2010
2011

BEDRIJFSECONOMISCHE WETENSCHAPPEN

*master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica: informatie- en communicatietechnologie*

# Masterproef

## *Nieuwe kennis halen uit productbeoordeling websites aan de hand van datamining*

Promotor :
dr. Benoit DEPAIRE


Merijn Campsteyn


*Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica , afstudeerrichting informatie- en communicatietechnologie*

universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST


Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt



universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST

2010
2011

# BEDRIJFSECONOMISCHE WETENSCHAPPEN

*master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica: informatie- en communicatietechnologie*

# Masterproef

*Nieuwe kennis halen uit productbeoordeling websites aan de hand van datamining*

Promotor :
dr. Benoit DEPAIRE


## Merijn Campsteyn


*Masterproef voorgedragen tot het bekomen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica , afstudeerrichting informatie- en communicatietechnologie*

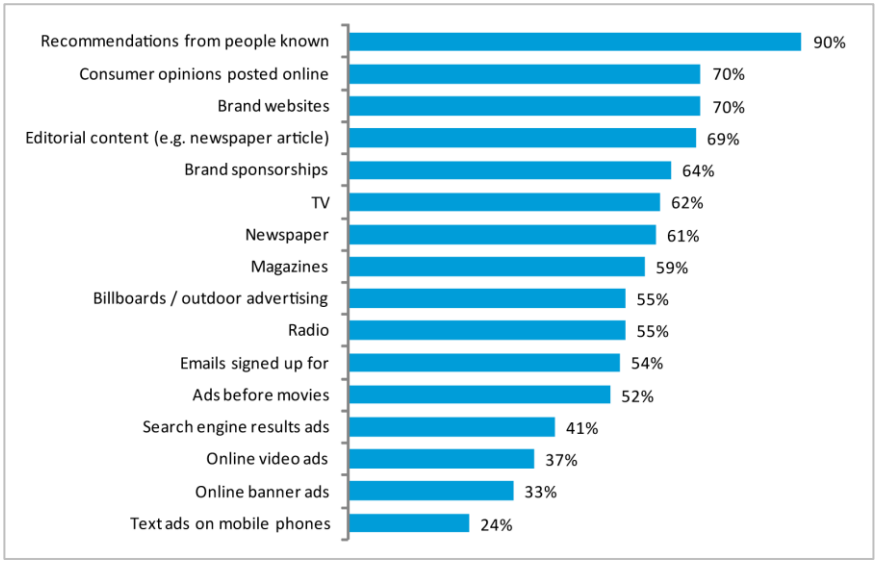
universiteit hasselt
UNIVERSITEIT VAN DE TOEKOMST

# 1 Inhoud

# 2 Inleiding

Het gebruik van internettoepassingen is een vast onderdeel geworden van onze levenswijze. Steeds vaker komt hier ook online winkelen bij. Volgens een studie van betalingsprovider Ogone (Ogone, 2011) verwerkte de Belgische e-commerce sector in 2010 10,7 miljoen transacties en was deze goed voor een omzet van 903 miljoen euro. Een stijging van 28% ten opzichte van 2009.

Veel webwinkels in zowel binnen- als buitenland bieden aan hun klanten de mogelijkheid om beoordelingen te schrijven over hun producten. Heel wat klanten maken gebruik van deze productbeoordelingen tijdens hun aankoopbeslissing. Klanten hechten immers een grotere geloofwaardigheid aan de mening van andere consumenten dan aan de beschrijving die een producent op zijn website voorziet (Nielsen Company, 2009). Het effect van online productbeoordelingen laat zich niet alleen voelen in het aankoop gedrag van consumenten bij online bestellingen maar steeds vaker gebruiken consumenten de informatie in beoordelingen voor hun aankopen in fysieke winkels (Deloitte & Touche, 2007).




**Mate van vertrouwen in advertentievormen volgens de Nielsen Global Online Consumer Survey April 2009**

*Percentage geeft het aandeel weer van alle respondenten die de advertentievorm als betrouwbaar hebben geëvalueerd.*

De grote hoeveelheid aan openlijk te raadplegen productbeoordelingen biedt een unieke kans aan organisaties en producenten om snel en in een groot volume feedback van klanten te krijgen. Waar de evaluatie van producten van vroeger vaak via de bevraging van een panel of via enquêtes diende te gebeuren, kunnen werknemers van een bedrijf nu op zoek gaan naar beoordelingen op heel wat webwinkels, fora en blogs. Heel vaak gebeurt de verzameling van dergelijke informatie op een geautomatiseerde wijze aan de hand van *'web scraping'* software. Dergelijke software is in staat om webpagina's in te lezen, de informatie op te slaan waarin we geïnteresseerd zijn en vervolgens de logische linkstructuur in elektronische documenten kan volgen om hierna weer een nieuwe

webpagina op te halen. Er zijn ook gespecialiseerde firma's[1] die een bedrijfsmodel hebben opgebouwd rond het verzamelen en gestructureerd aanbieden van productbeoordelingen.

De keerzijde van de medaille is dat de hoeveelheid aan beschikbare informatie voor een operationele uitdaging zorgt. Hoe kan deze informatie op de meest efficiënte manier verzameld worden? Hoe kan de verzamelde informatie vervolgens het best verwerkt worden zodat hier compacte en relevante informatie uit gehaald kan worden? Aan de consumentenzijde kunnen we vragen stellen als hoe kunnen de meeste relevante productenbeoordelingen geselecteerd worden voor een bepaald product? Hoe kunnen productbeoordelingen helpen om consumenten te begeleiden in het sneller vinden van een gewenst product of dienst?

Het traditionele antwoord op deze vragen is het gebruik van datamining. Datamining is in staat om impliciet verborgen, maar mogelijk waardevolle verbanden te ontdekken (Cattral, Oppacher, & Deugo, 2001). Binnen het domein van datamining zijn er een hele verzameling van modellen en technieken die in staat zijn om patronen te ontdekken. Elk van deze technieken verschilt in het type data dat geanalyseerd kan worden en het type van onderzoeksvragen dat mogelijk beantwoord kan worden. Een gemeenschappelijke voorwaarde voor alle datamingtechnieken is dat deze de te analyseren data op een min of meer gestructureerd manier nodig hebben in de vorm van variabelen. Het grootste aandeel aan opgeslagen informatie bestaat echter uit tekst en is ongestructureerd. Dit is zeker ook het geval bij productbeoordelingen. Sommige websites bieden de mogelijkheid om een algemene beoordeling te geven aan het product in de vorm van een cijfer (vaak weergegeven door sterretjes), en soms een mogelijkheid tot quotering van enkele producteigenschappen. De informatiewaarde van deze numerieke data is vaak echter beperkt. De echte waarde zit in de tekst. En dit brengt ons bij het begrip Text Mining.

Text Mining is een relatief recent onderzoeksdomein dat tracht om de informatiewaarde die in tekst zit tracht te ontdekken en te structureren zodat deze ook aangewend kan worden. Een groot deel van de tijd en energie die besteed wordt bij het werken met Text Mining toepassingen gaat naar het voorbereiden van de data. De zoektocht naar mogelijke verbanden kan gebeuren met dezelfde technieken als die van datamining, al dienen sommige modellen licht aangepast te worden voor toepassing binnen Text Mining. Enkele van de meest voorkomende modellen zullen in deze masterproef uitgebreid besproken worden.

### *Economische waarde van productbeoordelingen*

De economische waarde die mogelijk uit productbeoordelingen gehaald kan worden kan het best geïllustreerd worden aan de hand van een eenvoudig voorbeeld. Zo was de omzet van Amazon.com een van de grootste online retailers ter wereld in 2010 maar liefst 34,2 miljard USD[2] (Amazon.com, 2011). Wanneer het gebruik van Text Mining ons beslissingsondersteunende informatie zou kunnen verschaffen waardoor de verkopen met slechts 0,01% zouden toenemen. Verhoogd dit de omzet met maar liefst 3,42 miljoen USD. Heel wat bedrijven zijn bereid om heel wat uren onderzoek te financieren voor een dergelijk resultaat.

[1] Internationale voorbeelden zijn onder andere reviews.cnet.com en epinions.com. Dichter bij huis is Ala Test een grote speler in de markt (http://nl.alatest.be/)
[2] ongeveer 23,78 miljard EUR op 22/08/2011

## *Onderzoeksopzet*

Deze masterproef start met een algemene inleiding rond Data- en Text Mining. Er wordt kort geschetst wat de raakvlakken en verschillen zijn tussen datamining en welke de onderliggende academische domeinen en technieken zijn.

In het eerstvolgende hoofdstuk wordt in gegaan op enkele vaak gebruikte methoden voor het classificeren van tekst, een veel gebruikte toepassing binnen Text Mining. De classifiers die aan bod komen zijn: Naïve bayes, beslissingsbomen, neurale netwerken en Support Vector Machines. Tot slot wordt er in dit hoofdstuk gekeken naar enkele kerncijfers die berekend kunnen worden om een classificatieresultaat te evalueren.

Vervolgens wordt er onderzocht welke de methoden zijn voor *'feature extraction'*. Dit is op zoek gaan naar de meest beschrijvende woorden voor een bepaalde groep documenten. In dit hoofdstuk zal voornamelijk de klemtoon liggen op het zoeken naar feature extraction technieken die in staat zijn om producteigenschappen te herkennen in productbeoordelingen.

In het hoofdstuk "experimenten" worden er enkele experimenten uitgevoerd om na te gaan of de modellen en technieken die in de voorgaande hoofdstukken besproken werden ook toepasbaar zijn voor gebruik bij productbeoordelingen. In de eerste plaatst wordt er nagegaan wat het classificatieresultaat is van de besproken classificatiemethoden. Vervolgens wordt de impact van technieken voor 'data preprocessing' getoetst op een Naïve Bayes classifier. In dit experiment wordt nagegaan of het classificatieresultaat verbetert wanneer er achtereenvolgens een filtering gebeurt van zogenoemde stopwoorden, en wanneer er een stemmingsalgoritme wordt gebruikt. Het testen van de Naïve Bayes classifier gebeurt aan de hand van een zelf geschreven classifier op basis van PHP en MySQL. Een bijkomend doel van de experimenten is onderzoeken hoe performant een Naïve Bayes classifier is binnen deze ontwikkelomgeving en hoe bruikbaar dit type van classifier is voor gebruik in online toepassingen.

## 2.1 Onderzoeksvragen

Een van de belangrijkste functies binnen het domein van Text Mining is het classificeren van een document. We zouden bijvoorbeeld willen weten of een review positief of negatief is. Hoe wordt een product beoordeeld in een artikel op een blog? Hoe evalueren klanten een product op online fora? Om deze classificatiefunctie uit te voeren zijn er tal van technieken en modellen ter beschikking van een onderzoeker. Dit leidt tot de eerste onderzoeksvraag:

**1. Welke zijn de meest gebruikte classificatietechnieken binnen Text Mining?**

Er wordt via een literatuurstudie nagegaan welke de meest voorkomende classificatietechnieken zijn binnen Data- en Text Mining. Vervolgens worden de voor- en nadelen van elke techniek onderzocht aan de hand van bestaande literatuur en door het uitvoeren van enkele experimenten.

Tekstuele documenten binnen een organisatie vormen een belangrijke bron van informatie. Een tweede functie van Text Mining is het extraheren van deze informatie. Toegepast op het thema van deze masterproef leidt dit tot volgende onderzoeksvraag:

**2. Hoe kunnen producteigenschappen herkend worden in een verzameling van productbeoordelingen?**

Er wordt onderzocht welke Tekst Mining-technieken er momenteel beschikbaar zijn voor het herkennen van features in documenten en of deze technieken gebruikt kunnen worden voor het herkennen van producteigenschappen in productbeoordelingen. Het onderzoek gebeurt ook hier aan de hand van een literatuurstudie en de uitvoering van enkele experimenten.

Een veel gebruikte classificatiemethode is Naïve Bayes. Deze *'classifier'* wordt gekenmerkt door zijn eenvoudige opzet en zijn snelheid in het verwerken van trainingdata en het classificeren van documenten. Dit type van classifier is daarom een ideale kandidaat om te gebruiken in een online omgeving waarbij de een gebruiker een snel resultaat verwacht. Dit brengt ons bij volgende onderzoeksvraag:

**3. Wat is de performantie van een Naïve Bayes classifier in een PHP/MySQL omgeving?**

Er wordt nagegaan hoe een Naïve Bayes classifier het best geprogrammeerd en geïmplementeerd kan worden op basis van de programmeertaal PHP en een onderliggende MySQL database[3].

[3] Deze begrippen worden uitgelegd in het volgende hoofdstuk.

## 2.2 Methodologie

### 2.2.1 Onderzoeksopzet

Het onderzoek werd gestart met een korte, verkennende literatuurstudie naar het probleem van informatieverzameling vanuit tekst en de eventuele toepassing hiervan op online productbeoordelingen. Hoewel het domein van Text Mining nog vrij recent is, is hier toch al een uitgebreide collectie literatuur over terug te vinden die verder bouwt op een oudere en ruimere basis van statistische technieken en modelleringsmethoden vanuit Data Mining. Binnen de bestaande literatuur is het aandeel dat handelt over de toepassing van Text Mining op productbeoordelingen nog redelijk beperkt.

### 2.2.2 Literatuurstudie

Tijdens de literatuurstudie werd er voornamelijk gebruik gemaakt van de e-bronnen van de UHasselt bibliotheek voor het opzoeken van wetenschappelijke artikelen en papers. Bijkomend werd er via een klassieke opzoeking via het internet gezocht naar verduidelijkingen of definities omdat de wetenschappelijke artikelen vaak een bepaalde methode of techniek in detail beschreven zonder in te gaan op de onderliggende basistheorieën. Er werd getracht om waar mogelijk een aanknopingspunt te vinden met academische literatuur.

### 2.2.3 Praktijkonderzoek

Na het uitvoeren van de literatuurstudie werd er nog een praktijkonderzoek uitgevoerd om de toepasbaarheid van enkele Text Mining technieken op productbeoordelingen te toetsen. In de eerste plaatst worden enkele verschillende classificatiemethoden geëvalueerd waaronder Naïve Bayes, Beslissingsbomen, Neurale Netwerken en Support Vector Machines.

Vervolgens wordt de Naïve Bayes classifier uitvoerig getest in een PHP/MySQL omgeving. Deze combinatie van programmeertaal en onderliggende database is een van de meest voorkomende bij dynamische webapplicaties. Via en praktijkonderzoek wordt nagegaan hoe performant een Naïve Bayes classifier kan zijn en of deze bruikbaar is om te implementeren in webapplicaties.

In een derde fase van het praktijkonderzoek wordt getracht om de meest relevante producteigenschappen te distilleren uit een corpus van productbeoordelingen.

#### 2.2.3.1 Dataverzameling

Voor de experimenten wordt er gebruik gemaakt van productbeoordelingen afkomstig van de websites BestBuy.com en reviews.cnet.com. De productbeoordelingen van BestBuy worden ter beschikking gesteld als een archief dat vrij te downloaden is op een aparte sectie van de website die speciaal voor ontwikkelaars is voorzien. Hierdoor was het mogelijk om snel een groot aantal productbeoordelingen te importeren. In totaal zijn er 499.022 productbeoordelingen in de dataset voor 46.234 producten. Het aantal beoordelingen per product varieert van geen tot meer dan 600. Tijdens het uitvoeren van het praktijkonderzoek zal van het totaal aantal reviews steeds een subselectie gemaakt worden voor de verschillende experimenten.

De reviews van CNet zijn niet opvraagbaar via een ontwikkelaarsprogramma en werden met behulp van *'webscraping'* gedownload. Er werden 16398 productbeoordelingen verzameld voor het domein mobiele telefoons. Alle reviews werden geïmporteerd in een mysql database.

#### 2.2.3.2 Gebruikte software en applicaties

Bij het ontwikkelen van applicaties is ervoor gekozen om steeds gebruik te maken van openbron programmeertalen en applicaties. Het grote voordeel van deze aanpak is dat er vaak uitgebreide documentatie beschikbaar is waardoor het gemakkelijk is om zelf applicaties te bouwen of te wijzigen zodat deze bruikbaar zijn voor deze masterproef.

##### *Apache*

Apache is een opensource webserver. Een webserver zorgt ervoor dat http-verzoeken van een *'client'* verwerkt worden en de gevraagde webpagina of informatie wordt teruggestuurd. Apache kan gebruikt worden op zowel Windows , Mac als Linux/unix machines.

##### *PHP*

PHP is een server-side scriptingtaal waarmee het mogelijk is om objectgeoriënteerd te programmeren. Er werd voor php gekozen omdat dit een open programmeertaal is die vrij en gratis gebruikt kan worden. Bovendien bestaat er een levendige PHP gemeenschap waardoor er tal van documentatie beschikbaar is die gebruikt kan worden tijdens het programmeren van applicaties. De php scriptingtaal is ook een van de meest populaire programmeertalen voor gebruik in webapplicaties waardoor de voorgestelde applicaties in principe meteen bruikbaar zijn voor implementatie in bestaande websites.

##### *MySQL*

Voor de opslag van alle data wordt gebruik gemaakt van MySQL. MySQL is een open source relationeel databasemanagementsysteem dat gebruik maakt van SQL (Wikipedia). Deze keuze voor de combinatie Apache, PHP en MySQL is zeker niet toevallig. Deze combinatie is zo goed als een standaard bij het ontwikkelen van webapplicaties. De drie onderdelen zijn eenvoudig met elkaar te combineren.

##### *USB Webserver*

(http://www.usbwebserver.net/)

USB Webserver is een klein programma waarmee eenvoudig een lokale Apache webserver met MySQL en PHP kan worden opgezet op een standaard computer. Apache, PHP en MySQL kunnen standalone gedraaid worden op een Windows machine zodat het niet meer noodzakelijk is om lange en soms ingewikkelde installatieprocedures te doorlopen.

##### *Dreamweaver*

De code werd rechtstreeks ingevoerd in Dreamweaver. Dreamweaver is een product van Adobe. Er werd geen gebruik gemaakt van de mogelijkheden van dreamweaver om kant en klare codeblokken te schrijven. Het grote voordeel van Dreamweaver is dat code automatisch kleurcodes krijgt

toegewezen waardoor deze gemakkelijker leesbaar is. Verder is Dreamweaver in staat om verschillende PHP bestanden met elkaar te 'linken' zodat de functies van een bepaald PHP bestand als suggesties verschijnen in een ander bestand. Dit vergemakkelijkt en versnelt de invoer van code.

### *RapidMiner*

Rapidminer is een openbron softwarepakket voor datamining. Het bevat een uitgebreide verzameling aan dataminingfuncties en kan optioneel worden uitgebreid met bijkomende modules. Voor de experimenten die verder in deze masterproef worden uitgevoerd werden de *Text Processing*, *Webmining* en *Weka*[4] extensie geïnstalleerd.

De Text processing extentie voegt een aantal specifieke functies toe om ruwe tekst om te zetten naar bruikbare data zoals bijvoorbeeld het genereren van tokens, filteren van stopwoorden en het genereren van n-grams. De webmining extentie kan gebruikt worden voor het indexeren van webpagina's en het verwerken van HTML. De Weka extentie voegt alle functies en modellen van Weka toe aan RapidMiner. De extentie bevat een 100-tal extra modellen zoals bijkomende beslissingsbomen, 'rule-learners' en regressiemodellen.

Een groot voordeel aan RapidMiner is dat het in Java geprogrammeerd is en dat het vrij eenvoudig geïmplementeerd kan worden in andere Java applicaties. Dit maakt RapidMiner geschikt voor de ontwikkeling van webapllicaties. Een dergelijke implementatie zal voor deze masterproef niet uitgevoerd worden maar het is een suggestie om dit te overwegen voor online data- en textmining toepassingen die complexe modelleringsmodellen vereisen.

[4] http://www.cs.waikato.ac.nz/ml/weka/

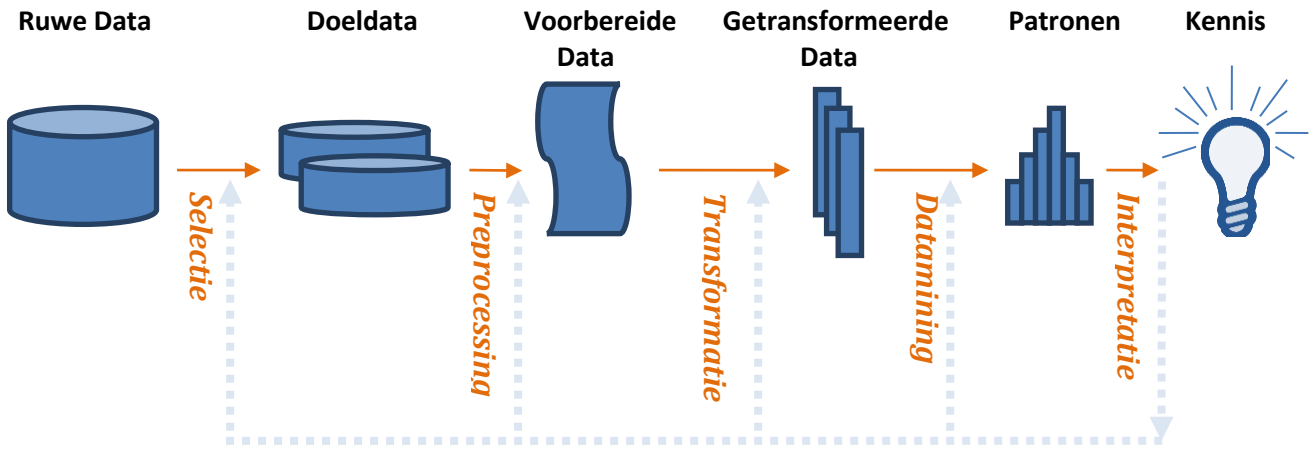
# 3 Text Mining

## 3.1 Inleiding

We leven in een informatietijdperk waarbij de hoeveelheid aan opgeslagen informatie exponentieel toeneemt. Een onderzoek van IDC becijferde dat in 2006 de hoeveelheid aan opgeslagen digitale informatie 161 exabytes[5] bedroeg, ofwel 161 biljoen gigabytes. Dit is meer dan 3 miljoen keer de hoeveelheid informatie in alle boeken ooit geschreven. Op vier jaar tijd verviervoudigde de hoeveelheid digitale data tot maar liefst 988 exabytes. Aan deze groei van informatie lijkt geen einde te komen door de stijgende rekenkracht van computers en het blijvend goedkoper worden van digitale opslag. In deze enorme verzameling aan data zit een grote hoeveelheid data verborgen. Steeds meer bedrijven zijn zich hier van bewust en centraliseren hun bedrijfsdata in gestructureerde databases en *database management systemen (DBMS)*.

