aantal projecten in beide sets 75% om 25% is. Vervolgens *schatten* we het model op basis van de gegevens in de trainingsset. In de volgende fase moeten we kijken of het model wel voldoende *kwaliteit* heeft. Deze kwaliteit bepalen we door aan de hand van de testset voorspellingen te doen en te kijken in hoeverre deze voorspellingen juist zijn geweest. Als deze kwaliteit laag is dan moeten we het model opnieuw specificeren door bijvoorbeeld andere verklarende variabelen in het model op te nemen of zelfs een andere modelleertechniek te gebruiken. In het andere geval hebben we de bouwfase van het model afgerond en kunnen we met het model gaan voorspellen. We gaan het model nu in de praktijk toepassen op nieuwe projecten. Als de kwaliteit van deze nieuwe voorspellingen achteraf niet zo goed blijken te zijn dan moet het hele proces opnieuw doorlopen worden. We zullen in de volgende sectie ingaan op een aantal voorspelmodellen aan de hand van de *modelspecificatie*. Verder geven we informatie over de *schattingsmethode* en gebruikte *kwaliteitsmetrieken*.

[1]het samenvatten van variabelen bestaat uit het bepalen van de verdelingen en verder uit het bepalen van afhankelijkheden tussen bepaalde variabelen. We verwijzen graag naar [7] voor een goed overzicht hiervan.

4.2 Lineaire regressie

Dit model is de meest simpele vorm van regressie analyse en de eerste gepubliceerde vorm van lineaire regressie was de *kleinste-kwadratenmethode*; deze werd voor het eerst gepubliceerd door Legendre in 1805 [26], alhoewel Gauss heeft geclaimd dat hij de methode al kende in 1795. In een lineair regressie model wordt verondersteld dat een continue respons variabele lineair afhangt van de waarden van een of meerdere verklarende variabelen. Deze sectie is voornamelijk gebaseerd op [7].

Modelspecificatie

We beschouwen n willekeurige observaties van een continue responsvariabele Y_i, die afhangen van n observaties van p verklarende variabelen x_{ij} met $i = (1\ldots,n)$ en $j = (1,\ldots,p)$:

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \tag{4.1}$$

De willekeurige variabele ε_i wordt ook wel de meetfout genoemd en wordt verondersteld normaal verdeeld te zijn:

$$\begin{aligned} E\varepsilon_i &= 0, \\ E\varepsilon_i\varepsilon_j &= \sigma^2 \qquad i,j = 1,\ldots,n \end{aligned} \tag{4.2}$$

De onbekende waarden in formule 4.1 zijn de β-parameters of -coëfficiënten en de meetfouten ε_i. Allereerst zullen we nu deze β's moeten schatten. Deze coëfficiënten geven immers de invloed van de verklarende variabelen op de responsvariabele weer.

Schattingsmethode

We beschrijven de kleinste-kwadratenmethode, omdat we hebben verondersteld, dat de meetfouten normaal verdeeld zijn (4.2) en deze methode in dat geval goed werkt. We proberen nu de waarden van de β's te vinden door de volgende kwadraatsom te minimaliseren:

$$SS(\beta) = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{4.3}$$

We moeten nu dus een stelsel van p afgeleiden oplossen om de minimale kwadraatsom te vinden:

$$\sum_{i=1}^{n} \frac{\delta(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})}{\delta\beta_i} (Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij}) = 0, \quad j = 1,\ldots,p. \tag{4.4}$$

De gevonden kleinste-kwadraten schatter $\hat{\beta}$ uit het stelsel vergelijkingen van (4.4) leidt tot de zogenoemde *residuele kwadraatsom* RSS:

$$RSS = SS(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) = \sum_{j=1}^{n} (Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij})^2 \tag{4.5}$$

In het geval dat de meetfouten niet normaal zijn verdeeld kunnen we beter niet de kleinste-kwadraten methode gebruiken. Een mogelijkheid is het gebruik van robuuste schatttingsmethoden. In [12] worden enkele van deze methodes weergegeven als wel als een uitgebreide referentielijst van onderliggende theorie.

Kwaliteitsmetriek

Deze RSS vormt dus de basis voor een beoordeling van de kwaliteit van ons model. We kunnen de variantie van ons model schatten op basis van de RSS:

$$\hat{\sigma}^2 = \frac{RSS}{(n-p-1)} \tag{4.6}$$

Met behulp van deze variantie kunnen we de variantie van de $\hat{\beta}$'s berekenen, die ons een indicatie geven van de nauwkeurigheid van de betreffende schattingen. Aan de hand van de variantie kunnen we namelijk een boven- en ondergrens voor de schattingen bepalen. Voor uitgebreide informatie over deze zogenoemde betrouwbaarheidsintervallen verwijzen we u naar [8]. Een belangrijke kwaliteitsindicatie voor het gehele model is de determinatiecoefficient R^2:

$$R^2 = \frac{SS_{nulmodel} - RSS}{SS_{nulmodel}} \tag{4.7}$$

De $SS_{nulmodel}$ staat voor de kwadraatsom van het model zonder de verklarende variabelen en de R^2 staat dus voor de fractie van de door het gehele model verklaarde variantie. De R^2 heeft waarden tussen nul en één en het model met de grootste waarde is het beste model.