De uitdaging bestaat er nu in om deze enorme berg aan data om te zetten in bruikbare en beslissingsondersteunende informatie. Dit proces wordt vaak omschreven als *Knowledge Discovery in Databases (KDD)*. Dit proces bestaat uit verschillende fases. Het aantal fases dat vermeld wordt in de literatuur verschilt maar de gebruikte methodologie is gelijklopend (Fayyad, Gregory, & Padhraic, 1996) (Chapman, et al., 2000):

1. Selectie van data
2. Voorbereiden van data (*preprocessing*)
3. Transformeren van data
4. Datamining
5. Interpretatie en evaluatie




[5] 1 Gigabyte = 1000 Megabyte ; 1 Terrabyte = 1000 Gigabyte ;
1 Petabyte = 1000 Terrabyte ; 1 Exabyte = 1000 Petabyte

### 1. Selectie van de data

De eerste fase start met het beschrijven van de organisatieomgeving. Welke objectieven heeft de organisatie? Welke databronnen zijn er beschikbaar. En hoe kan dit vertaald worden naar een datamining probleem? Van zodra de probleemstelling bepaald is start de selectie van de eigenlijke data. Het is belangrijk voor de onderzoeker om een inzicht te verwerven in de samenstelling van de data. Welke typen data zijn er beschikbaar? Wat zijn de eerste zichtbare verbanden? Wat zijn mogelijke kwaliteitsproblemen van de data? Een eerste inzicht in de data kan helpen bij het vormen van hypothesen die verder in het KDD proces onderzocht worden (Chapman, et al., 2000).

### 2. Voorbereiden van de data (preprosessing)

De preprosessingfase omvat alle stappen die nodig zijn om de initiële data om te zetten naar een finale dataset die bruikbaar is voor dataminingtoepassingen. Deze stap is allesbehalve onbelangrijk omdat de huidige databases en datawarehouses door hun omvang zeer vatbaar zijn voor ruis, ontbrekende data en inconsistente data. De uitdaging in de data preprosessingfase bestaat er in om de kwaliteit van de initiële dataset te verbeteren en zo ook de kwaliteit van de ontdekte patronen in de daaropvolgende dataminingfase (Han & Kamber, 2006).

### 3. Transformeren van Data

De derde fase bestaat uit het selecteren van de meest relevante attributen in de dataset die van toepassing zijn voor het onderzoek. Door het gebruik dimensionaliteit reductie en transformatiemethoden kan het aantal variabelen dat in overweging wordt genomen in de dataminingfase beperkt worden.

### 4. Datamining

Nadat een solide dataset werd opgebouwd start het dataminingproces. Het is in deze fase dat getracht wordt om de onderliggende verbanden in de data naar boven te brengen. Hier zijn tal van dataminingtechnieken beschikbaar zoals classificatie, regressie en clustering. Enkele van deze technieken zullen later besproken worden. De onderzoeker dient hierbij de meest gepaste methoden te selecteren die bruikbaar zijn met de structuur van de data, en een mogelijk antwoord kunnen bieden op de initiële onderzoeksvragen.

### 5. Interpretatie en evaluatie

De laatste fase in het KDD proces bestaat uit het interpreteren van de gevonden resultaten en de toepasbaarheid van de gevonden resultaten op de onderneming.

Het KDD proces is een iteratief proces. Eens de gedistilleerde kennis en verbanden geïnterpreteerd worden door de onderzoeker kan ervoor gekozen worden om de modellen verder te verfijnen, de data verder te transformeren of om nieuwe databronnen toe te voegen om tot betere resultaten te komen.

Datamining is binnen het KDD proces een van de belangrijkste stappen. Hoewel elke stap in het KDD proces met zorg dient uitgevoerd te worden is het in de datamingfase waar de mogelijk nuttige verbanden aan het licht worden gebracht. Een nadeel van de traditionele dataminingtechnieken is dat deze nood hebben aan gestructureerde data. Meer en meer echter wordt er ook tekstuele data opgeslagen zoals emails, klachtenformulieren, medische verslagen,… etc. Deze data heeft vaak de grootste informatiewaarde maar kan niet meteen gebruikt worden voor datamining. Tekst Mining is een relatief nieuw domein dat tracht om de informatiewaarde in tekst te benutten. Vooral in de voorbereidende fase van het transformeren van data zijn specifieke methoden en technieken noodzakelijk. Voor het ontdekken van patronen in de data worden gelijkaardige technieken gebruikt als bij datamining. Er zal daarom eerst kort een beschrijving gegeven worden van het dataminingproces en enkele van de meest gebruikte dataminigtechnieken. Vervolgens wordt het verschil tussen Data- en Text Mining bekeken en worden enkele van de meest gebruikte Text Mining technieken beschreven.

## 3.2 Data Mining

Data Mining kan gezien worden als het vinden van impliciete, voorheen onbekende en mogelijk waardevolle informatie in data (Cattral, Oppacher, & Deugo, 2001).

### 3.2.1 Wat kan datamining ontdekken?

Het type van patronen dat ontdekt kan worden aan de hand van datamining is sterk afhankelijk van de gebruikte dataminingmethoden. Er zijn twee grote groepen in een datamining onderzoek. Beschrijvende datamining (*desciptive datamining*) waarbij getracht wordt om de algemene verbanden in een verzameling data te beschrijven, en voorspellende datamining (*predictive datamining*) waarbij getracht wordt om een voorspelling te doen op basis van de verzamelde kennis uit de dataset (Zaïane, 1999).

## 3.3 Dataminingtechnieken

### 3.3.1 Supervised en unsupervised learning

Binnen de datamingtechnieken zijn er twee grote groepen te onderscheiden. Technieken die gebruik maken van *supervised* en *unsupervised learning*.

#### 3.3.1.1 Supervised Learning

Supervised learning  maakt gebruik van een training set van bekende voorbeelden. Elk voorbeeld is een paar van een input object (vaak een vector) en een bekende output waarde. Het resultaat van een supervised learning algoritme is een functie die een discrete waarde kan aannemen (*classifier*) of een continue waarde (*regressiefunctie*). De afgeleide functie tracht de correcte output te vinden voor een nieuw onbekend voorbeeld. (Wikipedia) (Fayyad, Gregory, & Padhraic, 1996)

#### 3.3.1.2 Unsupervised Learning

Unsupervised learing maakt geen gebruik van een gelabelde trainging set. Een unsupervised learning algoritme probeert een verborgen structuur te vinden in een verzameling data. (Wikipedia)

### 3.3.2 Enkele vaak gebruikte dataminingtechnieken

Hieronder worden kort enkele dataminingtechnieken besproken. Enkelen worden meer in detail besproken in het hoofdstuk Text Mining.

#### *Beslissingsbomen*

Een beslissingsboom genereert een set van beslissingsregels in een procesvorm. Het bestaat uit een reeks van knooppunten waarbij op elk knooppunt een evaluatie wordt gemaakt. Afhankelijk van de uitkomst wordt een bepaald pad genomen totdat de onderkant van de beslissingsboom bereikt is. Het grote voordeel van beslissingsbomen is dat deze gemakkelijk te interpreteren zijn (CBIG). Een beslissingsboom heeft nood aan een set van voor-geclassificeerde observaties om een model te genereren. Het is een eerste voorbeeld van een model op basis van supervised learning.

### *Bayes Classificatie*

Bayes Classificatie is een tweede voorbeeld van supervised learning. Het maakt gebruik van voorwaardelijke kansen om de meest waarschijnlijke categorie te bepalen. De Bayes Classificatiemethode is genoemd naar Thomas Bayes de grondlegger van De Bayes Rule.

### *Neurale Netwerken*

Neurale netwerken bevatten een intergeconnecteerd netwerk van input en output knooppunten waarbij elke verbinding een initieel gewicht krijgt toegekend. Tijdens de trainingfase past het neurale netwerk gradueel de gewichten aan om betere voorspellingen te kunnen maken.

### *Support Vector Machines*

Support Vector Machines is een van de meest recente technieken binnen datamining. In zijn meest elementaire vorm tracht een Support vector Machine om een scheidingslijn te vinden die de maximale scheiding tussen twee groepen weergeeft. Support Vector machines worden in het hoofdstuk classificatietechnieken nog uitvoerig besproken.

### *Clustering*

Hiërarchische clustering en K-means clusteringzijn voorbeelden van unsupervised learning technieken. Deze technieken hebben geen noodzaak aan initiële training. Clusteralgrotitmen trachten om een verzameling observaties op te telen in homogenen groepen op basis van hun gemeenschappelijke kernmerken.

### *Regressie*

Regressie is een van de meest elementaire dataminingtechnieken. Bij een regressie analyse wordt er getracht om aan de hand van een numerieke dataset een formule op te stellen die de data het best beschrijft. Deze formule kan gebruikt worden om voorspellingen te maken.

## 3.4 Text Mining

Text Mining is in essentie datamining toegepast op tekst. In de eerste onderzoeken naar Text Mining werd daarom vaak gerefereerd naar de term *"text data mining"* (Hearst, 1999). In tegenstelling tot datamining waarbij gewerkt wordt met gestructureerde data is tekst ongestructureerd en moeilijk te verwerken. Desalniettemin is ongestructureerde tekst één van de meest gebruikte vormen van communicatie en is de waarde van de informatie die uit tekst gehaald kan worden aanzienlijk, zelfs met methoden die slechts gedeeltelijke resultaten opleveren (Sharp, 2001). Hoewel Text Mining een relatief nieuw begrip is bouwt het voort op heel wat voorgaand onderzoek waaronder *Information Retrieval* en *Natural Language Processing*.

### 3.4.1 De geschiedenis van Tekst Mining

#### 3.4.1.1 Information Retrieval (IR)

De opkomst van Information Retrieval (IR) is te situeren begin jaren '60 door de beschikbaarheid van de eerste computers die in staat waren om te gaan met ongestructureerde tekst. Veel van deze systemen waren grote mainframes die moeilijk bevraagbaar waren. De echte doorbraak kwam er pas in de jaren '80 door de introductie van *'client computersystemen'*.

De eerste Information Retreival systemen waren zogenoemde *'boolean systemen'* waarbij een gebruiker van het IR systeem een query kon opbouwen door het gebruik van boolean operators zoals AND, OR en NOT (Singhal, 2001). Boolean systemen hebben heel wat tekortkomingen. Zo is er geen onderliggende methode voor het rangschikken van gevonden documenten en het is niet altijd gemakkelijk om de juiste query te schrijven. Experten in boolean systemen waren in staat om snel zeer effectieve queries te schrijven voor het ophalen van documenten maar voor beginnende gebruikers was er een steile leercurve die veel *'trial and error'* inhield (Konchady, 2006).

Halverwege de jaren '90 werden de eerste grote vorderingen gemaakt in IR systemen die in staat zijn om documenten op te zoeken op basis van een vraag geformuleerd in natuurlijke taal. Dit is ook de periode waarin de eerste zoekrobots (*search engines*) voor informatie op het Internet hun intrede deden (Seymour, Frantsvog, & Kumar, 2011). Moderne IR systemen beschikken over een onderliggend systeem om de gevonden documenten te ordenen. Een IR systeem moet hiervoor een berekening maken van de mogelijke informatiewaarde per document op basis van de zoektermen.

#### 3.4.1.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) is ontstaan vanuit het domein van Artificiële Intelligentie (AI). Het doel was om een systeem te ontwikkelen dat in staat was om te communiceren in natuurlijke taal. Hiervoor dient het systeem in staat te zijn om natuurlijke taal te verstaan en te genereren. Het ontwikkelen van een systeem dat in staat is om natuurlijke taal te interpreteren bleek al snel een moeilijke opgave. Toch is het mogelijk om bruikbare toepassingen te ontwikkelen zonder de volledige betekenis van alle tekst te kennen.

#### 3.4.1.3 Text Mining

Text Mining tracht de methoden uit IR en NLP te gebruiken voor meer complexere systemen zoals bijvoorbeeld het groeperen van documenten in thema's, het samenvatten van teksten en vraag en antwoordsystemen.

### 3.4.2 Verschil tussen Tekst Mining en Data Mining

Bij datamining zit de informatie verborgen in de input data. De informatie is onbekend en zou moeilijk geëxtraheerd kunnen worden zonder de hulp van geautomatiseerde dataminingtechnieken. In tegenstelling tot bij datamining is bij Text Mining de informatie expliciet terug te vinden in de tekst. Heel wat schrijvers besteden de grootste zorg om teksten zo duidelijk en ondubbelzinnig mogelijk neer te schrijven. Het is door de hoeveelheid aan tekst dat we deze informatie toch ook als 'onbekend' kunnen bestempelen omdat het onmogelijk is om alle teksten in een bepaald domein zelf te lezen (Witten, 2005). De uitdaging bij Text Mining is het ontwikkelen van methoden om tekst op een geautomatiseerde wijze om te zetten naar een vorm die geïnterpreteerd kan worden door computers zodat vergelijkbare technieken als bij datamining kunnen worden toegepast.

Zoals eerder beschreven omvat het dataminingproces volgende vijf grote stappen:

1. Selectie van data
2. Voorbereiden van data (preprosessing)
3. Transformeren van data
4. Data Mining
5. Interpretatie en evaluatie

Text Mining volgt een gelijkaardig stappenproces maar voegt bij de voorbereidende fase het complexe proces van '*feature extraction*' toe. De kardinaliteit van een verzameling features gebaseerd op een verzameling documenten kan zeer hoog zijn en loopt gemakkelijk op tot enkele duizenden verschillende features (Dörre, Gerstl, & Seiffert, 1999). Deze eigenschap heeft twee consequenties voor het textminingproces:

1. Het is niet langer mogelijk om manueel na te gaan of een bepaalde feature gebruikt zal worden in het verdere proces of niet. Hiervoor zullen enkele geautomatiseerde methoden moeten geïmplementeerd worden zoals bijvoorbeeld het detecteren van irrelevante features die beschouwd kunnen worden als ruis voor de dataset.
2. De feature vectoren zijn vaak hoog dimensionaal en bevatten relatief weinig observaties wat een effect heeft op de analyse van de distributies. Vaak is een aangepaste implementatie noodzakelijk van de analytische modellen gebruikt in datamining om deze te kunnen toepassen op een Text Mining probleem.

### 3.4.3 Data Preprocessing in Text Mining

#### 3.4.3.1 Genereren van tokens

De eerste stap in data preprocessing voor Text Mining is het genereren van tokens. Het doel is om een reeks van karakters in een document op te splitsen in woorden of tokens. Deze stap lijkt relatief eenvoudig maar er dienen keuzes gemaakt te worden betreffende de plaatst waar de tekst moet opgesplitst worden. Bij elke zin? Elk woord? En wat met afkortingen en regelafbrekingen? Nadat een lijst van tokens in samengesteld dienen er nog enkele stappen ondernomen te worden om de lijst van initiële tokens op te kuisen en de kwaliteit ervan te verbeteren voor Text Mining toepassingen waaronder het verwijderen van stopwoorden en eventueel het toepassen van een woordstemming algoritme.

#### 3.4.3.2 Stopwoorden

Stopwoorden zijn woorden die in gesproken of geschreven taal vaak gebruikt worden zonder dat deze veel betekenis inhouden. Voorbeelden van woorden met een hoge frequentie van voorkomen in het Engels zijn *the, of, and*. Deze woorden voegen geen waarde toe aan een index en worden daarom vaak weggelaten om de performantie van een model te verbeteren. Een methode om stopwoorden te herkennen in een corpus is door de frequentie van alle woorden bij te houden en deze vervolgens naargelang hun frequentie te rangschikken en de *x%* woorden met de hoogste frequentie te verwijderen uit de index. (Konchady, 2006)

De lijst van stopwoorden die verder in deze masterproef wordt aangehouden is de volgende:

> *about add ago after all also an and another any are as at be because been before being between both but by came can come could did do does due each else end far for from get got has had have he her here him himself his how if into is it its just let lie like low make many me might must my never no nor not now of off old on only or other out over per pre put re said same see she should since so some still such take than that the their theirs them themselves then there these they this those through to too under up use very via want was way we well were what when where which while who will with would yes yet you your*

De lijst van stopwoorden is verschillend voor elke taal. Al naargelang de NLP toepassing dient dus aan andere lijst van stopwoorden aangehouden te worden (Ning, 2005).

##### *3.4.3.2.1 Foutief gespelde stopwoorden*

Foutief gespelde stopwoorden zijn geen echte woorden maar foutief gespelde woorden. Een persoon die een tekst leest kan een foutief gespelde woord vaak probleemloos herkennen en interpreteren. Maar voor een NLP toepassing is het herkennen van foutief gespelde woorden veel moeilijker. Een mogelijkheid om met deze foutief gespelde woorden om te gaan is om deze toe te voegen aan de lijst met stopwoorden en deze verder niet meer te gebruiken in de toepassing. Woorden die in een voldoende grote corpus maar een of slechts enkele keren voorkomen hebben een grote kans om foutief gespelde woorden te zijn.

#### *3.4.3.2.2 Domeinspecifieke stopwoorden*

Domeinspecifieke woorden zijn woorden die in normale gesproken of geschreven taal niet tot de stopwoorden behoren maar dit wel kunnen worden wanneer een verzameling van documenten over een bepaald domein gaat. Wanneer we bijvoorbeeld een verzameling productbeoordelingen hebben van consumentenelektronica dan is het woord *'telefoon'* geen stopwoord. Er zijn immers ook beoordelingen over computers, tv's, koelkasten,… Wanneer we echter een dataset van productbeoordelingen hebben met alleen maar beoordelingen over mobiele telefoons dan wordt 'telefoon' wel een stopwoord. Andere mogelijkheden in dit voorbeeld kunnen *'mobieltje'* of *'gsm'* zijn.

Wanneer in een voldoende grote corpus een woord in meer dan 80% van de documenten voorkomt heeft het een grote kans om een stopwoord te zijn en zou het verwijderd moeten worden. Dergelijke veel voorkomende woorden bieden geen toegevoegde waarde bij het classificeren van documenten (Ning, 2005).

### 3.4.3.3 Stemming

Bij stemming wordt getracht om aan de hand van een algoritme de stam van een woord te vinden. De stam van een woord is het deel dat overblijft wanneer alle buigingsvormen zijn weggenomen. Door woorden die een verschillende afleiding hebben te herleiden tot een gemeenschappelijke stamvorm daalt het aantal unieke tokens in de corpus wat de opslag voor de textmining toepassing beperkt. Het gebruik van stemming verhoogt ook de recall van een toepassing.

De meeste NLP toepassingen gebruiken een geautomatiseerde manier om woorden te stemmen. Enkele mogelijkheden hiervoor zijn:

#### *3.4.3.3.1 Table Lookup*

Deze methode tracht een woord te stemmen aan de hand van een tabel waarin alle basiswoorden te vinden zijn samen met hun afleidingen. Het grote voordeel van deze methode is dat het stemmen van woorden aan de hand van een opzoeking in een tabel zeer snel uitgevoerd kan worden. Het grootste nadeel is dat een volledige database van alle basiswoorden en afgeleide woorden voor geen enkele taal beschikbaar is. Zelfs niet voor het Engels. Er kunnen ook problemen opduiken tijdens het stemmingsproces wanneer er in documenten niet-standaardwoorden gebruikt worden of domeinspecifieke woorden (Stein & Potthast, 2007).

#### *3.4.3.3.2 Truncate Stemming*

Truncate Stemming beperkt de lengte van een woord tot de eerste $k$ letters. Wanneer een woord minder dan k letters heeft blijft het ongewijzigd.

#### *3.4.3.3.3 Successor Variety*

Successor Variety stemming is een aangepaste vorm van truncate stemming waarbij een woordspecifieke $k$ wordt berekend. Via een heuristische methode wordt nagegaan wat de frequentie en lengte is van een bepaald prefix. Bijvoorbeeld de woorden *'connection'*, *'connect'*, *'connectivity'* en *'connected'* hebben het prefix *'connect'* gemeenschappelijk (Stein & Potthast, 2007).

#### 3.4.3.3.4 N-gram Stemmers

N-gram Stemming maakt gebruik van overlappende sequenties van n karakters. Door het gebruik van deze methode kunnen veel van de voordelen van stemming behaald worden onder voorafgaande kennis van het taalgebruik in de documenten. Dit is omdat sommige van de gegenereerde n-grams het deel van een woord bevatten zonder affixen (Mayfield & McNamee, 2003).

#### 3.4.3.3.5 Affix Removal Stemmers

Stemmingsalgoritmen op basis van affix removal maken gebruik van een lijst van pre- en of suffixen en een lijst van beslissingsregels om een woord tot de stam te herleiden. Enkele voorbeelden van pre en suffixen in het Engels zijn:

> ***suffixen:*** *ly, ness, ion, ize, ant, ent , ic, al , ical, able, ance, ary, ate, ce, y, dom , ed, ee, eer, ence, ency, ery, ess, ful, hood, ible, icity, ify, ing, ish, ism, ist, istic, ity, ive, less, let, like, ment, ory, ty, ship, some, ure*
>
> ***prefixen:*** *anti, bi, co, contra, counter, de, di, dis, en, extra, in, inter, intra, micro, mid, mini, multi, non, over, para, poly, post, pre, pro, re, semi, sub, super, supra, sur, trans, tri, ultra, un*

De meeste stemmingalgoritmen laten de prefixen intact omwille van het vaak grote verschil in betekenis van een basiswoord met of zonder prefix.

### Lovins Method

De Levins Stemmer werd ontwikkeld door Julie Beth Lovins verbonden aan het Massachusetts Institute of Technology in 1968. Het algoritme maakt gebruik van een lijst van 297 suffixen en tracht steeds de langst mogelijke suffix te verwijderen, rekening houdend met enkele contextuele beperkingen en een minimale stamlengte van 3 karakters (Lovins, 1968).

De Lovins Stemmer wordt soms iteratief toegepast. Hierbij wordt een token enkele keren tot een stamvorm herleid totdat er geen wijziging meer is. Bijvoorbeeld het woord *'dictionary'* wordt tijdens de eerste iteratie herleid tot *'diction'*. In de tweede iteratie wordt dit herleidt tot *'dict'*. Hoe vaker het stemmingsalgoritme herhaald wordt, hoe kleiner het aantal unieke tokens in het document (Paynter, Cunningham, & Buchanan, 2000).

### Porters Stemming Algorithm

Porters Stemming Algorithm werd in 1980 ontwikkeld door Porter en is een van de meest gebruikte stemmingsmethoden. De Stemmer maakt gebruik van een reeks gedefinieerde regels die iteratief suffixen verwijderen van een woord.

Soms kan het voorkomen dat bij het gebruik van een stemmingsmethode een foutieve stamvorm wordt gegenereerd. Dit hoeft niet steeds een probleem te zijn. Bijvoorbeeld bij het classificeren van documenten heeft foutieve stemming geen impact op het classificatieresultaat. De correctheid van de stamvormen is wel van belang wanneer deze in een later stadium gebruikt zullen worden in een toepassing die interactie vraagt van de gebruiker (Buss).

Een belangrijk aandachtspunt bij het selecteren van een stemmingsalgoritme is dat een aantal methoden zeer taalspecifiek zijn. Zo bevat het Lovins en Porters Stemming algoritme lijsten van suffixen en bewerkingsregels die specifiek voor de Engelse taal ontwikkeld werden. Stemming op

basis van Truncate Stemming, Successor Variety en N-gram Stemming zijn niet taalafhankelijk al kan het succes van deze methoden sterk verschillen van taal tot taal.

### 3.4.4 Functies van Tekst Mining

Tekstmining heeft heel wat mogelijke toepassingen. Deze zijn onder te verdelen in 6 grote groepen van functies (Konchady, 2006):

**Zoekfunctie:** Een zoekfunctie is een van de meest essentiële elementen voor het bruikbaar maken van een verzameling documenten en informatie.

**Information Extraction:** Heel wat documenten binnen een bepaald domein hebben een gelijklopende structuur. Bijvoorbeeld online zoekers, profielen op sociale netwerken en ook productbeoordelingen. Wanneer deze patronen geautomatiseerd ontdekt kunnen worden verhoogd dit de waarde van de verzamelde data.

**Categoriseren / clusteren:** Het is eenvoudiger om documenten te analyseren wanneer deze onderverdeeld zijn in groepen per domein of thema.

**Samenvatten:** Bij het samenvatten wordt getracht om delen tekst van verschillende documenten te combineren zodat de gebruiker van een toepassing informatie zo compact mogelijk gepresenteerd krijgt.

**Monitoren van informatie:** Het aanbod aan online nieuwssites, blogs, fora e.d. neemt nog elke dag toe. Het is onmogelijk om de informatie van al deze online bronnen manueel te verwerken.

**Vraag en Antwoord:** Heel wat mensen maken momenteel gebruik van keyword gebaseerd zoeken. Soms willen we echter een vraag kunnen invullen in natuurlijke taal. Een intelligente zoekrobot is in staat om deze vraag te herkennen en op basis van Tekst Mining een antwoord te genereren.

In het volgende hoofdstuk zal de classificatiefunctie verder besproken worden door enkele van de meest gebruikte classificatiemethoden in Text Mining toepassingen te bespreken. De lijst van classificatietechnieken is zeker niet volledig maar bevat wel de meest voorkomende technieken. De classificatietechnieken zijn vergelijkbaar met de technieken die traditioneel gebruikt worden in Data Mining. De theorie achter de modellen is dan ook voor zowel Data als Text Mining vergelijkbaar maar de modellen zullen besproken worden vanuit het Tekst Mining perspectief.

# 4 Classificatietechnieken

Op het eerste zicht lijkt het toewijzen van een document aan een bepaalde groep weinig informatiewaarde toe te voegen, maar niets is minder waar. Een korte illustratie maakt dit meteen duidelijk. Een bedrijf heeft bijvoorbeeld een nieuw product op de markt gebracht en wil opvolgen hoe consumenten hun product beoordelen. Het bedrijf heeft hiervoor een grote lijst van online blogs samengesteld die producten zoals die van bedrijf x beoordeeld en waarbij bezoekers van de blog ook de mogelijkheid hebben om op het artikel te reageren. Omdat het aantal blogs en aantal artikelen dat op elk van deze blogs verschijnt zeer groot is, is het bijna onmogelijk om deze blogs handmatig op te volgen. Des te meer is dit het geval wanneer er regelmatig nieuwe reacties van bezoekers bijkomen. Een mogelijke oplossing is om deze lijst van blog geautomatiseerd op te volgen met classifiers. Een eerste classifier zou bijvoorbeeld kunnen detecteren wanneer een nieuw artikel over een product van bedrijf x gaat. Een tweede classifier gaat vervolgens na of het nieuwe product positief of negatief onthaald wordt. Dit is maar een van de talloze voorbeelden van nuttige toepassingen van classificatie. De classificatiemethode die als eerste hieronder besproken wordt is een classificatie op basis van Naïve Bayes.

### 4.1.1 Naïve Bayes Classifier

Een van de populairdere classificatietechnieken is Naïve Bayes. Deze maak gebruik van de theorie van voorwaardelijke kansen die werd geformuleerd door de wiskundige Thomas Bayes.

Wanneer er verder wordt gesproken over een Naïve Bayes Classifier wordt een Multinomial Näive Bayes (MNB) classifier bedoeld. Dit is een classifier die tracht een document toe te wijzen aan een van $k$ verschillende groepen ($k \in \mathbb{R}$). Ook wanneer er slechts in twee groepen geclassificeerd wordt (Binomiaal) wordt de term Multinomial aangehouden. Een Multinomial classifier is in feite een veralgemening van een Binomiaal classifier.

#### 4.1.1.1 De term Naïve in Naïve Bayes

De term “naïve” geeft aan dat de classifier er van uitgaat dat features in een tekst onafhankelijk zijn van elkaar. Dit wordt ook wel een *“bag of words model”* genoemd. Wanneer we de woorden in een document willekeurig van plaats zouden wisselen wordt het document net hetzelfde geclassificeerd als het originele document. Deze assumptie is in regel vals omdat in elke tekst de woorden niet onafhankelijk van elkaar geplaatst worden. Woorden maken deel uit van zinnen waarbij een combinatie van woorden een betekenis heeft. Ondanks dat deze assumptie altijd geschonden wordt halen classifiers op basis van Naïve Bayes in de praktijk zeer goede resultaten.

Omdat de assumptie van onafhankelijkheid steeds geschonden wordt kunnen we geen rechtstreeks gebruik maken van de berekende kans dat een document tot een bepaalde categorie behoort. Het is

echter wel mogelijk om de berekende kansen met elkaar te vergelijken. Een Naïve Bayes classifier wordt daarom onder generatieve modellen gerekend. Voor elke categorie wordt een model opgesteld en wordt het resultaat berekend. Het model met het beste resultaat wordt vervolgens gekozen.

#### 4.1.1.2 Bayes Rule

Een Naïve Bayes Classifier is gebaseerd op de *Bayes Rule*. Ook wel de *regel van Bayes*, *Bayes law* of *Bayes theorem* genoemd. Bayes Rule is een regel uit de kansrekening waarmee het omgekeerde van een voorwaardelijke kans bekerend kan worden.

De kans dat twee gebeurtenissen A en B onafhankelijk van elkaar gebeuren kan geschreven worden als:

$$P(A \cap B) = P(A|B).P(B)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

De kans op een gebeurtenis B gegeven een gebeurtenis A kan geschreven worden als:

$$P(B|A) = P(A|B).P(B)$$

$$P(A|B) = \frac{P(B \cap A)}{P(A)}$$

Wanneer deze twee vergelijkingen gecombineerd worden komen we tot Bayes Rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Toegepast op de classificatie van een document:

$$P(Categorie|Document) = \frac{P(Document|Catgeorie)P(Categorie)}{P(Document)}$$

*P(Categorie)* is de kans dat een willekeurig document behoort tot een bepaalde categorie. *P(Categorie)* kan dus eenvoudig worden berekend door het aantal documenten in een categorie te delen door het totaal aantal documenten.

*P(Document)* hoeft strikt genomen niet berekend te worden omdat bij een Naïve Bayes classificatie gebruikt gemaakt word van een relatieve classificatie. De waarde van de noemer heeft bijgevolg geen impact op de uiteindelijke classificatie.

#### 4.1.1.3 Multinomial Naïve Bayes Classifier

Een Naïve Bayes Classifier start eerst met de kans te berekenen dat een bepaald token, vaak een woord, tot een bepaalde categorie behoort. De kans dat een token $w_i$tot een bepaalde categorie behoort kan als volgt berekend worden:

$$P(w_i|Categorie) = \frac{N(w_i, Categorie) + 1}{N(w_i) + k} = \frac{\begin{matrix} aantal\ keer\ dat\ token\ w_i \\ voorkomt\ in\ de\ categorie \end{matrix} + 1}{\begin{matrix} aantal\ keer\ dat\ token\ w_i \\ voorkomt\ in\ alle\ categorieën \end{matrix} + \begin{matrix} aantal \\ categorieën \end{matrix}}$$

De term 1 in de teller en $k$ in de noemer zijn *smoothing parameters*[6]. Hier wordt Laplace smoothing toegepast. De Laplace smooting paramater in de teller zorgt ervoor dat het product van alle kansen niet 0 is wanneer een van de tokens niet voorkomt in een categorie. Zonder de Laplace smooting zou er een kans $P(w_i|Categorie)$=0 zijn waardoor het berekende product van de kansen voor elke $w_i$ in het document ook 0 wordt. De Laplace smoothing parameter in de noemer voorkomt een deling door 0 wanneer een token niet voorkomt in de training data. Wanneer er een onbekend token voorkomt in de testdata krijgt het token een gelijke kans om tot elk van de categorieën te behoren. Bij bijvoorbeeld 5 categorieën is $P(w_i|Categorie)$voor elke categorie 0,2. Deze arbitraire toewijzing van kansen lijkt intuïtief aanvaardbaar te zijn.

De *Naïve Base Estimate* voor een volledig document wordt geschreven als:

$$P(Categorie\ |Document) = \frac{P(Categorie) * \prod_{i=1}^{2} P(w_i\ |Categorie)}{P(Document)}$$

Omdat de berekende kansen in tekstclassificatietoepassingen over het algemeen vrij laag zijn en bijgevolg het product van de kansen snel naar 0 gaat wordt vaak ook de som van het logaritme van elk kans genomen.

$$\sum_{i=1}^{n} log\left(\frac{N_{ci} + \alpha_i}{N_c + \alpha}\right)$$

#### 4.1.1.4 Verbeteren van de Naïve Bayes Classifier

##### *4.1.1.4.1 Complement Naïve Bayes (CNB)*

Een grote tekortkoming van de basis Naïve Bayes Classifier is dat deze vrij gevoelig is voor *'skewed data'*. Dit is wanneer er meer training data is voor een categorie ten opzicht van een andere. Het resultaat van de Naïve Bayes Classifier leunt steeds in de richting van een categorie met meer training data. Dit kan in een gecontroleerde training set tegengegaan worden door de training set zo te selecteren dat alle categorieën in gelijke mate vertegenwoordigd zijn.

Bij het gebruik van een Naïve Bayes Classifier in een productieomgeving zal het niet steeds mogelijk zijn om aan deze voorwaarde te voldoen. Verder is het soms wenselijk om zo veel mogelijk training data te gebruiken met het oog op een beter classificatieresultaat ook al betekent dit de facto dat niet elke categorie een gelijke grootte zal hebben. Ook wanneer het aantal documenten in beide

[6] Een uitgebreidere beschrijving van *'smoothing'* is terug te vinden in het glossarium.

categorieën gelijk is maar de gemiddelde documentgrootte per categorie verschillend is, kan er sprake zijn van *'skewed data bias'*.

Een oplossing voor deze ongelijke representatie is Complement Naïve Bayes (CNB). Bij de standaard Naïve Bayes werd er gebruik gemaakt van training data in een enkele categorie $x$. Bij CNB wordt alle training data gebruikt met uitzondering van categorie $x$. In essentie wordt dezelfde hoeveelheid data gebruikt als bij MNB alleen is de manier waarop de data wordt gebruikt bij CNB minder vatbaar voor *'skewed data bias'*.

De CNB estimate voor een token is:

$$\frac{N_{\tilde{c}i} + \alpha_i}{N_{\tilde{c}} + \alpha}$$

De CNB estimate voor een volledig document is

$$\sum_{i=1}^{n} \log\left(\frac{N_{\tilde{c}i} + \alpha_i}{N_{\tilde{c}} + \alpha}\right)$$

Waarbij $N_{\tilde{c}i}$ het aantal keer is dat token $i$ voorkomt in documenten in categorieën verschillend van $c$. $N_{\tilde{c}}$ is de totale frequentie van woorden in categorieën verschillend van $c$. De parameters $\alpha_i$ en $\alpha$ zijn de *'Laplace Smoothing'* parameters. Meestal wordt voor $\alpha_i$ de waarde 1 genomen en is $\alpha$ gelijk aan het aantal categorieën verbonden aan de classifier.

Het grondtal van het logaritme is hier *e* maar het gebruikte grondtal heeft geen effect op de uiteindelijke classificatie.

Omdat Complement Naïve Bayes meer data in overweging neemt dan Multinomial Naïve Bayes is CNB aan te raden bij kleinere datasets omdat er sneller een betrouwbaar resultaat bekomen zal worden.

#### *4.1.1.4.2 Weighted Complement Naïve Bayes (WCNB)*

Een bijkomend nadeel van een standaard Naïve Bayes classifier is dat deze de categorieën bevoordeelt die de assumptie van onafhankelijkheid van features het meest schenden. Wanneer we bijvoorbeeld documenten willen classificeren per Amerikaanse stad en we hebben een dataset met volgende eigenschappen:

- De termen *"San Diego"* en *"Boston"* komen in een gelijke frequentie voor in de dataset.
- De termen *"San"* en *"Diego"* komen zelden apart van elkaar voor.

Wanneer we een nieuw document zouden willen classificeren waarin *"San Diego*" 3 keer voorkomt en *"Boston"* 5 keer dan zal de Naïve Bayes classifier het document verkeerdelijk toewijzen aan de categorie *"San Diego"*. De verklaring hiervoor is dat de Naïve Bayes Classifier de termen *"San"* en *"Diego"* als onafhankelijk van elkaar ziet en deze bijgevolg dubbel in overweging neemt.

#### 4.1.1.5 Sterkten en zwakten van een Naïve Bayes Classifier

Een van de grootste voordelen van classifiers op basis van Naïve Bayes is de snelheid waarmee deze getraind kunnen worden en de snelheid waarmee de classificatie kan worden uitgevoerd. Zelfs met een enorm uitgebreide dataset zijn er meestal maar een beperkt aantal features die in overweging genomen moeten worden bij het trainen of classificeren van een nieuw document. De snelheid waarmee een nieuw document aan de trainingset kan worden toegevoegd is ook een groot voordeel wanneer het model gradueel wordt getraind (Segaran, 2007).

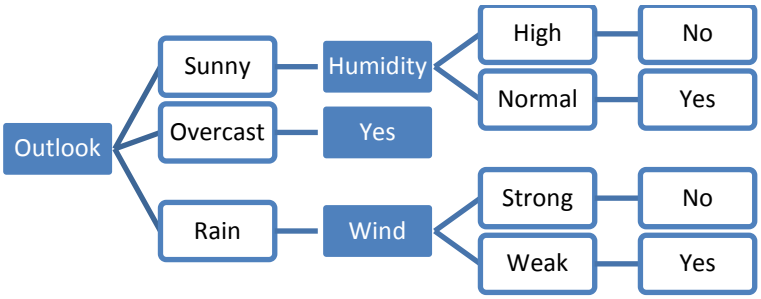
Een ander groot voordeel van Naïve Bayes is zijn eenvoudige opzet en het gemak waarmee het model geïnterpreteerd kan worden. De Naïve Bayes Classifier maakt gebruik van kansen om een document te classificeren. We kunnen deze kansen voor elk feature opzoeken en zo de meest discriminerende features bepalen. Deze kunnen helpen bij de interpretatie van de classificatie van een bepaald document of de eigenheid van een groep.

Een nadeel van Naïve Bayes is dat de classifier niet in staat is om te gaan met combinaties van features. Een van de mogelijke oplossing is gebruik maken van WCNB of het opnemen van n-grams in de classifier.

### 4.1.2 Beslissingsbomen

Een beslissingsboom is een geordende set van toewijzingsregels. Voor een bepaald document worden de verschillende knooppunten in de beslissingsboom gevolgd om uiteindelijk het document aan een categorie toe te wijzen. Elk tussenliggend knooppunt maakt gebruik van een bepaald token of feature om een test uit te voeren en om de beslissing te nemen in welke richting het pad van de beslissingsboom gevolgd moet worden. De bladeren van een beslissingsboom zijn de uiteindelijke categorieën waaraan een document kan worden toegewezen.

Bovenaan in een beslissingsboom vinden we de tokens terug met de grootste discriminatie tussen de categorieën.




#### 4.1.2.1 Sterkten en zwakten

Een van de grootste voordelen aan het gebruik van beslissingsbomen is het gemak waarmee deze geïnterpreteerd kunnen worden. Ook plaatst het algoritme de meest belangrijke beslissingsfactoren bovenaan in de beslissingsboom. Dit maakt dat een beslissingsboom niet allen nuttig is voor de classificatie van tekst maar ook voor interpretatie. Net als bij Naïve Bayes is het mogelijk om de echte werking van het model te bekijken en informatie te verzamelen hoe het werkt en waarom. Dit kan helpen bij het nemen van beslissingen buiten het classificatieproces (Segaran, 2007).

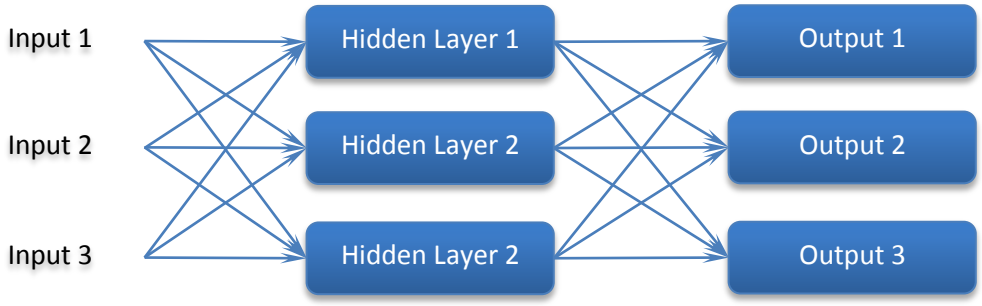
Beslissingsbomen kunnen zowel met numerieke als met categorische data gebruikt worden. Dit maakt dat beslissingsbomen breed inzetbaar zijn in classificatietoepassingen. Een beslissingsboom is echter vaak niet de beste oplossing voor gebruik met numerieke data. Het algoritme kan de data splitsen in gemiddelden met de laagste variantie, maar wanneer de data complex is zal de beslissingsboom zeer groot worden voor deze in staat is om accurate beslissingen te nemen.

Een voordeel van beslissingsbomen ten opzicht van Naïve Bayes is dat beslissingsbomen in staat zijn om te gaan met interacties van variabelen. Bijvoorbeeld: wanneer *feature 1* of *feature 2* voorkomt volg dan *pad A*. Wanneer *feature 1* en *feature 2* samen voorkomen volg dan *pad B*.