Verdere informatie

We hebben voor dit model aangenomen dat p onafhankelijke verklarende variabelen lineair van elkaar afhangen. Deze aannames dienen dan ook vooraf gecheckt te worden. Wij verwijzen graag naar [7], waarin technieken worden besproken om zowel verdelingen van variabelen als de onderlinge samenhang tussen deze variabelen vast te stellen. Een andere belangrijk onderdeel is het bepalen van de te gebruiken verklarende variabelen. In [7] worden de toetsen besproken voor het al dan niet invoegen of weglaten van een verklarende variabele in een model. Verder gaat men hier in op enkele technieken om uitbijters (uitschieters of opvallende afwijkende waarnemingen) in de verklarende variabelen te detecteren en zo het model beter te maken.

4.3 Niet-lineaire regressie

Volgens [6] zijn er veel situaties waarbij op basis van bijv. theoretische beschouwingen en praktisch bewijzen een lineair model niet geschikt wordt bevonden. In dit geval beschouwen we een model waarbij de verwachting van een observatie niet afhangt van een niet-lineaire functie van onbekende parameters. Deze functie is soms bekend, bijvoorbeeld als de relatie tussen de responsvariabele en verklarende variabelen wordt beschreven door een bekende natuurkundige wet. In veel gevallen zullen we een zo flexibel mogelijke functie moet kiezen om ons optimale model te kunnen vinden. We bespreken daarom een algemene vorm van het niet-lineaire model. Deze sectie is gebaseerd op [6].

Modelspecificatie

We beschouwen opnieuw n observaties van de respons Y_i en van de p verklarende variabelen, die we noteren als in een vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$:

$$Y_i \quad = \quad f(x_i, \theta) + \varepsilon_i, \quad \textit{met } \theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$$

De ε_i staat weer voor de i-de meetfout genoemd, maar reflecteert eigenlijk het verschil tussen de geobserveerde responsvariabele en de niet-lineaire functie met daarin de meetfout. De verwachting van ε_i wordt weer nul verondersteld en de bijbehorende onbekende variantie wordt beschreven door ${\sigma_i}^2$.

Schattingsmethode

Het schatten van de onbekende parameters gebeurt volgens [6] net als in het lineaire model door de kleinste-kwadraten schatter te bepalen door de kwadraatsom te minimaliseren.

$$S(\theta) = \sum_{i=1}^{n} (Y_i - f(x_i, \theta))^2 \tag{4.8}$$

$$\sum_{i=1}^{n} \frac{\delta f}{\delta \theta_j}(x_i, \theta)(Y_i - f(x_i, \theta)) = 0, \quad j = 1, \ldots, p. \tag{4.9}$$

We kunnen de kwadraatsom (4.8) minimaliseren door een stelsel van p afgeleiden (4.9) op te lossen. Een bekende iteratieve procedure om deze vergelijkingen op te lossen is de iteratieve *Gauss-Newton* methode. Voor meer specifieke informatie over deze methode verwijzen we de lezer naar [6].

Kwaliteitsmetriek

We hebben geen algemene kwaliteitsmetriek, zoals de R^2 bij het lineaire model, omdat we geen nulmodel (geen verklarende variabelen) kunnen vormen. De alge-

hele modelkwaliteit wordt nu dus puur weergegeven door de residuele kwadraatsom (RSS). We kunnen dus analoog aan het lineaire model *geneste*[2] modellen met elkaar vergelijken met behulp van de zogenoemde F-toetsen. Bij deze toets kijkt men of het model met meer parameters wel echt een duidelijk lagere RSS heeft. Als dit niet het geval is, dan geven we dus de voorkeur aan een kleiner model (met minder parameters).

Ook kunnen we met behulp van de betrouwbaarheidsintervallen weer de kwaliteit van de in het model opgenomen verklarende variabelen checken. De betrouwbaarheidsintervallen geven de onder- en bovengrens weer van alle geschatte parameters. Een parameter met een betrouwbaarheidsinterval met de waarde nul geeft aan dat de verklarende variabele niet echt een duidelijke invloed heeft en deze variabele kan dan beter worden weggelaten. Voor een uitgebreidere beschrijving van het hele niet-lineaire modelleerproces verwijzen we de lezer naar [6].

[2]Bij geneste modellen bevat het grotere model (met meer variabelen) altijd in ieder geval alle variabelen van het kleinere model.

4.4 Gegeneraliseerde Lineaire Modellen (GLM)

We hebben in de vorige secties geconstateerd dat aan de specifieke voorwaarden van lineaire regressie en niet-lineaire modellen niet vaak wordt voldaan en daarom introduceren we hier een algemeen raamwerk voor een klasse van modellen, die ook geld voor niet-lineaire en niet-normale modellen. Deze zogenoemde *gegeneraliseerde lineaire modellen* (GLMs) werden in 1972 geïntroduceerd door Nelder and Wedderburn. In deze sectie zullen we een overzicht geven van de mogelijkheden van verschillende glm's, maar we zullen vooral ingaan op logistische regressie, omdat deze techniek een binaire responsvariabele heeft en veel gebruikt is in medische studies om de kans te bepalen op het hebben van een bepaalde ziekte.