Een groot nadeel van een beslissingsboom is dat deze niet in staat zijn om op een later moment nieuwe trainingdata in overweging te nemen. Het is wel mogelijk om de originele dataset uit te breiden met nieuwe data maar het algoritme zou dan een volledig nieuwe beslissingsboom moeten samenstellen. Dit in tegenstelling tot Naïve Bayes waarbij het probleemloos mogelijk is om nieuwe observaties toe te voegen aan de dataset.

### 4.1.3 Neurale Netwerken

Neurale netwerken bestaan uit een verzameling knooppunten (neuronen) die onderling verbonden zijn door middel van 'gewogen' connecties (synapsen). In elk van deze neuronen wordt een berekening uitgevoerd. De output van het neuron is afhankelijk van de input die het krijgt van de voorgaande neuronen. (Kröse & Smagt, 1996)




In dit voorbeeld bestaat het netwerk uit twee lagen neuronen. De output van de eerste laag neuronen wordt doorgegeven aan de tweede laag door middel van de synapsen. Hoe hoger het gewicht van een bepaalde synaps, hoe groter de invloed die de connectie zal hebben op de output van het neuron. (Segaran, 2007)

#### 4.1.3.1 Trainen van een Neuraal Netwerk

De grote kracht van Neurale Netwerken is dat deze kunnen starten met willekeurige gewichten voor de synapsen en vervolgens zelfstandig leren tijdens het trainingsproces. De meest gebruikte methode voor het trainen van een Neuraal Netwerk is *backpropagation*. (Werbos, 1990)

Stel dat we een Neuraal Netwerk willen trainen op basis van *backpropagation* dat in staat is om een productbeoordeling als positief of negatief te classificeren. We starten met een eerste voorbeeld (in dit geval positief) en we laten het neurale netwerk een initiële gok doen. Wanneer de uitkomst van deze eerste gok een negatieve classificatie zou zijn dan worden de gewichten van de synapsen die naar de categorie *'negatief'* leiden lichtjes verlaagd. De synapsen die naar de categorie *'positief'* leiden worden lichtjes verhoogd. Op het einde van elke poging van het neurale netwerk om een document te classificeren krijgt het neurale netwerk het juiste antwoord te horen en wordt het pad dat naar dit juiste antwoord zou leiden licht verhoogd en de andere paden verlaagd.

Om te voorkomen dat een netwerk te gevoelig is voor woorden of features die maar enkele keren in de trainingdata voorkomen worden de gewichten slechts licht aangepast. Hoe vaker het netwerk een bepaald voorbeeld ziet, des te sterker zullen de connecties worden.

#### 4.1.3.2 Sterkten en zwakten van Neurale Netwerken

De grootste sterke van Neurale Netwerken is dat deze in staat zijn om te gaan met complexe niet lineaire functies en afhankelijkheden tussen verschillende inputs.

Neurale Netwerken zijn in staat om nieuwe trainingdata te ontvangen zonder dat de opbouw van een nieuw model noodzakelijk is. Ze kunnen vaak ook zeer compact opgeslagen worden omdat een Neuraal Netwerk enkel een lijst met nummers bevat met het gewicht van elke synaps. Bovendien is het niet noodzakelijk om de trainingdata bij te houden om het model te laten werken. Dit maak dat

Neurale Netwerken vaak gebruikt worden in situaties waarbij er een continue stroom is van nieuwe data.

Het grootste nadeel van Neurale Netwerken is dat deze een zogenaamde ‘black box’ methode zijn. Een netwerk kan honderden knooppunten hebben met duizenden synapsen waardoor het niet mogelijk is om meteen af te leiden hoe een bepaalde uitkomst berekend werd.

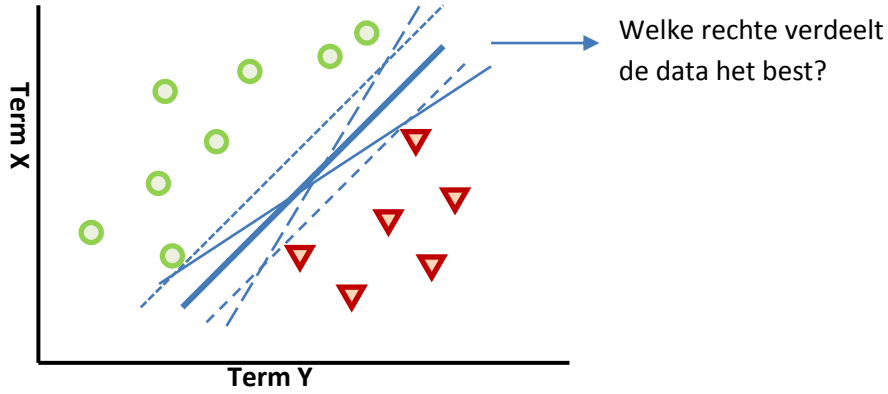
Een ander nadeel van neurale netwerken is dat er geen vaste richtlijnen zijn voor de grootte van de trainingset en de grootte van het netwerk voor een bepaald classificatieprobleem. Vaak moeten verschillende combinaties uitgeprobeerd worden om daarna het beste model te selecteren.

### 4.1.4 Support Vector Machines

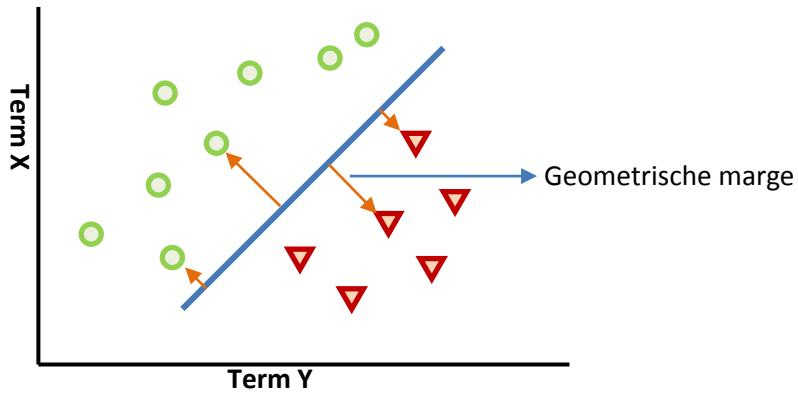
Een Support Vector Machine tracht een hypervlak te vinden die een numerieke dataset zo goed mogelijk kan scheiden. Om de logica achter Support Vector Machines duidelijk te maken zal eerste een eenvoudig voorbeeld worden geïllustreerd van een lineaire classificatie. Vervolgens zal dit verder worden uitgewerkt naar complexere niet-lineaire classificaties.

#### 4.1.4.1 Lineaire Classificatie

Er wordt in dit eenvoudige voorbeeld van de assumptie uit gegaan dat een trainingset lineaire deelbaar is. Wanneer we onderstaande scatterplot bekijken zijn er verschillende mogelijkheden om de data te verdelen aan de hand van een hypervlak[7]. Eigenlijk zijn er oneindig veel mogelijkheden om deze dataset te verdelen. Maar welk hypervlak  is het beste?

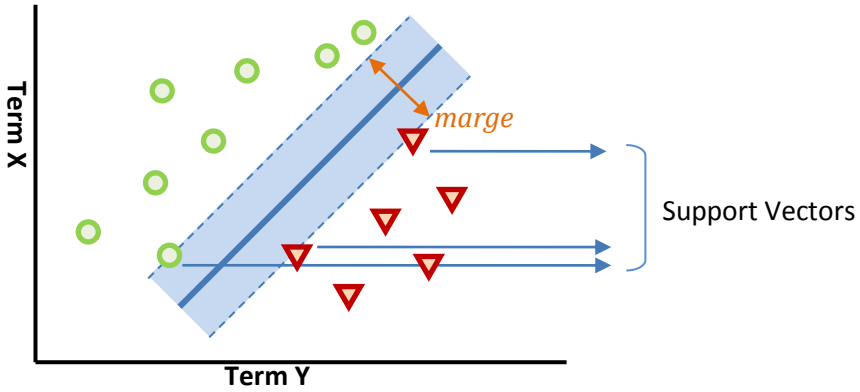



Intuïtief herkennen we het middelste dik gemarkeerde hypervlak als de beste scheiding. Deze rechte heeft een maximale afstand tussen de datapunten en scheidt de positieve van de negatieve observatie maximaal. De redenering is dat hoe verder een datapunt van het hypervlak ligt, hoe zekerder we er van kunnen zijn dat dit datapunt effectief tot de klasse behoord. Support Vector Machines passen eenzelfde logica toe bij het bepalen van het beste hypervlak voor lineaire classificatie. Hiervoor wordt gebruik gemaakt van *'geometrische marges'*. Een geometrische marge is de afstand van een observatie tot het hypervlak.




[7] Een hypervlak is een *n-1* subruimte van een *n* multidimensionale ruime. Het hypervlak in het voorbeeld van lineaire classificatie is een 1 dimensionaal hypervlak in een 2-dimensionale ruimte $R^2$.

Een Support Vector Machine tracht een hypervlak te vinden waarbij de geometrische marge maximaal is. De geometrische marge is de minimale afstand tussen het hypervlak die de twee klassen deelt en de korts bijgelegen datapunten van elke klasse (Campbell, 2008).




De punten op de stippellijn worden de support vectors genoemd. Ze worden zo genoemd omdat ze de ligging van het hypervlak ondersteunen. Wanneer we bijvoorbeeld alle driehoekige datapunten zouden wegdenken buiten de twee punten op de stippellijn, dan ligt het hypervlak met de maximale geometrische marge nog steeds op dezelfde plaats. Wanneer een van de datapunten op een van de twee evenwijdige hypervlakken wordt verwijderd dan zou de plaatsing van het hypervlak wijzigen.

#### 4.1.4.2 Optimal Margin Classifier

In een n-dimensionale ruimte kan een hypervlak geschreven worden als:

$$w.x + b = 0$$

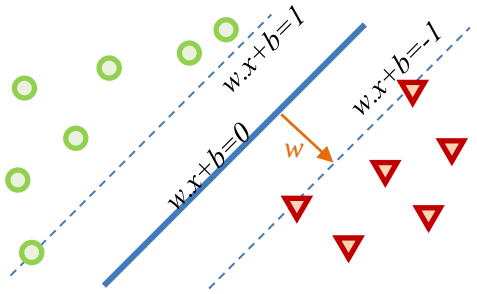
Waarbij $b$ ruis is (*bias*), $w$ het gewicht (*weight*) en $x$ de data. Wanneer alle documenten boven het hypervlak als positief geclassificeerd worden en alle documenten onder het hypervlak als negatief dan kan de beslissingsfunctie voor het classificeren van nieuwe documenten geschreven worden als volgt:

$$D(x) = sign(w.x + b)$$

Wanneer het resultaat van de berekening positief is, geeft de functie als resultaat +1 en wordt het document als positief geclassificeerd. Wanneer de uitkomst negatief is geeft de beslissingsfunctie -1 als resultaat en wordt het document als negatief geclassificeerd. Het is belangrijk om op te merken dat het resultaat van de classificatie enkel afhankelijk is van het teken van $w$ (positief of negatief), maar niet van het gewicht. Wanneer w en b gelijk herschaald worden heeft dit geen effect op het resultaat van de classificatie.

$resultaat\ D(x)\ is\ ongewijzigd\ onder\ herschaling\ w \rightarrow \lambda w\ , b \rightarrow \lambda b$ (Campbell, 2008)

Zoals eerder vermeld zijn we op zoek naar de grootst mogelijke marge. Deze marge wordt bepaald door de twee evenwijdige hypervlakken waarop de support vectors gelegen zijn. Voor het lineaire classificatieprobleem wordt bepaald dat de afstand $w$ van het hypervlak tot een support vector 1 is.




De vergelijking voor deze twee hypervlakken kan nu als volgt geschreven worden:

$$w.x + b = 1$$
$$w.x + b = -1$$

Wanneer we de tweede vergelijking van de eerste aftrekken krijgen we volgend resultaat:

$$w.(x_1 - x_2) = 2$$

De marge wordt gegeven door de projectie van de vector $(x_1 - x_2)$ op de normaalvector van het hypervlak $w/\|w\|$. Hieruit kunnen we afleiden dat de marge gegeven wordt door:

$$\gamma = \frac{2}{\|w\|}$$

Aangezien we een zo groot mogelijke marge willen, is het maximaliseren van $\gamma = 2/\|w\|$ gelijk aan het minimaliseren van $\|w\|$. Het optimaliseringsprobleem kan nu als volgt geschreven worden:

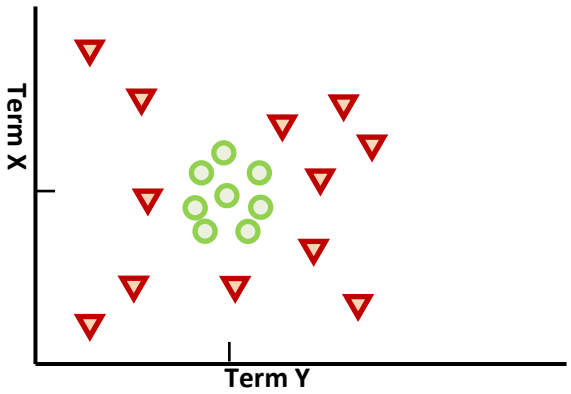
$$\min \quad \Phi(w) = \frac{1}{2}\|w\|^2$$

$$s.t. \quad y_i[(w.x_i) + b] \geq 1$$

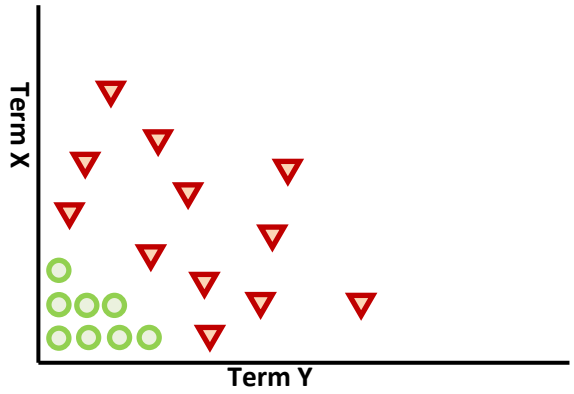
Het maximaliseringsprobleem is nu omgezet in een minimaliseringsprobleem dat opgelost kan worden aan de hand van kwadratisch programmeren als kan de rekentijd voor het oplossen van een dergelijk probleem snel oplopen wanneer er gewerkt wordt in een hoog dimensionale ruimte (Boser, Guyon, & Vapnik, 1992).

#### 4.1.4.3 De Kernel Trick

Stel dat we een dataset zouden hebben zoals hieronder weergegeven met twee categorieën. Wat zou dan het gemiddelde punt zijn voor elke categorie? Hier zou het gemiddelde punt voor beide categorieën exact hetzelfde punt zijn, namelijk in het midden van de plot. Het is dus niet mogelijk om de lineaire classifier te gebruiken om de twee categorieën van elkaar te scheiden.




Stel dat we voor elke observatie het kwadraat zouden nemen van elk x en y. Wanneer we de data nu weergeven op een plot ziet deze er uit als volgt:




De transformatie die hier werd uitgevoerd is de *polynomial* transformatie. Deze data kan nu wel weer door een rechte gescheiden worden. Om een nieuwe observatie te classificeren dient enkel het product van de $x$ en $y$ waarde voor deze observatie genomen worden om na te gaan aan welke zijde van de rechte de nieuwe observatie wordt toegewezen (Ng).

Hoewel deze methode vrij eenvoudig lijkt, is het bij de classificatie van echte documenten vaak veel moeilijker om een transformatiefunctie te vinden die in staat is om de dataset te scheiden in twee groepen. Hiervoor zou de dataset in honderden, zo niet duizenden, of zelfs oneindig veel verschillende dimensies getransformeerd moeten worden.

Om dit probleem te overkomen wordt de Kernel trick toegepast. In plaatst van de transformatie uit te voeren wordt de dot-productfunctie vervangen door een functie die weergeeft wat het dot-product zou zijn wanneer de data getransformeerd zou zijn in een andere ruime.

#### 4.1.4.4 Niet-lineaire classificatie

Een Kernel op basis van de dot-productfunctie gaat er nog steeds van uit dat de data lineair deelbaar is. Dit is vaak niet het geval. Het is echter ook mogelijk om elk dot-product te vervangen door een Kernel die een niet-lineaire transformatie uitvoert.

#### 4.1.4.5 Categorische data

Heel wat classificatiemethoden zijn in staat om met zowel numerieke als categorische data om te gaan zodat dat er voorafgaande bewerkingen aan de dataset noodzakelijk zijn. Support Vector Machines zijn echter niet in staat om rechtstreeks om te gaan met categorische data (Segaran, 2007). We moeten dus op zoek gaan naar een manier om deze data om te zetten naar een numeriek formaat zodat dit bruikbaar wordt voor een classifier op basis van Support Vector Machines.

##### *Voorbeeld Positief of negatief*

Op verschillende websites wordt getracht om zoveel mogelijk meningen van bezoekers te verzamelen door een eenvoudig beoordelingssysteem op te zetten met maar twee verschillende mogelijkheden. Op YouTube vinden we bijvoorbeeld onder elke video twee icoontjes waarmee de gebruiker kan aangeven of hij/zij deze video positief op negatief beoordeelt:


De eenvoudigste methode om dergelijke categorische data om te zetten naar numerieke data is elke positieve stem de waarde 1 te geven en elke negatieve stem de waarde -1. Wanneer een waarde wordt toegekend aan een bepaalde categorie is het aan te raden om na te gaan of het mogelijk is om een waarde toe te kennen met een mogelijke interpretatiewaarde. Het wordt iets moeilijker om dit te doen wanneer er bijvoorbeeld getracht wordt om een verzameling artikelen of dieren te classificeren per diersoort.

#### 4.1.4.6 Sterkten en zwakten van Support Vector Machines

Classificatie op basis van Support Vector Machines is een van de krachtigste aangehaalde methoden. Over het algemeen halen Support Vector Machines ook de beste classificatieresultaten. Eens een classifier voldoende getraind is, wordt de classificatie zeer snel uitgevoerd aangezien de classifier enkel moet nagaan aan welke zijde van de scheidingslijnen een nieuwe observatie zich bevindt.

Een nadeel aan support vector machines is dat de Kernel transformatiefunctie en de parameters voor deze functie verschillen voor elke dataset. Voor elke toepassing moeten deze dus opnieuw bepaald worden (Burges, 1998).

Over het algemeen zijn Support vector Machines het meeste geschikt voor grote datasets. Dit in tegenstelling tot bijvoorbeeld beslissingsbomen die ook al bij kleine datasets goede resultaten kunnen behalen.

Net als Neurale Netwerken is Support Vector Machines een ‘black box’ methode. Vaak worden complexe transformaties uitgevoerd in multidimensionale ruimten. Support Vector Machines kunnen dus wel goede resultaten behalen, de methode waarop is moeilijk te achterhalen of interpreteren (Segaran, 2007).

### 4.1.5 Validatie van een classifier

#### 4.1.5.1 Evaluatiecriteria voor een binaire classifier

We starten de bespreking van de evaluatiecriteria voor een binaire classifier met *categorie A* en *categorie B*. Voor de berekening van de meest courante evaluatiecriteria dienen we voor de documenten in de validatieset een optelling te maken van de goed-positieven, fout-positieven, goed-negatieven en fout-negatieven.

Goed-positieven (*true positive*) zijn documenten die door de classifier voorspeld zijn te behoren tot categorie A en effectief behoren tot categorie A. Fout-positieven (*false positive*) zijn documenten die door de classifier toegewezen zijn aan categorie A maar in werkelijkheid behoren tot categorie B. Goed-negatieven (*true negative*) zijn documenten die door de classifier correct worden toegewezen aan categorie B. Fout-negatieven (*false negative*) zijn documenten die verkeerdelijk werden toegewezen aan categorie B en in werkelijkheid tot categorie A behoren. De som van deze vier dient gelijk te zijn aan het totaal aantal documenten $N$ in de validatieset.

$$\sum goed\text{–}positief + \sum fout\text{–}positief + \sum goed\text{–}negatief + \sum fout\text{–}negatief = N$$

##### *Precision* ($\pi$)

Precision kan geïnterpreteerd worden als de kans dat een nieuw document $d$ dat geclassificeerd wordt onder categorie $c$, inderdaad tot categorie $c$ behoord (Sebastiani, 2002).

$$Precision(\pi) = \frac{\sum goed\text{–}positief}{(\sum goed\text{–}positief + \sum fout\text{–}positief)}$$

##### *Recall* ($\rho$)

Recall is de kans dan een document $d$ dat verwacht wordt geclassificeerd te worden onder categorie $c$ effectief aan categorie $c$ wordt toegewezen.

$$Recall(\rho) = \frac{\sum goed\text{–}positief}{(\sum goed\text{–}positief + \sum fout\text{–}negatief)}$$

##### *Accuracy*

Accuracy is het percentage van de documenten in de validatieset dat correct werd geclassificeerd.

$$Accuracy = \frac{(\sum goed\text{–}positief + \sum goed\text{–}negatief)}{N}$$

##### *Error*

De error is het percentage van de documenten in de validatieset dat foutief werd geclassificeerd.

$$Error = \frac{(\sum fout\text{–}positief + \sum fout\text{–}negatief)}{N} = 1 - Accuracy$$

$F_{1.0}$

$$F_{1.0} = \frac{2 * \sum goed\text{–}positief}{2 * \sum goed\text{–}positief + \sum fout\text{–}positief + \sum fout\text{–}negatief} = \frac{2 * Recall * Precision}{(Recall + precision)}$$

Bij de interpretatie van *Accuracy* en *Error* dient rekening gehouden te worden met het aandeel van positieve (categorie A) en negatieve (categorie B) observaties in de validatieset. Stel dat we een validatieset gebruiken waarbij het aantal observaties voor categorie A 1% is en het aantal observaties, en categorie B 99% van de observaties. Wanneer we een classifier zouden hebben die per definitie elk nieuw document als categorie B classificeert dan zou de classifier in die geval een *Accuracy* hebben van 99%. Om deze reden worden precision, recall en F1 vaker gebruikt bij de evaluatie van tekstclassifiers.

Dezelfde kerncijfers die worden gebruikt voor de evaluatie van classifiers worden ook gebruikt in andere NLP toepassingen zoals *Information Retreival*, groeperen van documenten, *entity extraction* en Vraag & Antwoord systemen (Konchady, 2006).

#### 4.1.5.2 Evaluatiecriteria voor een multinomial classifier

De eenvoudigste methode voor de evaluatie van een multinomial classifier is om het gemiddelde te nemen van de scores voor elke binaire taak. De berekende gemiddelde worden ***macro-averaged*** *recall, precision, F1,* etc. genoemd.

Een tweede methode is om eerste de som te nemen van alle goed-positieven, fout-positieven, goed-negatieven en fout-negatieven en vervolgens alle scores te berekenen aan de hand van de formules voor binaire evaluatie. De berekende scores worden ***micro-averaged*** scores genoemd (Lewis, Evaluating text categorization. In Proceedings of Speech and Natural Language Workshop, 1991).

#### 4.1.5.3 Testen van hypothesen

Wanneer we aan de hand van de evaluatiecriteria verschillende classificatiemethoden met elkaar vergelijken, willen we nagaan of een beter resultaat wel degelijk statistisch significant is. Over het algemeen is de nulhypothese $H_0$, de hypothese waarvan we geloven dat ze ongeldig is. De alternatieve hypothese $H_1$ is die waarvan we geloven dat ze geldig is. De hypothesetest bevestigt of verwerpt de alternatieve hypothese met een bepaalde significantie. Hypothesetesten kunnen een of tweezijdig uitgevoerd worden.

Omdat het niet mogelijk is om een beslissing te nemen met 100% zekerheid is er steeds een foutmarge (*error*). Er zijn hier twee typen van error. Een eerste is dat de alternatieve hypothese verkeerdelijk is bevestigd wanneer deze verworpen zou moeten worden (*type I error*). Het tweede type error is het verkeerdelijk bevestigen van de nulhypothese wanneer deze verworpen zou moeten worden (*type II error*) (Konchady, 2006).

| | $H_0$ *Waar* | $H_1$ *Waar* |
|---|---|---|
| *Bevestig* $H_0$ | Correct | Type II error |
| *Bevestig* $H_1$ | Type I error | Correct |

**Tabel 1 Contingecy tabel[8] voor hypothesen**

Wanneer we bijvoorbeeld een eerste classifier hebben die normaal verdeeld is met een gemiddelde nauwkeurigheid (*accuracy*) van 85% en een standaard deviatie van 0,06 en een tweede classifier waarvan we geloven dat deze een beter resultaat heeft met een nauwkeurigheid van 90%. De vraag die we ons nu stellen is of deze tweede classifier wel degelijk beter is dan de eerste. De nulhypothese hiervoor is dat het tweede model niet beter is. Wanneer we de nulhypothese kunnen verwerpen met een significatieniveau van 95% dan aanvaarden we de alternatieve hypothese. Dit wil zeggen dat het tweede model wel degelijk een beter classificatieresultaat behaalt. In dit voorbeeld gaat het om een eenzijdige test omdat we enkel interesse hebben in de waarden groter dan het gemiddelde classificatieresultaat.

Hiervoor wordt de z-waarde berekend voor een standaard normaal verdeelde distributie:

$$z = \frac{x - \mu}{\sigma} = \frac{0{,}90 - 0{,}85}{0{,}06} = 0{,}83$$

De kans voor een z-waarde van 0,83 is 0,7967 (zie bijlage). Dit is minder dan 0,95 en dus kan de nulhypothese momenteel niet verworpen worden. Op basis van de hypothesetest kan in dit voorbeeld niet met een redelijke zekerheid besloten worden dat het tweede model wel degelijk een verbetering is ten opzichte van het tweede.

#### *4.1.5.3.1 Chi-kwadraat*

Op basis van de contingecy tabel is het ook mogelijk om de Chi-kwadraat waarde te berekenen. De chi-kwadraattoets gaat na of twee of meer verdelingen van elkaar verschillen.

De Chi-kwadraat waarde kan berekend worden aan de hand van volgende formule:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Waarbij $O$ de geobserveerde waarde is en $E$ de verwachte waarde.

Het aantal vrijheidsgraden is gelijk aan het aantal categorieën minus het aantal beperkingen (*contraints*). Bij een 2 x 2 contingency tabel zoals hierboven is het aantal vrijheidsgraden 1 (2 categorieën – 1 beperking).

De toepassing van Chi-kwadraat zal verder ook nog aan bod komen bij het selecteren van producteigenschappen in sectie 4 bij feature extraction.

[8] Een contingecy tabel wordt ook een kruistabel genoemd.

# 5 Feature Extraction

Het selecteren van features is al beperkt aan bod gekomen bij het bespreken van de classificatiemethoden voor tekst. Elke classificatiemethode krijgt als input een lijst van features, tokens, woorden, ... Het classificatieresultaat is sterk afhankelijk van de features waarop de training van een classifier is gebeurd en de lijst van features gekoppeld aan een nieuw document voor classificatie. In deze sectie wordt onderzocht welke technieken er bestaan voor het selecteren van de meest relevante features. De technieken die hieronder besproken worden zijn breed inzetbaar binnen het domein van Text Mining maar de focus zal hier liggen op het selecteren van features die een producteigenschap beschrijven. Feature extraction technieken worden onder andere ook gebruikt bij classificatiemethoden om de dimensionaliteit van een lijst van features te reduceren of een gewicht toe te kennen aan de features in overeenstemming met hun belang.

Het correct kunnen identificeren van producteigenschappen maakt het mogelijk om in een later stadium te bepalen hoe positief of negatief consumenten een bepaalde producteigenschap evalueren en hoe de verhouding hiervan is tussen verschillende producten.

Het proces voor feature extraction toegepast op productbeoordelingen kan aan de hand van volgende illustratie beschreven worden:


## 5.1 Feature Extraction

Technieken om productfeatures te vinden kunnen net als Data- en Text Mining technieken in twee grote groepen onderverdeeld worden: technieken op basis van *supervised learning* en technieken op basis van *unsupervised learning*. Voor technieken op basis van supervised leaning is een training dataset noodzakelijk van gekende zinnen of zinsdelen. Vervolgens wordt er een model ontwikkeld om productfeatures uit nieuwe beoordelingen te halen. Hierbij zijn verschillende technieken mogelijk zoals Hidden Markov Models, Maximum Entropy en Naïve Bayes. (WWH)

Het grote nadeel van een supervised learning techniek is dat het samenstellen van een trainingset vrij arbeidsintensief kan zijn. Daar tegenover staan unsupervised learning technieken die op een geautomatiseerde wijze productfeatures kunnen ontdekken zonder de noodzaak aan een trainingset.

Ongeacht de gebruikte techniek dienen drie stappen ondernomen te worden om nuttige productfeatures te identificeren:

1. het vinden van productspecifieke features
2. meningen over de gevonden features identificeren
3. de richting bepalen van de gevonden meningen voor elk feature (positief of negatief)

In dit hoofdstuk zullen verder enkel technieken besproken worden voor unsupervised feature extraction.

## 5.2 Unsupervised Feature Extraction

### 5.2.1 Zelfstandige naamwoordgroepen

Uit observatie van de productbeoordelingen op BestBuy.com is gebleken dat de meeste eigenschappen die besproken worden bestaan uit een zelfstandig naamwoord of een groep zelfstandige naamwoorden die de eigenlijke producteigenschap zijn in combinatie met een bijvoeglijk naamwoord dat een beoordeling is van de producteigenschap. Deze bevinding wordt ook bevestigd door (Nakagawa & Mori, 2002).

Bijvoorbeeld: *"The battery life is good."*

De producteigenschap die we in deze zin willen herkennen is 'battery life' met het adjectief 'good' als positieve oriëntatie.

Wanneer producteigenschappen zoals hierboven expliciet worden beschreven is het mogelijk om een methode uit te werken om deze op een geautomatiseerde manier te herkennen. Maar vaak worden producteigenschappen ook impliciet beschreven waardoor deze veel moeilijker te herkennen zijn.

Bijvoorbeeld: "*The phone needs to be charged every day."*

Net als in het vorige voorbeeld wordt de batterijduur beschreven maar is dit niet expliciet vermeld. Het is ook moeilijker om te herkennen dat deze zin negatief georiënteerd is. De methoden die verder besproken zullen worden zijn niet in staat om dergelijke impliciete verbanden te ontdekken.

### 5.2.2 Part of Speech Tagging (POS)

Part of Speech Tagging (POS) is het proces van het toekennen van een woordklasse aan elk woord in een tekst. Een woord kan bijvoorbeeld een zelfstandig naamwoord, een bijvoeglijk naamwoord, een bijwoord of een andere woordklasse zijn. De grootste moeilijkheid bij het toekennen van een bepaalde woordklasse is dat een woord afhankelijk van zijn context een verschillende betekenis en vaak ook woordklasse heeft (Konchady, 2006). Zo zijn er veel woorden die zowel als een zelfstandig als een bijvoeglijk naamwoord voorkomen. Er zijn verschillende methoden voor het toekennen van een POS tag aan een woord:

#### *Unigram tagger*

Deze methode is de meest eenvoudige methode voor het toekennen van een woordklasse. Elk woord wordt afzonderlijk bekeken en krijgt de woordklasse toegekend die het meest waarschijnlijk is. Een unigram tagger dient hiervoor gebruik te maken van een voorgedefinieerde corpus waarin de woordklasse opgezocht kan worden (Hasan, UzZaman, & Khan, 2008). Het vocabularium in een corpus stijgt niet lineair met de tekst die het bevat (*Heap's Law*). Het is daarom zinvol om een tabel samen te stellen van de woorden in de corpus en de meest waarschijnlijke POS tag omdat de opslagruimte voor een dergelijke tabel beperkt is (Konchady, 2006).

#### *Hidden Markov Models (HMM)*

POS taggers op basis van HMM zien woorden niet langer als onafhankelijk van elkaar maar zijn in staat om een voorspelling te maken van de meest afhankelijke POS tag op basis van de voorgaande woorden (Hasan, UzZaman, & Khan, 2008). Een POS tagger beschikt als het ware over een 'geheugen'.

#### *Rule Based tagger*

Rule Based taggers maken gebruik van een set van regels voor het bepalen van een POS tag. Een methode voor een Rule Based tagger werd beschreven door (Tapanainen & Voutilainen, 1994) en is gebaseerd op het grammaticaal raamwerk van (Karlsson, 1990). Het basisidee is om voor een nieuw woord dat van een tag voorzien moet worden de mogelijke woordklassen op te zoeken. Vervolgens wordt het aantal mogelijkheden aan de hand van een aantal beslissingsregels gradueel beperkt.

### 5.2.3 Selecteren van kandidaat productfeatures

In deze fase worden de reviews omgezet naar een lijst met tokens en worden alle tokens weerhouden die een zelfstandig naamwoord of een groep van zelfstandige naamwoorden zijn. Ter illustratie van de uitgevoerde stappen zal gebruik worden gemaakt van een subselectie van productbeoordelingen voor de productcategorie mobiele telefoons. Deze productcategorie bevat zowel standaard mobiele telefoons als smartphones.

Kandidaat productfeatures tot drie elementen (trigrams) zullen worden geselecteerd. De frequentie van combinaties van vier of meer zelfstandige naamwoorden is zo laag dat het niet mogelijk is om na te gaan of deze statistisch relevante producteigenschappen zijn.

De procedure voor de selectie van kandidaat features is al volgt:

1. Zet de tekst om naar een lijst met tokens.
2. Selecteer elk token dat herkend wordt als een zelfstandig naamwoord op basis van de POS tagger en sla dit op als een kandidaat feature van type *NN*.
3. Selecteer elke combinatie van twee opeenvolgende tokens die elk een zelfstandig naamwoord zijn en sla deze combinatie op als een kandidaat feature van type *NN NN*.
4. Selecteer elke combinatie van drie opeenvolgende tokens die elk een zelfstandig naamwoord zijn en sla deze combinatie op als een kandidaat feature van type *NN NN NN*.
5. Hou voor elk van de kandidaat producteigenschappen bij hoe vaak deze in totaal vermeld worden in alle geselecteerde productbeoordelingen (frequentie) en het aantal beoordelingen waarin de kandidaat producteigenschap vermeld wordt (document frequentie).
6. Verwijder kandidaat features die eigen zijn aan merken of de productcategorie. In het uitgewerkte voorbeeld werden features verwijderd die een van volgende elementen bevatten: *phone, mobile, cell, product, best buy, blackberry, ipod*.

### 5.2.4 Ordenen van Productfeatures

Er zijn verschillende mogelijkheden om de lijst van kandidaat producteigenschappen te ordenen. We starten de discussie met enkele eenvoudige frequentiegebaseerde mogelijkheden. Vervolgens wordt een statistische rankingmethode op basis van Chi-kwadraat bekeken. Hierna wordt een van de meer complexe algoritmen voor het ordenen van features besproken, met name Non-Negative Matrix Factorization. Afsluitend wordt kort ingegaan op *Red Opal,* een methode die werd beschreven in een onderzoek van (Scaffidi, Bierhoff, Chang, Felker, Ng, & Jin, 2007).

#### 5.2.4.1 Term Frequency

Een eerste methode is om termen eenvoudigweg te rangschikken op basis van het aantal keer dat deze zijn voorgekomen in de selectie documenten. Hoe vaker een bepaalde kandidaat feature voorkomt hoe belangrijker deze geacht wordt te zijn.

Een selectie maken op basis van de frequentie van het voorkomen van een bepaald token of tokencombinatie heeft zin omdat heel wat productbeoordelingen ook elementen bevatten die niet direct aan een producteigenschap verbonden zijn. Verschillende schrijvers vertellen hierbij een verschillend verhaal. Maar wanneer verschillende schrijvers hun mening geven over een producteigenschap zullen zij vaak dezelfde termen gebruiken om deze eigenschap te benoemen. Tokens of tokencombinaties die vaak voorkomen hebben zodus een grotere kans om een producteigenschap te zijn (Hu & Liu, 2004).

#### 5.2.4.2 Inverse Document Frequency (IDF)

Inverse Document Frequency (IDF) maakt gebruik van het aantal keer dat een bepaalde term in een verzameling van documenten voorkomt op documentniveau. Wanneer een term dus meerdere keren voorkomt in een document wordt het maar 1 keer in overweging genomen. De Logica achter IDF is dat woorden die in een bepaalde set van documenten zeer vaak voorkomen een lager gewicht moeten krijgen.

Bij het gebruik van IDF in het ordenen van productfeatures wordt deze logica omgekeerd. Het zijn de eigenschappen die het meeste voorkomen die de het grootste gewicht dienen te krijgen zonder dat er irrelevante termen worden opgenomen.

#### 5.2.4.3 Chi-kwadraat

Naast een berekening van het belang van een bepaalde kandidaat feature op basis van frequentie gebaseerde methoden is het ook mogelijk om het belang te berekenen aan de hand van een statistische methode zoals Chi-kwadraat. Chi-kwadraat werd eerder al even kort besproken bij hypothesetesten. De Chi-kwadraat waarde wordt gegeven door volgende formule:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Waarbij $O$ de geobserveerde waarde is en $E$ de verwachte waarde.

De Chi-Square geeft de afhankelijkheid van een term $t$ en een categorie $c$ weer. Hoe hoger de Chi-kwadraat waarde, hoe hoger de associatie tussen de term $t$ en categorie $c$. Voor het gebruik in feature extraction wordt de waarde van de Chi-kwadraat enkel gebruikt om een volgorde op te stellen tussen de kandidaat features, niet om de statistische afhankelijkheid tussen de term $t$ en categorie $c$ te evalueren.

De Chi-kwadraat waarde kan berekend worden aan de hand van de *contingency tabel* van term $t$ en categorie $c$.

| | *c* | *Niet c* | *Totaal* |
|---|---|---|---|
| *t* | A | B | A+B |
| *Niet t* | C | D | C+D |
| *Totaal* | A+C | B+D | N |

De verwachte waarde voor elke cel kan gegeven worden als:

$$E = \frac{rijtotaal * kolomtotaal}{totaal\ aantal\ documenten\ in\ corpus}$$

Wanneer we dit toepassen op de contingency tabel krijgen we volgende verwachte waarden $E$:

| | *c* | *Niet c* |
|---|---|---|
| *t* | $\frac{(A+B)(A+C)}{N}$ | $\frac{(A+B)(B+D)}{N}$ |
| *Niet t* | $\frac{(A+C)(C+D)}{N}$ | $\frac{(B+D)(C+D)}{N}$ |

Vervolgens berekeken we voor elke cel de geobserveerde waarde min de verwachte waarde *(O-E)*:

| | *c* | *Niet c* |
|---|---|---|
| *t* | $\frac{(AD-BC)}{N}$ | $\frac{(BC-AD)}{N}$ |
| *Niet t* | $\frac{(BC-AD)}{N}$ | $\frac{(AD-BC)}{N}$ |

De laatste stap in het berekenen van de Chi-kwadraat waarde is het kwadrateren van elke cel en de sommatie maken:

$$\frac{(AD-BC)^2}{N(A+B)(A+C)}+\frac{(BC-AD)^2}{N(A+B)(B+D)}+\frac{(BC-AD)^2}{N(A+C)(C+D)}+\frac{(AD-BC)^2}{N(B+D)(C+D)}$$

$$=\frac{N(AD-BC)^2}{(A+B)(A+C)(B+D)(C+D)}$$

De parameters van de formule kunnen als volgt geïnterpreteerd worden:

N = Totaal aantal documenten in de corpus
A = Aantal documenten in categorie *c* die term *t* bevatten
B = Aantal documenten die term *t* bevatten in andere categorieën
C = Aantal documenten in categorie *c* die term *t* niet bevatten
D = Aantal documenten in andere categorieën die term t niet bevatten

#### 5.2.4.4 Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization is een van de meest complexe algoritmen voor feature extraction. NMF bewijst vooral zijn sterkte wanneer er features zijn die een dubbelzinnig karakter hebben of die moeilijk te voorspellen zijn. Door features te combineren is NMF in staat om betekenisvolle patronen of thema's te herkennen (Oracle®).

Wanneer in een document eenzelfde woord voorkomt op verschillende plaatsen kan dit een verschillende betekenis hebben. Bijvoorbeeld het woord "hike" kan betrekking hebben op "outdoor sports" in de betekenis van rondtrekken of op "intrest rates" in de betekenis van het verhogen van een intrest ratio.

*"hike" + "mountain" -> "outdoor sports"*
*"hike" + "interest" -> "interest rates"*

Door verschillende features met elkaar te combineren creëert NMF context waarmee het correcte thema voorspeld kan worden.

#### 5.2.4.4.1 Rekenen met matrices

Non-Negative Matrix Factorization maakt veelvuldig gebruik van het product van matrices. Daarom wordt nog even kort herhaald hoe dit juist in zijn werk gaat.

### Product

$$\underset{\mathbf{A}}{\begin{bmatrix}1 & 2\\ 4 & 3\end{bmatrix}}\underset{\mathbf{B}}{\begin{bmatrix}1 & 2 & 3\\ 3 & -4 & 7\end{bmatrix}} = \begin{bmatrix}1*1+2*3 & 1*2+2*(-4) & 1*3+2*7\\ 4*1+3*3 & 4*2+3*(-4) & 4*3+3*7\end{bmatrix} = \underset{\mathbf{C}}{\begin{bmatrix}7 & -6 & 17\\ 13 & -4 & 33\end{bmatrix}}$$

Om twee matrices met elkaar te vermenigvuldigen moet de eerste matrix (A) hetzelfde aantal kolommen hebben als de tweede matrix (B) rijen heeft. Het berekende product is ook een matrix (C) met evenveel rijen als de eerste matrix en evenveel kolommen als de tweede matrix.

De waarde van elke cel in matrix C wordt berekend door het product te nemen van de waarden in dezelfde rij in matrix A en de waarden van dezelfde kolom in matrix B. Vervolgens wordt de som gemaakt van deze producten.

Bijvoorbeeld voor de berekening van cel (1,1) in matrix C nemen we de waarden in rij 1 van matrix A (1,2) en de waarden in kolom 1 van matrix B (1,3). De twee eerste elementen worden met elkaar vermenigvuldig (1*1=1), vervolgens vermenigvuldigen we de elementen op plaats 2 (2*3=6) enz. Dan wordt de totale som genomen van elk product (1+6=7).