Modelspecificatie

We gaan weer het uit van n observaties van de respons Y_i en van de p verklarende variabelen $x_i j$ met $i = (1,\ldots,n)$ en $j = 1,\ldots,p)$. Het algemene model van de GLM's ziet er nu als volgt uit:

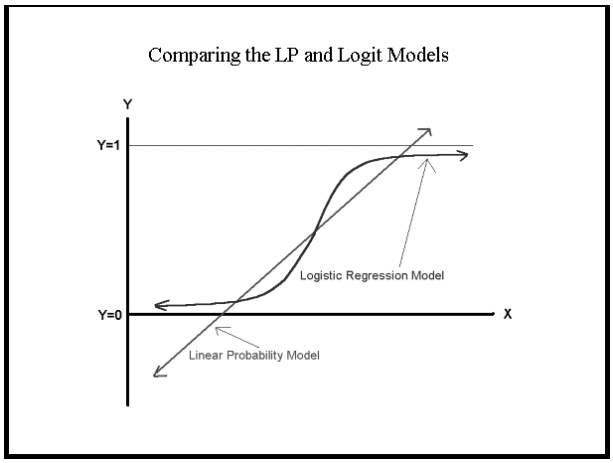
$$\begin{aligned} &(i) && Y_i \; heeft \; de \; kansdichtheids functie \; f_i \\ &(ii) && \eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \\ &(iii) && \eta_i = g(\mu_i) \end{aligned} \tag{4.10}$$

Kansverdeling	**Linkfunctie**	**Variantie**
Normaal - Gauss	μ	1
Binomiaal (logit)	$log(\mu/(1-\mu))$	$\mu(1-\mu)/n$
Poisson (probit)	$log(\mu)$	μ
Gamma	$1/\mu$	μ^2
Inverse Normaal - Gauss	$1/\mu^2$	μ^3
Quasi	$g(\mu)$	$V(\mu)$

Tabel 4.1: Overzicht GLM's per verdeling van de respons variabele

In (4.10) wordt het model ingedeeld in een *random* component(i), die staat voor de kansverdeling van de responsvariabele en een *systematische* component (ii), die weergeeft dat de te verklaren variabelen in het model voorkomen als een lineaire combinatie. De *linkfunctie* (iii) g geeft de verbinding weer tussen de systematische

en de random component. De linkfunctie transformeert de uitkomst van de random component, de verwachting van de responsvariabele μ_i in dezelfde vorm als de systematische component η_i. In [12] worden een aantal gegeneraliseerde modellen onderscheiden. We geven in Tabel 4.1 een aantal kansverdelingen weer met de bijbehorende linkfunctie en variantie.



Figuur 4.2: Verschil van logistisch regressie met lineaire regressie.

We zullen nu in het bijzonder even ingaan op het voorspellen van een binomiale responsvariabele, die dus gebruik maakt van de *logit* link functie. We zien in Figuur 4.2 duidelijk, dat een lineair model niet geschikt is voor het verklaren van deze binaire respons. Dit specifieke GLM wordt ook wel aangeduid met *logistische regressie* en is ideaal om responsvariabelen met weinig informatie te voorspellen. We beschrijven dit logistische regressie model nu in de GLM notatie (4.10):

$$\begin{aligned} (i) \quad & n_i Y_i \sim Bin(n_i, \mu_i), \\ (ii) \quad & \eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_i, \\ (iii) \quad & \eta_i = g(\mu_i) = log[\mu_i/(1-\mu_i)] \end{aligned} \tag{4.11}$$

We weten dat een binomiale verdeling met parameters n_i voor het aantal waarnemingen en $\mu_i = pr_i$ voor de kans op een succes (of andersom gedefinieerd een mislukking) de volgende verwachting en variantie van de responsvariabele Y_i heeft:

$$E(Y_i) = pr_i \quad Var(Y_i) = \phi \frac{pr_i}{1-pr_i} \tag{4.12}$$

Voor deze algemene uitleg nemen van aan dat de *schaal* parameter ϕ uit (4.12) de waarde één heeft.

Schattingsmethode

We kunnen nu de meeste aannemelijke schatter $\hat{\beta} = (\hat{\beta}_0 \ldots, \hat{\beta}_p)^T$ schatten door de zogenoemde *log-likelihood* functie $l(\beta)$ te maximaliseren:

$$l(\beta) = \sum_{i=1}^{n} [Y_i log(pr_i) + (1 - Y_i) log(1 - pr_i)] \tag{4.13}$$

Het maximaliseren van deze log-likelihood functie wordt gedaan met behulp van een numeriek algoritme. In [6] kan men een uitgebreid voorbeeld vinden van de *Fisher scoring* methode van Nelder en Wedderburn uit 1972.

Kwaliteitsmetriek

Wij besluiten het logistische regressie model met een metriek voor de kwaliteit van het model. Deze McFadden *Pseudo* R^2 is vergelijkbaar met de R^2 van het lineaire model en wordt aldus [28] als volgt bepaald:

$$McFadden\ R^2 = 1 - \frac{-2l(\hat{\beta})}{-2l(\beta_0)} \tag{4.14}$$

In (4.14) staat $l(\hat{\beta})$ staat voor de log-likelihood van het model met alle verklarende variabelen en $l(\beta_0)$ staat voor de log-likelihood van het model met zonder alle verklarende variabelen. De McFadden R^2 heeft waarden tussen nul en één en het model met de grootste waarde is het beste model.