### Transponeren

Een andere veelgebruikte matrix operatie bij NMF is het transponeren van een matrix. Hierbij worden de kolommen rijen en de rijen kolommen.

$$\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}^T = \begin{bmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{bmatrix}$$

#### 5.2.4.4.2 Het gebruik van matrices in NMF

### De documentmatrix

De documentmatrix heeft een rij voor elk document en een kolom voor elk woord. De waarde van de cellen is het aantal keer dat een bepaald woord in een bepaald document voorkomt.

$$\begin{array}{c} \\ document\ 1 \\ document\ 2 \\ \vdots \\ document\ k \end{array}\begin{array}{c} \begin{array}{cccc} woord\ 1 & woord\ 2 & \dots & woord\ n \end{array} \\ \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & & \\ \vdots & & \ddots & \\ v_{k1} & & & v_{kn} \end{bmatrix} \end{array}$$

Omdat de grootte van de documentmatrix snel kan toenemen bij een grote dataset wordt vaak een subselectie gemaakt van het aantal woorden. Zo kunnen woorden die in een of slechts enkele documenten voorkomen beter weggelaten worden. Deze hebben een zeer kleine kans om mogelijke features te zijn. Verder kunnen ook woorden weggelaten worden die in bijna alle woorden terug te vinden zijn.

### *De features matrix*

De features matrix heeft een rij voor elk feature en een kolom voor elk woord. De waarden van de cellen vertegenwoordigen het belang dat een bepaald woord heeft ten opzicht van een feature.

$$\begin{array}{c} \\ feature\ 1 \\ feature\ 2 \\ \vdots \\ feature\ k \end{array} \begin{array}{c} \begin{array}{cccc} woord\ 1 & woord\ 2 & \dots & woord\ n \end{array} \\ \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & & \\ \vdots & \ddots & & \\ v_{k1} & & & v_{kn} \end{bmatrix} \end{array}$$

### *De gewicht matrix*

De gewicht matrix vormt de verbinding tussen de features en de documentmatrix. Elke rij is een document en elke kolom is een feature. De waarde van de cellen geeft weer in welke mate een feature verbonden is aan een bepaald document.

$$\begin{array}{c} \\ document\ 1 \\ document\ 2 \\ \vdots \\ document\ n \end{array} \begin{array}{c} \begin{array}{cccc} feature\ 1 & feature\ 2 & \dots & feature\ k \end{array} \\ \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1k} \\ v_{21} & v_{22} & & \\ \vdots & \ddots & & \\ v_{n1} & & & v_{nk} \end{bmatrix} \end{array}$$

### *Relatie tussen de matrices*

$$[gewicht\ matrix]\ x\ [features\ matrix] = [document\ matrix]$$

Non-Negative Matrix Factorization streeft ernaar om een grote set observaties (documenten, artikelen, productbeoordelingen,… ) te reduceren tot hun gemeenschappelijke features. De *non-negative* in NMF duidt er op dat het resultaat van deze methode een set features en gewichten is met geen negatieve waarden. Bij toepassing van deze methode op tekst is dit steeds het geval. Het is immers niet mogelijk om een negatieve waarde te hebben voor het aantal keer dat een woord voorkomt.

#### 5.2.4.5 Red Opal

Red Opal is een alternatieve methode om productfeatures uit productbeoordelingen te halen en werd beschreven in een onderzoek van (Scaffidi, Bierhoff, Chang, Felker, Ng, & Jin, 2007). Volgens de resultaten van dit onderzoek heeft de Red Opal methode een precisie van 88% in het herkennen van productfeatures. Deze methode zal vooral gebruikt worden als vergelijkingsbasis tegenover de eerder genoemde rankingmethoden.

### *Unigrams*

Voor unigrams wordt de RedOpal score als volgt berekend:

$$\ln\big(P(n_x)\big) \approx (n_x - p_x N) - n_x \ln\left(\frac{n_x}{p_x N}\right) - \frac{\ln(n_x)}{2}$$

Waarbij $N$ het aantal keer is dat een zelfstandig naamwoord voorkomt in alle beoordelingen van de gekozen productcategorie.

$$N = \sum_{x} n_x$$

$p_x$ is de kans dat een willekeurig geselecteerd zelfstandig naamwoord in het Engels gelijk is aan $x$. Deze kans wordt berekend op basis van een lijst met lemma-frequenties voor zelfstandige naamwoorden die werd samengesteld uit de British National Corpus (BNC)[9]. De British National Corpus bevat ongeveer 100 miljoen woorden en bevat een brede selectie van gesproken en geschreven bronnen uit verschillende domeinen.

$$p_x = \frac{aantal\ keer\ dat\ lemma\ x\ voorkomt\ in\ de\ BNC\ als\ zelfstandig\ naamwoord}{totaal\ aantal\ keer\ dat\ een\ zelfstandig\ naamwoord\ voorkomt\ in\ de\ BNC}$$

### *N-grams*

De Red Opal score voor een bigram wordt als volgt berekend:

$$ln\left(P(\eta_{xy})\right) \approx (\eta_{xy} - \rho_{xy}H) - \eta_{xy} \ln\left(\frac{\eta_{xy}}{\rho_{xy}H}\right) - \frac{\ln(\eta_{xy})}{2}$$

Waarbij $H$ het totale aantal bigrams bestaande uit twee zelfstandige naamwoorden is in de gekozen productcategorie.

$$H = \sum_{x,y} \eta_{xy}$$

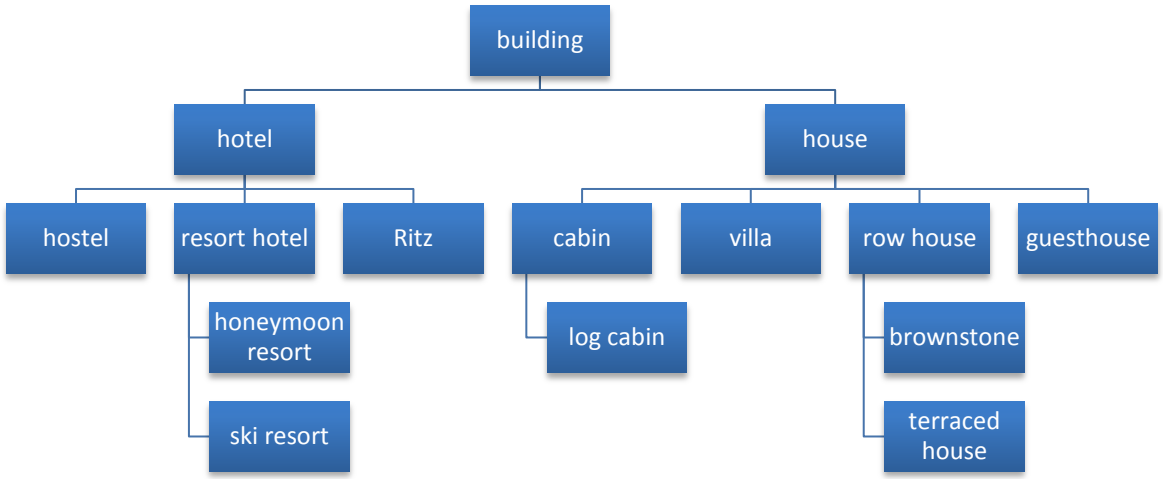
Waarbij $\rho_{xy} = p_x p_y$. Hierbij wordt gebruik gemaakt van de assumptie dat het voorkomen van lemma $x$ op positie i onafhankelijk is of dat lemma $x$ al dan niet voorkomt op een andere positie $j$. De berekening van de Red Opal Score voor n-grams van een hogere termfrequentie gebeurt met dezelfde rekenmethode.

[9] De British National Corpus is te raadplegen via http://www.natcorp.ox.ac.uk/. Voor de berekening van de lemma-frequenties werd gebruik gemaakt van de datasets gebruikt in het boek "Word Frequencies in Written and Spoken English: based on the British National Corpus" (Leech, Rayson, & Wilson, 2001). De datasets zijn te downloaden via http://ucrel.lancs.ac.uk/bncfreq/flists.html.

## 5.2.5 Groeperen van product features

### 5.2.5.1 Groeperen op basis van semantische afstand

Groeperen op basis van tekstuele overeenkomst is duidelijk niet de beste keuze. Daarom wordt nu getracht om context te geven aan de kandidaat producteigenschappen zodat deze semantisch gelinkt kunnen worden. Hiervoor worden de semantische links in WordNet gebruikt. De belangrijkste semantische linken in WordNet zijn hyperoniemen[10] en hyponiemen[11]. Daarnaast bevat WordNet ook nog een lijst van antoniemen, holoniemen en meroniemen. Voor deze toepassing worden enkel de hyperoniem en hyponiem linken gebruikt.




### 5.2.5.2 Afstand tussen twee termen

De eenvoudigste methode om de afstand tussen twee lemma's te vinden is om voor elk van deze lemma's een lijst met hyperoniemen op te stellen. Met name, we gaan 1 niveau omhoog in de semantische orde van WordNet. Dit wordt net zolang gedaan tot er een gezamenlijke link is gevonden. Zo is in het voorbeeld hierboven de afstand tussen 'hotel' en 'house' 2. Ze delen beide de link 'building'. Door van beide lemma's tegelijkertijd te vertrekken en naar boven toe te werken, wordt zo efficiënt mogelijk een overeenkomst gevonden. Bovendien wordt door deze zoekmethode gegarandeerd dat het kortste pad tussen twee lemma's gevonden wordt. Het is ook niet noodzakelijk om eenzelfde aantal niveaus omhoog te gaan. Zo is de afstand tussen *'brownstone'* en *'hotel'* 4. Ook hier is de gemeenschappelijke link *'hotel'*. Alleen is *'hotel'* een niveau lager en *'brownstone'* drie.

Gebruik maken van het kortste pad tussen twee lemma's als meeteenheid van semantische afstand is een goede methode wanneer de dichtheid van termen (hyponiemen/hyperoniemen) constant is doorheen het semantische netwerk (Richardson, Smeaton, & Murphy, 1994). Over het algemeen stijgt het aantal connecties naarmate verder wordt afgedaald in het semantische netwerk. Met andere woorden de bredere termen bovenin het netwerk vertegenwoordigen een kleiner volume aan connecties dan de meer specifieke termen lager in het netwerk. Bijgevolg vertegenwoordigt een

[10] Een *hyperoniem* is in de woordsemantiek een woord dat de betekenis van een ander woord in dezelfde of een andere taal volledig omvat, maar geen synoniem is van dat andere woord. Het hyperoniem heeft hierdoor altijd een ruimere betekenis dan het onderliggende woord, het hyponiem. (Wikipedia)

[11] Een *hyponiem* is een woord waarvan de betekenis volledig wordt gedekt door een ander woord met een doorgaans ruimere betekenis. (Wikipedia)

afstand van 1 helemaal bovenaan in het netwerk een bredere link dan een afstand van 1 lager in het netwerk tussen specifiekere termen.

Bijvoorbeeld de afstand tussen *'plant'* en *'animal'* in WordNet is 2. Ze hebben een gezamenlijk hyperoniem *'organism'*. De afstand tussen *'zebra'* en *'horse'* is ook twee. Deze twee termen delen het hyperoniem *'equine'* (paardachtigen). Intuïtief wordt de link tussen 'zebra' en 'horse' veel sterker gezien dan de link tussen *'plant'* en *'animal'*. Enkel en alleen de afstand gebruiken tussen twee lemma's is dus niet voldoende. Wanneer we het niveau van een koppel termen waarvan we de semantische afstand willen kennen in rekening nemen, kunnen we wel een betrouwbare afstandsmaat samenstellen. (Lewis, Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring Semantic Similarity, 2001)

#### *5.2.5.2.1 Niveau van een lemma*

Het niveau van een bepaald lemma is eenvoudig te achterhalen. Hiervoor wordt het pad gevolgd van hyperoniemen totdat er voor de laatste gevonden term geen hyperoniemen meer beschikbaar zijn. Het aantal connecties tussen de hoogste term en het originele lemma is het niveau (diepte) van het lemma.

### 5.2.5.3 Groeperen op basis van overeenkomst en oriëntatie

Deze methode combineert de mate van overeenkomst en de semantische afstand met de oriëntatie van kandidaat producteigenschappen. Er wordt vertrokken vanuit de assumptie dat twee kandidaat producteigenschappen die voorkomen in een set van beoordelingen voor een bepaald product en die eenzelfde oriëntatie hebben (positief of negatief), een hogere kans hebben om eenzelfde producteigenschap te beschrijven.

Bijkomend wordt er van een tweede assumptie uitgegaan dat voor alle producteigenschappen die vermeld worden in een bepaalde beoordeling de score die gekoppeld is aan deze beoordeling van toepassing is. Als de algemene score die gekoppeld is aan een bepaalde productbeoordeling negatief is dat worden alle productbeoordelingen als negatief beschouwd.

# 6 Experimenten

## 6.1 Toepassing Naïve Bayes

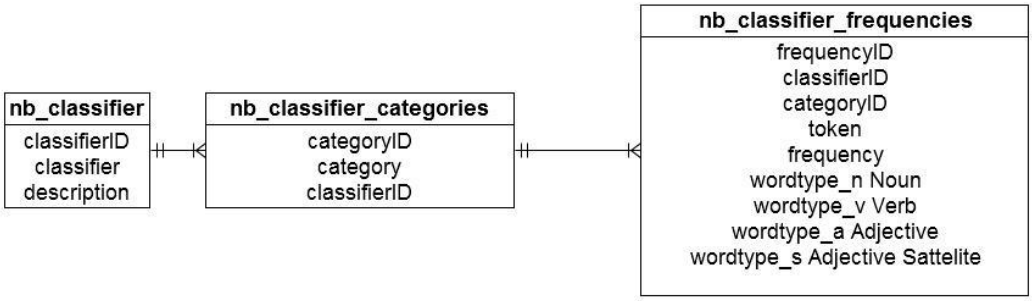
Er is gekozen voor naïve bayes als basis voor de classificatie van documenten omwille van zijn eenvoudige opzet. Hierdoor is een classifier eenvoudig te programmeren en kunnen classificaties van documenten zonder veel rekenkracht worden uitgevoerd. Tevens is empirisch bewezen dat naïve bayes ondanks zijn eenvoudige opzet zeer goede classificatieresultaten kan halen en vergelijkbare resultaten kan halen als meer gesofisticeerde classifiers zoals Neurale Netwerken en Support Vector Machines.

Andere classificatiemethoden werden tijdens het samenstellen van deze materproef onderzocht maar werden technisch niet haalbaar bevonden om met een basiskennis programmeren zelf volledig te ontwikkelen. Er zijn wel enkele pakketten beschikbaar die een kant en klare oplossing bieden om met deze andere typen van classifiers aan de slag te gaan in een PHP omgeving maar de eigenlijke werking van de gebruikte technieken blijft verborgen voor de gebruiker *('closed box').*

### 6.1.1 Naïve Bayes Classifier in PHP

Het trainen van de Naïve Bayes Classifier en de classificatie van documenten gebeurt op basis van een aantal PHP scripts en een onderliggende MySQL database. De structuur van de programma's en de database is zo opgebouwd dat deze bruikbaar zijn voor verschillende trainingsets en toepassingen.

#### 6.1.1.1 Structuur van de database




databse structuur naïve bayes classifier

**nb_classifier_classifier**

Deze tabel houdt de ID en naam van de verschillende classifiers bij.

**nb_classifier_categories**

Deze tabel bevat voor elke classifier de mogelijke categorieën.

**nb_classifier_frequencies**

De tabel *nb_classifier_frequencies* is de eigenlijke tabel die alle informatie bevat voor de Näive Bayes classifier. De tabel onthoudt voor elk token het aantal keer dat het voorkomt in een bepaalde

categorie (frequentie). Voorts wordt bij het trainen van de classifier nagegaan of het token voorkomt in WordNet. Wanneer dit het geval is wordt voor dit token opgeslagen welke woordtypen het kan aannemen.

De tabel *nb_classifier_frequencies* werd niet volledig genormaliseerd. Zo wordt er voor elke regel in de tabel ook de classifierID opgeslagen. Dit is in principe niet noodzakelijk omdat de classifier ID steeds gevonden kan worden door de tabel *nb_classifier_frequencies* en *nb_classifier_categories* met elkaar te combineren. Er werd gekozen om deze waarde toch ook op te slaan in de tabel omdat er vaak berekend moet worden hoe vaak een token voorkomt in een bepaalde classifier. Bij volledige normalisatie zou hier telkens een '*join*' van twee tabellen moeten uitgevoerd worden. Door de classifierID ook in deze tabel op te slagen kan de snelheid van de classificatie verhoogd worden. Bijkomend is de extra opslag voor de classifierID's zeer beperkt.

#### 6.1.1.2 Beschrijving van de php functies

Voor de Naïve Bayes classifier werd een PHP- klasse geschreven. Hieronder volgt een beschrijving van de functies van de klasse voor Naïve Bayes. Voor elke functie wordt opgegeven welke inputdata de functie nodig heeft, de handelingen die de functie uitvoert met de data, welke andere functies aangeroepen worden tijdens het manipuleren van de data en het eindresultaat van de functie. De volledige broncode is te raadplegen in de bijlagen.

##### *6.1.1.2.1 Algemene tekstfuncties*

**splitText(***tekst***)**
Splitst een stuk tekst bij elk niet alfanumeriek karakter en geeft een array terug met alle stukken.
*resultaat: array*

**cleanString(***string***)**
Zet letters met accenten om naar de basisletter.
*resultaat: string*

**stringToToken(***string***)**
Kuist een string op zodat deze bruikbaar is als token. De functie verwijdert onnodige witruimte aan het begin en einde van een string, zet alle letters om naar lowercase en roept de functie cleanString aan om alle accenten te verwijderen.
*maakt gebruik van: cleanString()*
*resultaat: string*

**textToTokens(***tekst***)**
Zet een stuk tekst om naar bruikbare tokens. Eerst wordt de tekst gesplitst, vervolgens worden alle tekstdelen 'opgekuist' en opgeslagen in aan array met 'ruwe' tokens. Bijkomend wordt nagegaan voor elk ruw token of het token voorkomt in WordNet. Indien dit niet het geval is wordt het Porter Stemming Algoritme uitgevoerd op het token. Als laatste stap worden alle tokens van drie of minder karakters geweerd.
*maakt gebruik van: splitText(), stringToToken(), stem()*
*resultaat: array*

**stem(***string***)**
De functie stem past het Porter Stemming Algoritme toe op een string. De code werd overgenomen van de stemmer class van Jon Abernathy.
*resultaat: string*

#### *6.1.1.2.2 Functies voor de training van de classifier*

**trainText(***classifier, categorie, tekst***)**
De functie trainText() zet een tekst om naar tokens en gaat vervolgens na voor elk van deze tokens of deze reeds voorkomen in de lijst met frequenties. Wanneer het token reeds voorkomt wordt de bestaande frequentie verhoogd met de frequentie van het token in het document dat wordt toegevoegd aan de training data. Wanneer het token nog niet bestaat in de lijst met frequenties dan wordt een nieuwe regel aangemaakt met als basisfrequentie het aantal keer dat het token voorkwam in het document dat toegevoegd wordt aan de training data. Bijkomend wordt nagegaan of het token voorkomt in de WordNet database en desgevallend wat de woordtypen zijn.
*maakt gebruik van: textToTokens()*
*resultaat: true*

#### *6.1.1.2.3 Functies voor de classificatie van documenten*

**tokenInClassifier(***classifier,token***)**
Geeft de frequentie dat token voorkomt in de trainingdata van een bepaalde classifier. Alle categorieën van deze classifier worden aangesproken.
*resultaat: integer*

**tokenInCategory(***categorie,token***)**
Geeft de frequentie dat een token voorkomt in een bepaalde categorie.
*resultaat: integer*

**countCategories(***classifier***)**
Geeft het totaal aantal categorieën voor een bepaalde classifier als een integer.
*resultaat: integer*

**tokenInOtherCategories(***token, classifier,categorie***)**
Geeft de frequentie dat een token voorkomt in categorieën verschillend van een gegeven categorie voor een classifier.
*resultaat: integer*

**countOccurencesInOtherCategories(***classifier,categorie***)**
Geeft de totale frequentie van alle tokens voor alle categorieën verschillend van de gegeven categorie voor een classifier.
*resultaat: integer*

**categories(***classifier***)**
Geeft als resultaat een array met de benaming voor elke categorie in een classifier en de corresponderende categoryID.
*resultaat: array*

**MNB()**
Classificeert een array met tokens en frequenties volgens Multinomial Naïve Bayes.

*maakt gebruik van: countCategories(), categories(), tokenInCategory(), tokenInClassifier()*

**classifyMNB(***tekst, classifier***)**
Classificeert een document volgens Multinomial Naïve Bayes.
*maakt gebruik van: textToTokens(), MNB()*

**CNB()**
Classificeert een array met tokens en frequenties volgens Complement Naïve Bayes.
*maakt gebruik van: countCategories(), categories(),tokenInOtherCategories(), countOccurencesInOtherCategories ()*

**classifyCNB(***tekst, classifier***)**
Classificeert een document volgens Complement Naïve Bayes.
*maakt gebruik van: textToTokens(), CNB()*

#### 6.1.1.3 Trainen van de Classifier

Het trainen van de classifier gebeurt eenvoudigweg door een nieuwe instantie van de naïve bayes class aan te roepen en vervolgens de functie *trainText* toe te passen op het document.

```
$classify = new NaiveBayes();
$result = $classify->trainText(classifier, categorie, tekst);
```

De stap wordt uitgevoerd voor alle documenten in een trainingset. Het is zonder probleem mogelijk om op een later tijdstipt nieuwe documenten toe te voegen aan de trainingdata.