Verdere informatie

We zien dus dat de GLM's een wijde reeks aan responsvariabelen aan kunnen. We zijn nog niet ingegaan op de beoordeling van de kwaliteit van de coëfficiënten en ook niet op een methodes om al dan niet bepaalde verklarende variabelen op te nemen. We verwijzen hiervoor graag naar een algemeen boek over GLM's in [9]. In dit boek kunt u ook informatie vinden over een modelleertechniek voor een responsvariabele met meerdere categorieën en wordt ook ingegaan op additieve modellen, waarbij de verklarende variabelen een niet-lineaire relatie hebben.

4.5 Beslissingsbomen

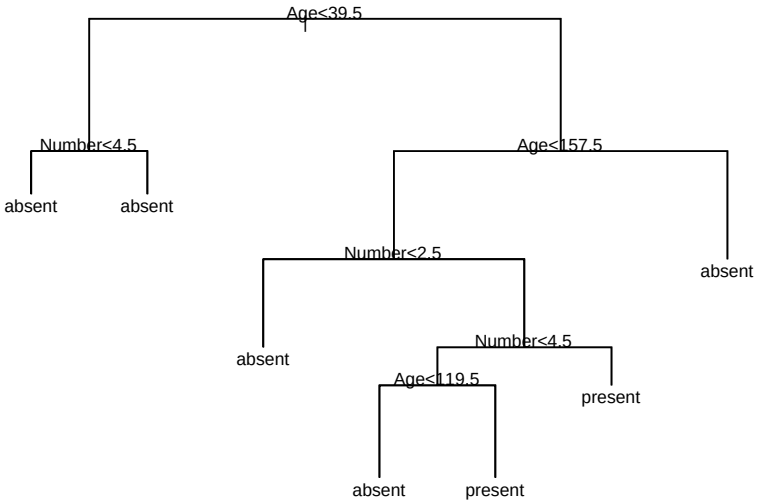
De vorige drie behandelde modelleertechnieken waren allemaal puur statistische technieken. Op het veld van beslissingsbomen komt statistiek samen met het gebied van *datamining*. Datamining wordt in [27, p. 3] gedefinieerd als *een automatisch of semi-automatisch proces dat patronen in de data ontdekt* en wordt ook vaak wel *machine learning* genoemd. Midden jaren 80 publiceerden zowel vier statistici over inductie van beslissingsbomen in hun book [17] als de vooraanstaande "machine learning" onderzoeker Quinlan, die in diezelfde tijd bezig was met het ontwikkelen van een systeem om bomen af te leiden uit voorbeelden (zie [21]).
In [11] wordt het doel van beslissingbomen beschreven als *het verkrijgen van een verzameling regels of een diagram, die makkelijk kunnen worden gelezen om zo een beslissing te kunnen maken bij welke groep een bepaald object hoort*.
We onderscheiden *classificatie* bomen, waarbij de responsvariabele een categorische variabele is en *regressie* bomen, waarbij de responsvariabele continu is. We gaan in deze sectie in op classificatie bomen en deze sectie is gebaseerd op [11]. We zullen in Secties 4.5 en 4.6 de datamining technieken beschrijven aan de hand van achtereenvolgens de *modelspecificatie, modelbouw* en de *modelkwaliteit*.

Modelspecificatie

We kunnen niet zoals bij de statistische modellen de classificatieboom met een wiskundige formule beschrijven. We zullen een algemene uitleg geven en daartoe zullen we allereerst de variabelen hun bijbehorende "datamining" benamingen geven. De categorische responsvariabele (met k categorieën) wordt gerepresenteerd door k klasses en de l verklarende variabelen worden opgevat als attributen. We hebben weer n projecten geobserveerd, die worden aangeduid als objecten. De boom, die wij als voorbeeld geven in Figuur 4.3 op de volgende pagina is een zogenaamde binaire boom, waarbij de startknoop of *wortel* van de boom met n objecten steeds weer in twee subknopen wordt verdeeld op basis van bepaalde waardes van een attribuut. De onderste *eindknopen* of *bladeren* geven de klasse weer die bij de subverzameling van objecten hoort. In Figuur 4.3 wordt de ziekte Kyphosis geclassificeerd (absent/present) op basis van twee attributen (Age,Number). We zien duidelijk dat een boom een verzameling is van een aantal beslissingsregels, die we kunnen aflezen uit de paden vanuit de *wortel* naar de verschillende *bladeren* van de boom. Een beslissingsregel uit de boom van Figuur 4.3 is bijvoorbeeld:

- IF Age $<$ 39.5 AND Number $<$ 4.5 THEN Kyphosis = ABSENT

We willen verder toevoegen dat een boom ideaal is voor een combinatie van categorische en continue attributen. De continue attributen dienen wel in bepaalde categorieën te worden ingedeeld. Voor het discretiseren van attributen verwijzen we u naar [11].