#### 6.1.1.4 Classificeren van tekst

Voor het classificeren van een document wordt net als bij het trainen van de classifier eerst een nieuwe instantie van de naïve bayes class geïnitialiseerd. Vervolgens kan door middel van de functie classifyMNB en/of classifyCNB het document geclassificeerd worden.

```
$classify = new NaiveBayes();
$resultMNB = $classify->classifyMNB(tekst, classifier);
$resultCNB = $classify->classifyCNB(tekst, classifier);
```

### 6.1.2 Voorbeelden van Classificatie

#### 6.1.2.1 Dataverzameling

*BestBuy*

Best Buy is een van de grootste online retailers van consumentenelektronica in de Verenigde Staten. De productbeoordelingen op deze website zijn in het Engels. Er is bewust voor gekozen om Engelstalige beoordelingen te gebruiken omdat het aanbod van Engelstalige productbeoordelingen vele malen groter is dan het aanbod aan Nederlandstalige.

BestBuy heeft een developerprogramma dat er op gericht is om de data op de BestBuy website te delen met ontwikkelaars. Een van de mogelijkheden van het developerprogramma is het downloaden van alle producten, de productstructuur van de website en alle productbeoordelingen als een gecomprimeerde map. Deze gecomprimeerde map bevat een aantal xml files die de data bevatten. De xml files worden via een php script ingelezen en worden vervolgens gekopieerd naar een MySQL database.

De reviews die beschikbaar zijn via het developer programma van BestBuy bevatten de *'sku'* van het product, de naam van de reviewer, een score van 1 tot 5, een titel, een tekstuele beoordeling en de datum waarop de beoordeling geplaatst werd. De data wordt aangeleverd in volgend xml formaat:

```
<reviews>
 <review>
  <id></id>
  <sku></sku>
  <reviewer>
   <name> </name>
  </reviewer>
  <rating></rating>
  <title> </title>
  <comment></comment>
  <submissionTime></submissionTime>
 </review>
 …
<reviews>
```

Veel van de reviews op BestBuy.com bevatten bijkomend ook nog een apart veld voor positieve en negatieve eigenschappen van het product en naast een algemene score van 1 tot 5 ook aparte beoordelingen op enkele producteigenschappen. Jammer genoeg wordt deze data niet beschikbaar gesteld via het developer programma en is het ook niet toegestaan om deze door middel van *'web scraping'* technieken rechtsreeks van de website te halen.

*Beschrijving van de BestBuy Data*

| Trainingdata | |
|---|---|
| # positieve productbeoordelingen | 45.757 |
| Gemiddeld aantal karakters positieve beoordelingen | 443,46 |
| # negatieve productbeoordelingen | 45.757 |
| Gemiddeld aantal karakters negatieve beoordelingen | 519,40 |
| **Testdata** | |
| # positieve productbeoordelingen | 11.500 |
| Gemiddeld aantal karakters positieve beoordelingen | 446,12 |
| # negatieve productbeoordelingen | 11.500 |
| Gemiddeld aantal karakters negatieve beoordelingen | 508,87 |

#### 6.1.2.2 Voorbeeld 1: Productbeoordelingen classificeren volgens rating

Een eerste classifier maakte gebruik van het 5 punten systeem van productbeoordelingen van BestBuy. Deze classifier heeft dus 5 mogelijke categorieën waaraan een document kan toegewezen worden, namelijk rating 1, 2, 3, 4 of 5 met 1 de slechtst mogelijke score en 5 de best mogelijke beoordeling.

Er werd een trainingset opgesteld van in totaal 50.000 productbeoordelingen met 10.000 beoordelingen voor elke categorie. Zowel de titel als de geschreven productbeoordelingen worden gebruikt voor de training van de classifier en het later classificeren van documenten. Omdat de titel van de beoordeling vaak al een goede indicatie is of een productbeoordeling positief of negatief is wordt hier een hogere waardering aan gegeven. Om dit te realiseren wordt de titel twee keer toegevoegd aan de tekst voor training of classificatie.

Vervolgens werd een validatie uitgevoerd met 5.000 ongeziene beoordelingen. Het classificatieresultaat van de classifier was eerder teleurstellend. Ongeveer 55% van de beoordelingen werd aan de correcte categorie toegewezen. Mogelijk wordt dit veroorzaakt doordat het moeilijk is om kleine nuances te leggen tussen hoe positief of hoe negatief een beoordeling is.

#### 6.1.2.3 Voorbeeld 2: Productbeoordelingen classificeren volgens positief of negatief

Aangezien de eerste classifier geen betrouwbare resultaten kon genereren werd de structuur van de classifier aangepast naar een met twee mogelijke categorieën, namelijk een productbeoordeling kan positief of negatief zijn. Productbeoordelingen met een rating van 1 of 2 worden gezien als positieve beoordelingen. Productbeoordelingen met een rating 4 of 5 als positieve. Neutrale productbeoordelingen met een rating van 3 werden niet gebruikt voor deze classifier.

De training set voor de classifier bestaat uit in totaal 91.514 productbeoordelingen waarvan 45.757 positieve en 45.757 negatieve productbeoordelingen. De validatie set bevat 23.000 productbeoordelingen waarvan 11.500 positieve en 11.500 negatieve productbeoordelingen. Er werd gekozen voor een gelijke grootte van categorieën om problemen met de MNB Classifier te vermijden. Het aantal beoordelingen in elke categorie is ook voldoende groot om de classifier te trainen. Net als in het vorige voorbeeld wordt een dubbele waardering toegekend aan de titel van de productbeoordeling.

De validatie van de classifier geeft een correct resultaat van 83% voor een classificatie met MNB en een correct classificatieresultaat van 89% voor een classificatie op basis van CNM. Vervolgens kunnen we de classifiers toepassen op de beoordelingen van een bepaald product om de verwachte gemiddelde beoordeling te berekenen. Elke productbeoordeling wordt geclassificeerd als positief of negatief. Voor een negatieve classificatie wordt een waarde 1 toegekend, voor een positieve classificatie waarde 5. Na de classificatie van alle beoordelingen wordt het gemiddelde van alle classificatieresultaten genomen.

$$Voorspelde\ waardeling = \frac{\sum_{i=1}^{n} Resultaat\ classificatie\ productbeoordeling\ i}{Aantal\ productbeoordelingen\ (n)}$$

Ter illustratie worden de resultaten gegeven voor de classificatie van het spel *Grand Theft Auto IV Platinum Hits - Xbox 360 (sku 7959033)*[12]. Voor dit product zijn er 300 beoordelingen beschikbaar. De classifier op basis van MNB kan voor 89,00% van de productbeoordelingen correct voorspellen of een deze positief of negatief van toon zijn en geeft een voorspelde gemiddelde waardering van 4,33 op een schaal van 1 tot 5. De classifier op basis van CNB zorgt in 93,62% van de gevallen voor een correcte classificatie en geeft een voorspelde gemiddelde waardering van 4,72. De werkelijke gemiddelde waardering voor dit spel is 4,64.

Een algemene trend bij het valideren van de classifiers op productniveau is dat de gemiddelde voorspelde waardering van het product in het algemeen een fractie lager is dan wat deze waarde zou moeten zijn. Dit is mogelijk te verklaren door een verschil in lengte van de productbeoordelingen voor de twee categorieën. De gemiddelde grootte van een beoordeling met een negatieve rating is 517 karakters. Daarentegen is de gemiddelde grootte van een positieve beoordeling gemiddeld slechts 445 karakters. Door dit verschil in lengte zal een classifier op basis van Naïve Bayes sneller een beoordeling als negatief classificeren.

[12] Een deel van het classificatieresultaat is opgenomen in de Bijlagen *(bijlage 9.4).*

### 6.1.3 De invloed van data preprocessing op een Naïve Bayes Classifier

In dit experiment worden gradueel enkele data preprocessingtechnieken toegepast om hun invloed te onderzoeken op een Naïve Bayes Classifier. Voor dit experiment wordt gebruik gemaakt van 20000 willekeurig geselecteerde productbeoordelingen van de BestBuy dataset waarvan 10000 positieve en 10000 negatieve beoordelingen. De invloed van de toevoeging van de preprocessingtechnieken werd geëvalueerd via graduele aanpassingen in de PHP-klasse voor Naïve Bayes. Er wordt gestart met een basisversie die enkel tekst splitst in tokens en alle tokens van minder dan 3 karakters weglaat. Vervolgens wordt het filteren van stopwoorden toegevoegd en in de laatste test wordt stemming op basis van Porter toegevoegd.

#### *Basis Classifier*

De basisversie van de classifier splitst de tekst in de documenten op in tokens bij elk niet alfanumeriek karakter. Vervolgens worden alle tokens omgezet naar de *lowercase* variant en worden tokens van minder dan 3 karakters verwijderd.


| Multinomial Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |
| voorspeld positief | | 389 | |
| voorspeld negatief | | 9611 | |
| recall groep | | | |

| Accuracy | |
|---|---|

| Complement Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |
| voorspeld positief | | | |
| voorspeld negatief | | | |
| recall groep | | | |

| Accuracy | |
|---|---|

#### *Filteren van Stopwoorden*


| Multinomial Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |

| | | | |
|---|---|---|---|
| voorspeld positief | 7656 | 389 | 95,16% |
| voorspeld negatief | 2344 | 9811 | 80,72% |
| recall groep | 76,56% | 98,11% | |

| | |
|---|---|
| Accuracy | 87,34% |

| Complement Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |
| voorspeld positief | 9177 | 1371 | |
| voorspeld negatief | 823 | 8629 | |
| recall groep | 91,77% | 86,29% | |

| | |
|---|---|
| Accuracy | 89,03% |

### *Woordstemming(Porter)*


| Multinomial Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |
| voorspeld positief | 7486 | 307 | 96,06% |
| voorspeld negatief | 2514 | 9693 | 79,41% |
| recall groep | 74,86% | 96,93% | |

| | |
|---|---|
| Accuracy | 85,90% |

| Complement Naïve Bayes | | | |
|---|---|---|---|
| | verwacht positief | verwacht negatief | precisie groep |
| voorspeld positief | 8915 | 967 | |
| voorspeld negatief | 1085 | 9033 | |
| recall groep | 89,15% | 90,33% | |

| | |
|---|---|
| Accuracy | 89,74% |

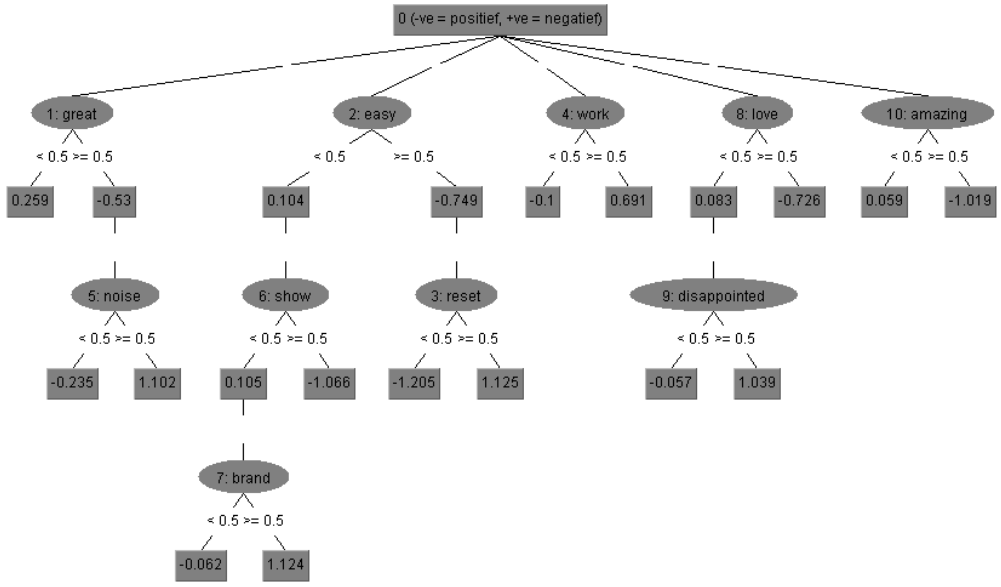
### 6.1.4 Classificatie met Beslissingsbomen

#### *Classificatie van kleine samples*

Een van de voordelen van beslissingsbomen is dat deze vaak al bij zeer kleine datasets goede resultaten kunnen behalen. Om dit te testen wordt een eerste classificatie gedaan op basis van slechts 200 willekeurig geselecteerde productbeoordelingen waarvan 100 positieve en 100 negatieve. Het model werd gevalideerd aan de hand van een 10-keer crossvalidatie.

| | verwacht positief | verwacht negatief | precisie groep |
|---|---|---|---|
| voorspeld positief | 60 | 19 | 75,95% |
| voorspeld negatief | 40 | 81 | 66,94% |
| recall groep | 60,00% | 81,00% | |

Accuracy: 70.50% +/- 10.36%

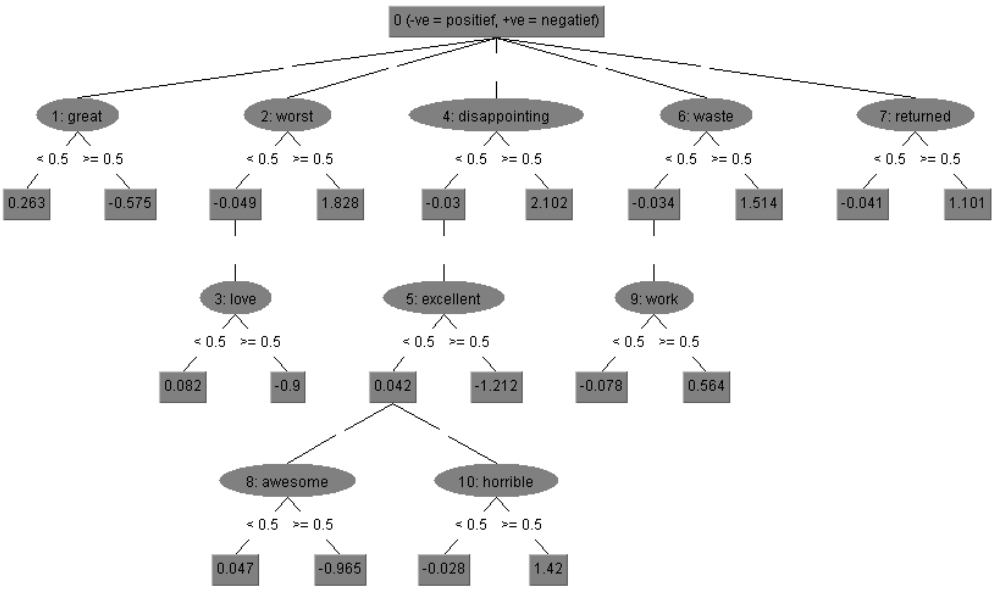



#### *Classificatie van 6000 samples*

In de volgende test wordt de beslissingsboom getest op een subsample van 3000 positieve en 3000 negatieve beoordelingen om de werking van de beslissingsboom beter te vergelijken met de andere classificatiemethoden.

| | verwacht positief | verwacht negatief | precisie groep |
|---|---|---|---|
| voorspeld positief | 393 | 63 | 86,18% |
| voorspeld negatief | 218 | 526 | 70,70% |
| recall groep | 64,32% | 89,30% | |

Accuracy: 76.58%


0 (-ve = positief, +ve = negatief)
1: great
2: worst
4: disappointing
6: waste
7: returned
< 0.5 >= 0.5
0.263
-0.575
-0.049
1.828
-0.03
2.102
-0.034
1.514
-0.041
1.101
3: love
5: excellent
9: work
0.082
-0.9
0.042
-1.212
-0.078
0.564
8: awesome
10: horrible
0.047
-0.965
-0.028
1.42

### 6.1.5 Classificatie met Neurale Netwerken in RapidMiner

| | verwacht positief | verwacht negatief | precisie groep |
|---|---|---|---|
| voorspeld positief | 684 | 293 | |
| voorspeld negatief | 202 | 621 | |
| recall groep | 77,20% | 65,80% | |

Accuracy: 72,50%

### 6.1.6 Classificatie met Support Vector Machines in RapidMiner

| | verwacht positief | verwacht negatief | precisie groep |
|---|---|---|---|
| voorspeld positief | 2847 | 877 | 69,30% |
| voorspeld negatief | 153 | 2123 | 74,27% |
| recall groep | 77,20% | 65,80% | |

Accuracy: 82,83% +- 1.17%

Zoals verwacht is Support Vector Machines een van de betere methoden voor tekstclassificatie. De precisie van classificeren ligt in het bovenstaande voorbeeld wel nog onder het resultaat van de Naïve Bayes classifier maar het is nog mogelijk om de instellingen van de Support Vector Machine verder te verfijnen naargelang de toepassing. Een groot nadeel is de tijd die nodig is voor het trainen van een model en het uitvoeren van classificaties. Voor het verwerken van de 6000 productbeoordelingen in RapidMiner was er ongeveer 6 uur rekenwerk noodzakelijk. Dit maakt deze methode minder geschikt voor snelle classificaties.

### 6.1.7 Unsupervised Feature Extraction in BestBuy productbeoordelingen

Voor dit experiment werd een selectie gemaakt van 14.349 productbeoordelingen die een beoordeling zijn van een mobiele telefoon. De productbeoordelingen werden vervolgens omgezet in tokens met de tekstfuncties van de naïve bayes PHP- klasse. Voor elk document werd naast een lijst van gewone tokens (unigrams) ook een lijst van bigrams en trigrams gegenereerd. Alle kandidaat tokens zijn zelfstandige naamwoorden of groepen van zelfstandige naamwoorden naar aanleiding van de bevinding van (Nakagawa & Mori, 2002) dat de meeste producteigenschappen bestaan uit een zelfstandig naamwoord of een groep van zelfstandige naamwoorden.

#### 6.1.7.1 Part of Speech Tagging (POS)

Om een token te identificeren als een bepaald woordtype en hier specifiek als een zelfstandig naamwoord werd een aangepaste tabel samengesteld vanuit WordNet. Omdat heel wat lemma's verschillende mogelijke woordtypen hebben worden heel wat adjectieven ook als mogelijk zelfstandig naamwoord aangeduid. Een van de mogelijkheden is om hier het woord te kiezen met het hoogste aantal definities. Volgens een studie van (Charniak, 1993) kan hier een precisie tot 90% mee gehaald worden. Enkele steekproeven wezen echter uit dat heel wat mogelijke producteigenschappen hierdoor niet herkend konden worden. Daarom werd er een aangepaste toewijzingsregel toegepast:

1. Selecteer elk lemma in de WordNet database en bereken het aantal gekoppelde definities per woordsoort.
2. Selecteer de definitie met het hoogste aantal definities en sla deze op.
3. Behoudt ook alle woordsoorten waarvan het aantal vermelde definities minimaal 75% is van de woordsoort met het hoogste aantal definities voor dat lemma.
4. Behoudt ook alle woordsoorten die minimaal 10 definities hebben in WordNet.

Op basis van deze toewijzingsregel is het mogelijk dat een lemma meerdere woordsoorten blijft behouden. Ter illustratie bekijken we de veel voorkomende producteigenschap *'blue tooth'* voor mobiele telefoons[13]. Deze eigenschap is een bigram en bevat twee lemma's die in WordNet terug te vinden zijn. Namelijk *'blue'* en *'tooth'*. Het lemma *'blue'* heeft in WordNet 8 vermelde definities als zelfstandig naamwoord *(noun)*, 7 definities als bijvoeglijk naamwoord *(adjective)* en 1 vermelding als werkwoord *(verb)*. Het lemma *'tooth'* heeft 5 vermeldingen als zelfstandig naamwoord. Door de regel voorgesteld voor (Charniak, 1993) toe te passen zou deze producteigenschap als een bigram van type *JJ NN* geclassificeerd worden. De aangepaste toewijzingsregel herkent deze bigrams als type *JJ NN* en als type *NN NN*[14].

#### 6.1.7.2 Ordenen van kandidaat eigenschappen op basis van frequentiemethoden

Nadat de drie lijsten voor kandidaat producteigenschappen werden samengesteld worden de lijsten via een aantal methoden geordend. De eerste kolom is gesorteerd op basis aantal keer dat een bepaald n-gram voorkwam in de dataset. De tweede kolom is gesorteerd op basis van het aantal

[13] Deze producteigenschap werd willekeurig geselecteerd en er werd niet nagegaan of deze producteigenschap wel degelijk significant is voor gebruikers van mobiele telefoons. Het voorbeeld is enkel ter illustratie van de toewijzingsregel voor woordsoorten aan lemma's.

[14] *JJ* betekend dat de POS tag voor een token adjectief is. *NN* betekend dat de POS tag voor een token een zelfstandig naamwoord is.

documenten waarin een bepaald n-gram voorkwam. De derde kolom werd geordend op basis van de Inverse Document Frequency (IDF)

De laatste kolom bevat een sortering op basis van Red Opal.