Figuur 4.3: Voorbeeld van een beslissingsboom.

Modelbouw

De vier belangrijke elementen bij het bouwen van een boom zijn aldus [11]

- Hoe kunnen we elke knoop van de boom het best verdelen in subknopen.
- Wanneer stoppen we met verdelen en vormen we de bladknopen van de boom.
- Welke klasse wijzen we toe aan elke bladknoop.
- Hoe kunnen we de boom simpel houden om zo een algemeen beeld te krijgen.

We gaan er dus vanuit dat we k reponsklasses hebben en dan kunnen we voor elke knoop t de zogenaamde informatie berekenen met functie (4.15), waarbij we opmerken dat p_i staat voor de fractie van objecten van klasse i, die aanwezig zijn in de knoop.

$$Info(t) = -\sum_{i=1}^{k} p_i log(p_i) \tag{4.15}$$

Als we nu deze knoop gaan verdelen in l subknopen ($t_1, \ldots, t_l$) aan de hand van een bepaalde attribuut X, dan verkrijgen we de informatie van deze subknopen volgens formule (4.16) en kunnen we de zogenoemde informatiewinst (engels: gain) bereken door simpelweg naar het verschil in informatie tussen de originele knoop en de verzameling subknopen.

$$Info(X,t) = \sum_{i=1}^{l} \frac{\mid t_i \mid}{\mid t \mid} Info(t_i) \tag{4.16}$$

$$Gain(X,t) = Info(t) - Info(X,t) \tag{4.17}$$

We kunnen nu voor elk attribuut X de informatiewinst bepalen en kiezen dan om de knoop te verdelen volgens het attribuut met de grootste informatiewinst. We kunnen deze splits doorvoeren totdat een stopcriterium is bereikt. Een voorbeeld van een heel simpel stopcriterium is door gewoon een minimale grootte op te geven van de te vormen eindknopen. Natuurlijk wordt er eerder gestopt als een knoop een informatie van nul heeft, dan is het de perfecte eind knoop. In deze eindknopen kunnen we nu gewoon kijken naar de grootste waarde van p_i om zo te bepalen welke reponsklasse bij deze knoop hoort. We zullen hier niet ingaan op het generaliseren van een boom, maar verwijzen hiervoor graag naar [21]. In dit boek kunt u ook informatie over extra verdeel-, stop en toewijzingsmethodes.

Modelkwaliteit

Een belangrijke algehele kwaliteitsmaat voor een boom is de classificatiefout. Deze staat simpelweg voor de fractie van verkeerd geclassificeerde objecten. Aan elke beslissingsregel kan ook een kans worden meegegeven. We beschouwen de regels voor de twee meest linkse bladeren in onze boom van Figuur 4.3.

- IF Age $<$ 39.5 AND Number $<$ 4.5 THEN Kyphosis = ABSENT (100%)
- IF Age $<$ 39.5 AND Number $\geq$ 4.5 THEN Kyphosis = ABSENT (85%)

De kwaliteit van de regels wordt nu weergegeven door het percentage van correct geclassificeerde objecten (in dit geval het percentage van niet-zieke patient, die inderdaad geen ziekte hebben). De in deze sectie besproken kwaliteitsmaten hebben weer alleen zin, als de data is verdeeld in een trainings- en een testset.

4.6 Rough data models

Rough data models is echt een pure classificatie techniek en heeft dan ook een categorische responsevariabele. Deze techniek werkt ook alleen met categorische verklarende variabelen, zodat alle continue variabelen weer moeten worden gediscretizeerd. We noemen net als in de vorige sectie de responsvariabele en de verklarende variabelen weer in hun benaming van respectievelijk klasses en attributen. We hebben deze sectie gebaseerd op de uitleg in [11, Hoofdstuk 5].

Modelspecificatie

Een rough data model (RDM) bestaat eigenlijk uit een aantal georderende clusters[3], die samen dus alle mogelijke waarden van de attributen bestrijken. Bij elk cluster wordt ook een bepaalde respsonsklasse gekozen.Elk cluster representeert dus een beslissingsregel en deze beslissingsregels zijn dus geordend qua belangrijkheid.

Modelbouw

Een RDM is wordt bepaald door een aantal keuzes:

- *classificatiefunctie*: Welke classificatiefunctie wordt er gebruikt? De beschikbare classificatie functies zijn hetzelfde als in de vorige sectie over beslissingsbomen. Ze bepalen welke klasse (van de repsons variabele) aan elk cluster wordt toegewezen. De meest simpele vorm is de meerderheidsregel, waarbij de grootste klasse wordt toegewezen aan het cluster.

- *Attributen*: Welke attributen (verklarende variabelen) nemen we mee in het model? Dit kan van te voren gedaan worden met behulp van exploratieve statistische analyses of door verschillende RDM's te bouwen en deze daarna met elkaar te vergelijken.

- *lineaire ordening*: De keuze van de ordening van de clusters hangt af van het beoogde doel. Je kan met je model bijvoorbeeld een kleine hoeveelheid projecten zo goed mogelijk willen voorspellen, maar je kan ook zoveel mogelijk projecten willen indelen aan de hand van zo weinig mogelijk regels.

We zien dus dat de keuze van attributen eigenlijk impliceert dat je een grote hoeveelheid modellen moet bouwen, waarin je de meegenomen verklarende variabelen varieert en dan hieruit het beste model selecteert.

[3] een cluster is eigenlijk een mogelijke combinatie van alle attribuutwaarden

Modelkwaliteit

De modelkwaliteit van het RDM is gebaseerd op de kwaliteitskenmerken van de clusters C_i:

- Class(C_i) is de responsklasse, die aan cluster C_i is toegewezen op basis van de gekozen classificatie functie.
- Size(C_i) is het aantal observaties in cluster C_i
- Corr(C_i) is het aantal observaties in C_i met de bijhorende responsklasse Class(C_i).
- Accuracy(C_i) is het aantal goed geclassificeerde observaties van een cluster C_i en wordt weergegeven door $\frac{Corr(C_i)}{Size(C_i)}$

Een voorbeeld van een algehele kwaliteitsmetriek is de cummulatieve accuracy van het model.

$$cma(i) = \frac{Corr(C_1) + \ldots + Corr(C_i)}{Size(C_1) + \ldots + Size(C_i)} \tag{4.18}$$

Ons model is een verzameling van geordende clusters en de cma(i) representeert dan ook de kwaliteit van de i meest belangrijke clusters.

Hoofdstuk 5

IT voorspelmodellen

In dit hoofstuk zullen we ingaan op het gebruik van wiskundige voorspelmodellen op het gebied van IT projecten. We hebben gezien dat vanuit historisch perspectief de meeste voorspelmodellen zijn ontwikkeld op het gebied van software engineering en dan met name op het gebied van de software ontwikkeling. De eerste modellen waren software kostenschattings modellen. In zowel [3] als [23] wordt het SDC kostenmodel van Nelson uit 1965 genoemd als een van de eerste algoritmische modellen. Dit lineaire model schat de voor een software project benodigde inspanning (in man-maanden[1]) aan de hand van 13 verklarende variabelen (zoals het aantal subprogramma's of het feit of een *high-level* programmeertaal al dan niet was gebruikt). Eind jaren 70 werden vele kostenmodellen ontwikkeld, die toen een flinke vooruitgang betekenden van het vakgebied. Voor een historisch overzicht verwijzen we u graag naar [3]. Vele van de toen ontwikkelde modellen waren gebaseerd op het de correlatie tussen programmeer inspanning en grootte van het programma. Dit verband werd in de modellen veelal weergegeven door een niet-lineaire relatie tussen. Een voorbeeld van een niet-lineair kostenmodel is de zogenoemde *software equation* van Putnam uit [20].

$$\text{Omvang}(\%\text{Fouten}) = (\text{Inspanning}/\beta)^{(1/3)}\text{x Tijd}^{(4/3)}\text{x Proces Productiviteit} \quad (5.1)$$

Hierbij is de β een omvangs afhankelijke parameter, die als bedoeling heeft om een groter gewicht te geven aan de inspanningsfactor bij hele kleine systemen. De vergelijking (5.1) is een door de jaren heen iets aangepaste versie van de oorspronkelijke vergelijking uit 1978. De vergelijking, die in [3] ook wel het SLIM model wordt genoemd SLIM model was de basis van commerciële[2] produkten, die helpen om software projecten te plannen en te controleren gedurende de hele software levenscyclus. Deze kostenschattings modellen zijn heel handig om voor het

[1] Het aantal man-maanden staat voor het werk van een persoon in een maand en daaraan is meestal vaste kostprijs worden verbonden; vandaar dat inspanning en kosten vaak als een en dezelfde worden beschouwd in deze kostenschattingmodellen.

[2] Putnam's bedrijf heet QSM (Quantitative Software Management). Bekijk hun website (www.qsm.com) voor meer info.

begin van het project een schatting te maken van de verwachte kosten en deze tijdens het process te beheersen. Ze geven je de mogelijkheid om bij het eind van een bepaalde fase in het project weer even in te schatten hoeveel het geplande budget voor de resterende fase is veranderd door de huidige gang van zaken.
De enige vorm van risicovoorspelling in deze techniek zit hem in het variëren van bijvoorbeeld de verwachte omvang[3], waardoor ook een boven- en ondergrens in de kostenschatting wordt gegeven. Deze technieken zijn dus handig om individuele projecten te beheersen, maar we willen toch ook graag in staat zijn om te kunnen bepalen welk projecten het meest risicovol zijn binnen een groep projecten.

In [24] worden *niet-lineaire* modellen geïntroduceerd, die de kansen berekenen dat bepaalde software projecten gefaald of te laat zijn. Hierbij wordt onder een gefaald project verstaan, dat het helemaal niet is opgeleverd. Wij geven het model weer voor de kans dat een outsourced[4] project faalt:

$$cf_{out}(f) = 0.33 \cdot (1 - exp(-0.003 \cdot f^{0.678})) \tag{5.2}$$

De faalkans (cf) wordt dus berekend aan de hand van het aantal functiepunten (f), die de hoeveelheid functionaliteit representeren en een objectieve maat zijn voor de omvang van de software. De niet-lineaire relatie is afgeleid aan de hand van zes algemene benchmarks uit Jones database [24, p.47], die het percentage van gefaalde projecten weergeven per aantal functiepunten. Het grootste aantal functiepunten is 100000 en het model is dan ook niet geschikt voor grotere projecten, omdat de vergelijking geen hogere kans kan aannemen dan 0.5. In dit opzicht is dit model dus wel geschikt om projecten onderling met elkaar te vergelijken als je puur uitgaat van een categorie. Dit model is eigenlijk alleen aan te raden als je inderdaad weinig waarnemingen hebt.

Op het gebied van gegeneraliseerde modellen en dan met name logistische regressie hebben we geen publieke onderzoeken gevonden, die ingaan op het voorspellen van projectrisico's. Een toepassing van een datamining techniek zien we bij Labbi [18] in 2001. Zijn IBM onderzoeksrapport gaat in op het gebruik van beslissingsbomen[5] om projecten met een bepaalde kans in te delen in bepaalde risicocategorieën. In dit specifieke geval gaat het om software projecten, die IBM voor zijn klanten doet en het te voorspellen risico is dan ook het verlies (verlaging in de winstmarge) van een project. Bij elke risicocategorie hoort een bepaalde kansverdeling van het verlies. Het totale verwachte verlies van een project kan nu worden berekend door de kansen op iedere categorie te vermenigvuldigen met het verwachte verlies in de bijbehorende projectcategorie. Deze aanpak is heel mooi vanuit het oogpunt van de ontwikkelaar, die de software oplevert op een bepaald

[3] QSM heeft een enorme historisch database opgebouwd met afwijkingen in deze schattingen, waaruit een bepaalde boven en ondergrens kan worden gegeven.

[4] een software project, dat door een commerciële IT ontwikkelaar voor een klant wordt ontwikkeld.

[5] In dit onderzoek zijn de beslissingsbomen gebouwd met het AdaBoost algoritme, zie [18].

punt en dan er in principe van af is. Op dat punt start immers het onderhoud, waarvoor vaak een onderhoudscontract wordt afgesloten.

We zien dus dat de in dit hoofdstuk besproken IT voorspelmodellen meestal alleen gebruikt worden om een afwijking dan wel een kans te bepalen. Het IBM risicomodel van Labbi is het enige model dat apart een kans voorspelt en het daarbijbehorende verlies inschat (al is dit op basis van verdelingen en niet op basis van een van de besproken technieken). We hebben verder ook opgemerkt dat geschikte technieken als *logistische regressie* of *rough data models* helemaal niet gebruikt worden bij het bepalen van risicovolle IT projecten.

We geven in Tabel 5.1 een overzicht van de geschiktheid van voorspelmodellen om een *kans* of een *afwijking* te voorspellen. We hebben hierbij onderscheid gemaakt tussen *uitgebreide* en *beperkte* projectinformatie, waaronder we respectievelijk de aanwezigheid van voornamelijk continue projectkenmerken dan wel categorische projectkenmerken verstaan. De geschiktheid van de technieken is voornamelijk afgeleid vanuit de theoretische kenmerken van de modellen. Verder hebben we daar waar mogelijk gebruik gemaakt van de praktische voorbeelden uit dit hoofdstuk. Het niet-lineaire model (5.2) kan gebruikt worden om een kans te voorspellen, maar is alleen in bepaalde situaties bruikbaar. Het is namelijk alleen bruikbaar, wanneer er maar één ongewenste uitkomst is. We hebben dan ook aangegeven, dat dit model soms geschikt (S) is.

Project informatie	Uitgebreid		Beperkt	
Voorspelmodel	Afwijking	Kans	Afwijking	Kans
Lineaire modellen	**G**	**O**	**G**	**O**
Niet-lineaire modellen	**G**	**S**	**G**	**O**
GLM (logit) model	**O**	**M**	**O**	**G**
Overige GLM modellen	**M**	**O**	**M**	**O**
Classificatiebomen	**O**	**G**	**O**	**M**
Regressiebomen	**G**	**O**	**G**	**O**
Rough data model	**O**	**G**	**O**	**G**

M	model is meest geschikt	G	model is geschikt
S	model is soms geschikt	O	model is ongeschikt

Tabel 5.1: Overzicht geschikte voorspelmodellen

Hoofdstuk 6

Conclusies

De doelstelling van dit werkstuk was oorspronkelijk om aan te geven welke wiskundige voorspelmodellen geschikt zijn om het risico te kwantificeren van een IT project. We hebben echter al snel moeten constateren, dat het risico van IT projecten niet met één model te kwantificeren is.

We raden daarom de volgende stappen aan om tot kwantificeerbaar risico te komen; Allereerst zullen de ongewenste uitkomsten van een project moeten worden onderkend en vervolgens worden onderverdeeld in eenduidige en onderling onafhankelijke risicocategorieën. We hebben vastgesteld dat de definities van ongewenste uitkomsten uit Hoofstuk 3 niet altijd even eenduidig zijn; hierin ligt dus de eerste en ook belangrijkste uitdaging[1]. In de tweede stap moeten we voor ieder project de *kans* bepalen, dat het tot een bepaalde risicocategorie behoort. We hebben in Hoofdstuk 4 gezien, dat van de geschikte technieken alleen *beslissingsbomen* worden gebruikt. Het toepassen van *logistische regressie* en *rough data models* zal zeker tot nieuwe inzichten kunnen leiden. Tenslotte zullen we (per project) ook een bepaalde gekwantificeerde *afwijking* bij elke risicocategorie moeten bepalen. Dit kunnen we doen aan de hand van een van onze voorspelmodellen, maar ook door naar verdelingen van historische afwijkingen te kijken.

We zien dus dat we met onze behandelde technieken in theorie goed in staat zijn om de twee belangrijke onderdelen van een risicomodel te kunnen voorspellen. We hebben in de praktijk gezien, dat de huidige IT voorspelmodellen vooral op de risico's ingaan die de kosten van software projecten beïnvloeden. De grootste uitdaging bij risicomodellering voor IT projecten wordt dan ook om van kop tot staart het risico op een lagere NCW te kunnen kwantificeren en beheren.

[1]Met eenduidige definities kunnen we met één model de kansen bepalen op de ongewenste uitkomsten. In het andere geval moeten we meerdere kansmodellen maken, waarbij de samenhang tussen deze kansen van belang is

Bibliografie

[1] H. Arsham. Decision making in economics and finance (9th edition). University of Baltimore,2003. Beschikbaar via: http://www.webversity.com/Courses/Statistics_1/opre330Forecast.htm#rsintroduction.

[2] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.

[3] B. W. Boehm. Software engineering economics. *IEEE Transactions on Software Engineering*, SE-10(1):4–21, 1984.

[4] B. W. Boehm. Software risk management: Principles and practices. *IEEE Software*, 8(1):32–41, 1991.

[5] M. Curley. *Managing Information Technology for Business Value*. Intel Press, 2004.

[6] M. de Gunst. *Statistische modellen*. Vrije Universiteit, 1999.

[7] M. de Gunst and A. van der Vaart. *Statistische Data Analyse*. Vrije Universiteit, 1998.

[8] L. de Vries. Software metrieken: Een statistische analyse. BWI werkstuk, Vrije Universiteit Amsterdam, Februari 1999.

[9] A. J. Dobson. *An Introduction to Generalized Linear Models - Second edition*. Chapman and Hall, 2002.

[10] P. Gardiner and K. Stewart. Revisiting the golden triangle of cost, time and quality: the role of npv in project control, success and failure. *International Journal of Project Management*, 18(4):251–256, 2000.

[11] R. Groenevelt and S. van Westerop. Datamining techniques: Rough sets, decision trees, and rough data models. BWI werkstuk, Vrije Universiteit Amsterdam, November 1998.

[12] Insightful Corporation. *S-PLUS 6 for Windows Guide to Statistics, Volume 1*, 2001.

[13] R. P. J.F.A. Spangenberg and E. van Heijningen. Investeren in informatietechnologie - rendement op onzekerheid? *de Accountant*, nr. 4:242–246, 1999.

[14] B. Kersten and C. Verhoef. It portfolio management: A banker's perspective on it. *Cutter IT Journal*, 16(4):27–33, 2003. Beschikbaar via: http://www.cs.vu.nl/∼x/bp/bp.pdf.

[15] A. Khodabandeh and P. Palazzi. Software development: People, process, technology). CERN,1995. Beschikbaar via: http://ipt.web.cern.ch/IPT/Papers/Sopron94/proceedings.ps.

[16] G. Koole. Optimization of business processes: Applications and theory of mathematical modeling. Vrije Universiteit Amsterdam, May 2004. Beschikbaar via http://www.cs.vu.nl/∼koole/obp/obp.pdf.

[17] R. O. L. Breiman, J.H. Friedman and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks/Cole, 1984.

[18] A. Labbi and M. Cuendet. Boosted decision trees for project risk assessment and pricing. IBM Research Report, April 2002.

[19] L. Putnam and W. Myers. *Measures for Excellence: reliable software on time, within budget*. Prentice Hall, 1992.

[20] L. Putnam and W. Myers. *Five Core Metrics: the intelligence behind successful software management*. Dorset House, 2003.

[21] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[22] The Standish Group. *Extreme CHAOS*, 2001. Beschikbaar via: http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf.

[23] H. van Vliet. *Software Engineering: Principles and Practice*. John Wiley & Sons, 1993.

[24] C. Verhoef. Quantitative it portfolio management. *Science of Computer Programming*, 45(1):1–96, 2002. Available via: http://www.cs.vu.nl/∼x/ipm/ipm.pdf.

[25] P. Weill and M. Broadbent. *Leveraging the new Infrastructure - How Market Leaders Capitalize on IT*. Harvard Business School Press, 1998.

[26] Wikipedia. *Linear regression*, 2004. Beschikbaar via: http://en.wikipedia.org/wiki/Linear_regression.

[27] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

[28] J. Woolridge. *Econometric Analysis of Cross Section and Panel Data*. The MIT Press, 2002.

[29] Word iQ. *Online Encyclopdia*, 2004. Beschikbaar via: http://www.wordiq.com/definition/Information_technology.

[30] K. Yeo. Critical failure factors in information system projects. *International Journal of Project Management*, 20(3):241–246, 2003.