#### 6.1.7.3 Unigrams (NN)

| frequentie | document frequentie | IDF | Red Opal |
|---|---|---|---|
| battery | battery | battery | battery |
| screen | screen | screen | screen |
| case | love | love | call |
| love | life | life | love |
| call | time | time | camera |
| life | call | call | better |
| time | better | better | keyboard |
| better | problem | problem | text |
| text | price | price | touch |
| problem | case | case | button |
| camera | camera | camera | headset |
| quality | quality | quality | works |
| sound | text | text | sound |
| back | back | back | using |
| price | works | works | android |
| headset | sound | sound | ear |
| works | thing | thing | case |
| touch | using | using | device |
| keyboard | touch | touch | worth |
| using | need | need | quality |

#### 6.1.7.4 Bigrams (NN NN)

| frequentie | document frequentie | IDF | Red Opal |
|---|---|---|---|
| battery life | battery life | battery life | battery life |
| touch screen | touch screen | touch screen | touch screen |
| screen protector | sound quality | sound quality | screen protector |
| sound quality | screen protector | screen protector | sound quality |
| call quality | call quality | call quality | battery last |
| battery last | battery last | battery last | call quality |
| car charger | car charger | car charger | car charger |
| customer service | customer service | customer service | blue tooth |
| memory card | memory card | memory card | memory card |
| android market | android market | android market | customer service |
| blue tooth | data plan | data plan | belt clip |
| data plan | blue tooth | blue tooth | android market |
| ear piece | music player | music player | ear piece |
| music player | ear piece | ear piece | ear bud |
| ear bud | talk time | talk time | slide keyboard |
| talk time | task killer | task killer | task killer |
| belt clip | camera video | camera video | data plan |

| | | | |
|---|---|---|---|
| task killer | background noise | background noise | camera video |
| voice dial | voice dial | voice dial | background noise |
| background noise | worth price | worth price | music player |

#### 6.1.7.5 Trigrams (NN NN NN)

| **frequentie** | **document frequentie** | **IDF** | **Red Opal** |
|---|---|---|---|
| battery life better | battery life better | battery life better | battery life better |
| battery life isn | battery life isn | battery life isn | front facing camera |
| front facing camera | front facing camera | front facing camera | battery life last |
| problem battery life | problem battery life | problem battery life | problem battery life |
| battery life last | battery life last | battery life last | better battery life |
| better battery life | better battery life | better battery life | keyboard touch screen |
| complaint battery life | complaint battery life | complaint battery life | complaint battery life |
| keyboard touch screen | keyboard touch screen | keyboard touch screen | battery last longer |
| battery last longer | battery last longer | battery last longer | touch screen keyboard |
| touch screen keyboard | touch screen keyboard | touch screen keyboard | blue tooth headset |
| hours talk time | hours talk time | hours talk time | battery life battery |
| blue tooth headset | blue tooth headset | blue tooth headset | love touch screen |
| battery life battery | battery life battery | battery life battery | con battery life |
| love touch screen | love touch screen | love touch screen | touch screen works |
| case screen protector | case screen protector | case screen protector | case screen protector |
| con battery life | con battery life | con battery life | battery last hours |
| touch screen works | touch screen works | touch screen works | screen protector screen |
| call customer service | call customer service | call customer service | hours talk time |
| screen protector screen | screen protector screen | screen protector screen | longer battery life |
| longer battery life | longer battery life | longer battery life | call customer service |

#### 6.1.7.6 Evaluatie n-grams

Uit een visuele evaluatie van de kandidaat features blijkt dat unigrams heel wat kandidaten bevat die geen echte producteigenschappen zijn.

De bigrams daarentegen vertegenwoordigen wel voor een groot deel echte producteigenschappen. Bijkomend zijn de bigrams vaak specifieker in het omschrijven van een producteigenschap dan een unigram. Wanneer bijvoorbeeld het woord *'battery'* voorkomt in een productbeoordeling is de kans zeer groot dat hier de duur dat de batterijen operationeel zijn wordt geëvalueerd. Andere eigenschappen van een batterij zoals kleur, grootte, vorm,… zullen voor deze productcategorie zelden beschreven worden.

De trigrams scoren ook goed op het herkennen van productfeatures maar vertonen een grote overlapping waarbij verschillende kandidaten eenzelfde producteigenschap zijn. Vaak beschrijven ze ook een producteigenschap die reeds in de bigrams terug te vinden zijn. Bijvoorbeeld voor het bigram *'battery life'* zijn zeer veel afleiding terug te vinden in de lijst van trigrams. De frequentie waarin de trigrams voorkomen is ook zeer laag. Het meest voorkomende trigram 'battery life better' komt bijvoorbeeld maar 74 keer voor in de meer dan 14.000 reviews. Dergelijke lage frequentie maakt het zo goed al onmogelijk om later producten op een voldoende betrouwbare manier met

elkaar te gaan vergelijken. Daarom zullen voor de volgende fase enkel de bigrams weerhouden worden.

### *Evaluatie Scoringsmethoden*

Alle gebruikte scoringsmethoden suggereren ongeveer dezelfde producteigenschappen zij het met kleine verschillen in de ordening van de kandidaat productfeatures. Opvallend is dat de frequentiegebaseerde scoringsmethoden niet moeten onderdoen voor een meer geavanceerde methode als Red Opal.

#### 6.1.7.7 Selecteren van kandidaat productfeatures op basis van Chi-kwadraat

In een eerste test op basis van Chi-kwadraat wordt een poging gedaan om producteigenschappen te selecteren op basis van de verschillen tussen positieve en negatieve reviews.

| **Unigrams** | **Bigrams** |
|---|---|
| love | waste money |
| waste | waste time |
| work | tech support |
| horrible | customer service |
| return | piece junk |
| support | surround sound |
| junk | bang buck |
| price | worth penny |
| problem | time money |
| brand | error message |
| money | sound system |
| thought | buyer beware |
| customer | product work |
| service | love camera |
| week | dont waste |
| repair | thing work |
| surround | love movie |
| family | quality price |
| dissapointment | horrible product |
| mistake | return product |

Een visuele inspectie van bovenstaande tabel leert dat deze methode toepassen op deze twee groepen niet het gehoopte resultaat geeft. Het is dus niet zo dat de producteigenschappen die in positieve of negatieve reviews besproken worden verschillend zijn.

Een tweede mogelijkheid voor het toepassen van de Chi-kwadraat methode is het gebruik van beoordelingen voor twee productcategorieën. Er wordt van de assumptie uit gegaan dat de eigenschappen van een product voor elk van de twee productcategorieën verschillend zijn en dat de kandidaat features die door de Chi-kwadraat methode als eerste geselecteerd worden effectieve producteigenschappen zijn.

| **Unigrams** | **Bigrams** |
|---|---|
| headset | camera price |
| picture | point shoot |
| zoom | picture quality |
| call | sound quality |
| lens | bluetooth headset |
| sound | camera picture |
| shoot | camera quality |
| shot | price camera |
| text | quality camera |
| image | camera zoom |
| voice | camera battery |
| point | purchase camera |
| video | camera time |
| volume | image stabilization |
| photo | phone camera |
| keyboard | phone battery |
| talk | time camera |
| case | camera phone |
| shutter | case phone |
| flash | feature camera |

### 6.1.8 Filteren van kandidaat productfeatures

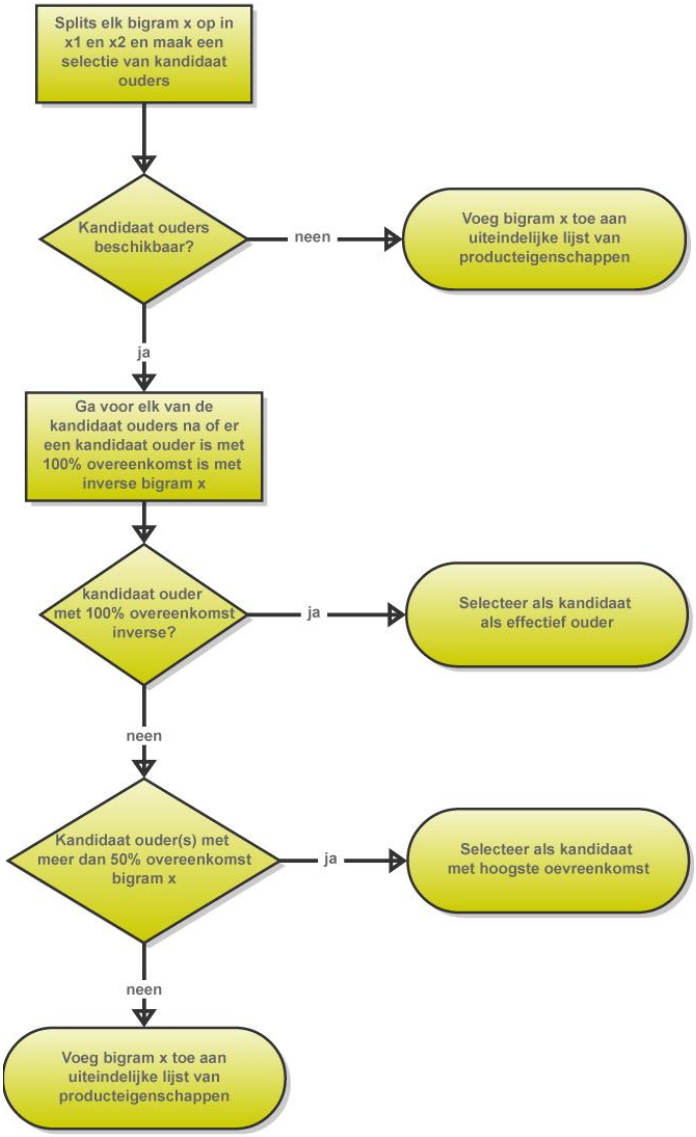
De lijst van bigrammen bevat momenteel nog enkele 'dubbele kandidaten'. Dit zijn kandidaat productfeatures die eenzelfde producteigenschap beschrijven. Op de eerste plaats vinden we bijvoorbeeld kandidaat producteigenschap *'battery life'*. Op plaats zes vinden we de kandidaat *'last battery'*. Beide kandidaten beschrijven dezelfde eigenschap voor een mobiele telefoon. Namelijk wat de batterijduur is van de telefoon. In deze sectie trachten we een methode te vinden om kandidaten die eenzelfde producteigenschap beschrijven te groeperen.

### 6.1.9 Groeperen op basis van tekstuele overeenkomst

De eerste methode die getest zal worden tracht kandidaat producteigenschappen te koppelen op basis van hun tekstuele overeenkomst. Hierbij wordt volgende methoden gevold:

1. Splits elk bigram op in zijn twee aparte delen $x_1, x_2$. Ga voor elk van deze delen na of er een kandidaat producteigenschap bestaat die minstens een van deze twee delen bevat en hoger in de ranglijst staat dan bigram $x$. Alle gevonden eigenschappen zijn *'kandidaat ouders'* voor bigram $x$. Wanneer er geen kandidaat ouders zijn wordt bigram $x$ toegevoegd aan de uiteindelijke lijst van producteigenschappen.
2. Vervolgens wordt aan de hand van de php functie *similar_text()*[15] nagegaan wat de overeenkomst is tussen bigram $x$ en elk kandidaat ouder. Wanneer er een of meerdere kandidaat ouders zijn die meer dan 50% overeenkomst hebben wordt het kandidaat ouder met de hoogste overeenkomst gekozen. Vervolgens wordt nagegaan wat de overeenkomst is van de inverse van bigram $x$ met elk van de kandidaat ouders. Wanneer de inverse een overeenkomst heeft van 100% met een ven de kandidaat ouders wordt deze kandidaat ouder geselecteerd als effectief ouder. Zoniet wordt bigram toegevoegd aan de uiteindelijke lijst van producteigenschappen.

[15] http://php.net/manual/en/function.similar-text.php




## *Voorbeeld*

- Bigram$_1$ '*battery life'*. Geen kandidaat ouders dus *'battery life'* wordt toegevoegd aan de uiteindelijke lijst van producteigenschappen.

  …
- Bigram$_6$ *'battery last'* heeft 1 kandidaat ouder namelijk *'battery life'*. Er is een overeenkomst van 75% dus wordt *'battery life'* ook effectief een ouder van *'battery last'*.

  …
- Bigram$_{31}$ 'drain battery' heeft als kandidaat ouders 'battery life' en 'battery last' met telkens een overeenkomst van 56%. De kandidaat ouder die het eerste in rang is wordt gekozen namelijk 'battery life'.

  …

Toegepast op de kandidaat productfeatures voor mobiele telefoon geeft dit volgend resultaat:

```
battery life
        battery last
        drain battery
```

```
                complaint battery
                battery drain
        life battery
                problem battery
                better battery

touch screen
        home screen

sound quality
        call quality
                picture quality
                high quality
                call text
                camera quality
        quality sound
        audio quality

screen protector
        screen size
        screen resolution
        screen cover

car charger
        need charge
        wall charger
        battery charge
….
```

### *Evaluatie*

Een analyse van de volledige lijst van toewijzingen wees uit dat deze methode wanneer een eigenschap kandidaat was om gekoppelde te worden aan een kandidaat ouder in 56% van de gevallen de juiste koppeling wist uit te voeren. Dit is ruim onvoldoende om van een betrouwbare werkmethode te spreken.

### 6.1.10 Oriëntatie bepalen van besproken product features

Voor de oriëntatiebepaling wordt van de assumptie uitgegaan dat wanneer de rating voor een beoordeling negatief is, alle eigenschappen die in deze beoordeling besproken worden negatief georiënteerd zijn. Tegenovergesteld worden alle besproken eigenschappen als positief geclassificeerd wanneer de productbeoordeling een positieve rating heeft.

Om producten met elkaar te vergelijken wordt voor elke productbeoordeling nagegaan welke producteigenschappen er besproken worden. Hierbij wordt gebruik gemaakt van de lijst bigrams die in de vorige sectie gevonden werd gesorteerd op basis van IDF. Elk besproken features krijgt dezelfde oriëntatie als de algemene beoordeling. Vervolgens wordt op productniveau een gemiddelde score berekenend voor elk van de besproken eigenschappen.

| ***Iphone 4G*** | | ***Virgin Mobile - LG Optimus V No-Contract Mobile Phone - Black*** | |
|---|---|---|---|
| battery life | 4,64 | battery life | 4,5 |
| touch screen | 4,67 | touch screen | 4,80 |
| screen protector | 4,00 | screen protector | 5,00 |
| sound quality | 5,00 | sound quality | 5,00 |
| call quality | 5,00 | call quality | 5,00 |
| battery last | 5,00 | battery last | 5,00 |
| car charger | N/A | car charger | N/A |
| customer service | 4,00 | customer service | N/A |
| memory card | N/A | memory card | 4,00 |
| android market | N/A | android market | 5,00 |

# 7 Conclusie

Bij de start van deze materproef werden drie initiële onderzoekvragen gesteld:

1. *Welke zijn de meest gebruikte classificatietechnieken binnen Text Mining?*
2. *Hoe kunnen producteigenschappen herkend worden in een verzameling van productbeoordelingen?*
3. *Wat is de performantie van een Naïve Bayes classifier in een PHP/MySQL omgeving?*

In dit afsluitend hoofdstuk wordt nagegaan in hoeverre deze vragen beantwoord zijn aan de hand van de uitgevoerde literatuurstudie en praktijkexperimenten.

## *Een evaluatie van de meest gebruikte classificatietechnieken binnen Text Mining*

Op basis van een verkennende literatuurstudie werd er nagegaan welke de meest voorkomende tekst-classificatiemethoden zijn. De geïdentificeerde methoden zijn: *Naïve Bayes, beslissingsbomen, neurale netwerken* en *Support Vector Machines*. Voor elk van deze classificatiemethoden werd onderzocht wat de sterkten en zwakten zijn. Een *Naïve Bayes classifier* blinkt vooral uit in zijn eenvoud waardoor de logica achter een classificatie eenvoudig te interpreteren is. Dit kan een onderzoeker helpen bij het identificeren van verbanden en discriminerende eigenschappen tussen groepen. Naïve Bayes is ook een van de snelste classificatie technieken. Het trainen van de classifier op een initiële dataset gaat vrij snel ten opzichte van de andere classificatiemethoden en het is mogelijk om de classifier op een later moment bij te trainen zonder dat het model van nul opnieuw dient opgebouwd te worden. Hoewel een Naïve Bayes model uitgaat van de assumptie van onafhankelijke features wat de facto onjuist is, behaalt deze methode opmerkelijke goede classificatieresultaten.

*Beslissingsbomen* zijn net als de Naïve Bayes classifier zeer eenvoudig te interpreteren. Door de verschillende paden te volgen kan een onderzoeker inzicht verwerven in de structuur van de data en verbanden identificeren. Beslissingsbomen bewijzen vooral hun nut bij kleine datasets maar wanneer de dimensionaliteit van een model toeneemt, worden beslissingsbomen al snel te complex voor interpretatie en vermindert het classificatieresultaat.

Een model op basis van *neurale netwerke*n is een krachtige classificatiemethode. De logica achter de werking van een neuraal netwerk is echter complex en de verbanden zijn niet meteen zichtbaar voor de onderzoeker *('black box methode').* Bijkomend is het classificatieresultaat vaak minder goed dan bij Support Vector Machines waardoor het gebruik van neurale netwerken voor tekstclassificatie afneemt ten voordele van Support Vector Machines.

*Support Vector Machines* maakt gebruik van afstandsmaten om een hypervlak te berekenen dat een maximale scheiding geeft tussen twee groepen. De berekeningstijd kan bij een hoogdimensionale

vectorruimte zeer snel oplopen. Daarom wordt er gebruik gemaakt van de Kernel Trick. De standaardversie van Support Vector Machines maakt gebruik van een Kernel op basis van de dot-product functie. Hierbij wordt er steeds van uit gegaan dat de data lineair deelbaar is. Dit is vaak niet het geval. Om hieraan tegemoet te komen zijn er intussen voor Support Vector Machines tal van Kernels ontwikkeld die ook in staat zijn om niet lineaire classificaties uit te voeren. Net als neurale netwerken vallen Support Vector Machines onder de categorie van *'black box methoden'* waardoor interpretatie van de onderliggende verbanden zo goed als onmogelijk is.

De *correctheid van classificeren* werd onderzocht via RapidMinder, een openbron softwarepakket voor Datamining en dat ook specifieke functies bevat voor Text Mining. Uit dit praktijkonderzoek kwamen ongeveer dezelfde bevindingen als uit de literatuurstudie. De meest performante classificatiemethode is Support Vector Machines. Vervolgens kwam classificatie op basis van neurale netwerken en op de laatste plaats beslissingsbomen. De werking van Naïve Bayes werd onderzocht via een zelf geschreven PHP script en behaalde een beter classificatieresultaat dan de Support Vector Machines. Omdat de gebruikte software dermate verschillend is, is het niet mogelijk om met voldoende zekerheid te besluiten dat Naïve Bayes bij de classificatie van productbeoordelingen wel degelijk een significant beter resultaat behaalt dan de andere methoden.

### *Technieken voor het herkennen van producteigenschappen*

De eerste technieken die besproken werden waren frequentiegebaseerde methoden. Hierbij wordt nagegaan hoe vaak een bepaald feature voorkomt in een verzameling tekst, of hoeveel documenten dit feature bevatten. Er wordt bij deze methoden van uit gegaan dat features die vaak voorkomen binnen een bepaalde categorie een groter belang hebben. Hoewel deze veronderstelling op het eerste zicht simplistisch lijkt te zijn is er toch literatuur beschikbaar die goede resultaten op basis van deze technieken beschrijft en de bruikbaarheid werd ook bevestigd via een praktijkonderzoek.

Alternatieve methoden die onderzocht werden zijn de statistische methode Chi-kwadraat, Non-Negative Matrix Factorization en *'Red Opal'*. De complexe *'Red Opal'* methode haalde vergelijkbare resultaten als de eenvoudige frequentiegebaseerde methoden. De Chi-kwadraatmethode was in staat om enkele relevante producteigenschappen te herkennen maar het aandeel relevante eigenschappen lag wel veel lager dan bij de frequentiegebaseerde methoden.

Binnen deze vraag werd er ook gekeken of er een verschil was in resultaat bij het gebruik van unigrams, bigrams (features bestaande ui twee termen) of trigrams (features bestaande uit drie termen). De bigrams haalden veruit de beste resultaten.

### *Performantie van een Naïve Bayes classifier in een PHP/MySQL omgeving*

PHP als programmeertaal en MySQL als database is een van de meest gebruikte combinaties voor het ontwikkelen van dynamische webapplicaties. Aangezien deze masterproef handelt over online productbeoordelingen wordt er ook gekeken naar de mogelijkheden om de besproken technieken toe te passen in online applicaties. Een belangrijke factor hier is de verwerkingstijd. Vanuit een kort voorafgaand literatuuronderzoek werd Naïve Bayes al aangeduid als een goede kandidaat.

Om de bruikbaarheid te testen werd een PHP-klasse geschreven die in staat is om enkele data preprocessingtechnieken toe te passen, de Naïve Bayes classifier te trainen en classificaties van nieuwe documenten uit te voeren. De training van de classifier gebeurde aan de hand van 91.514

productbeoordelingen. De classifier had ongeveer 30 minuten nodig om deze data te verwerken. Het classificeren van nieuwe documenten neemt gemiddeld ongeveer 0,8 seconden in beslag. Bijkomend waren de classificatieresultaten van de Naïve Bayes classifier verrassend goed en moest deze zeker niet onderdoen voor andere technieken die getest werden via RapidMiner. Op basis van het uitgevoerde praktijkonderzoek kan besloten worden dan Naïve Bayes een van de meest vooraanstaande methoden is voor gebruik van classificatie in online webapplicaties.

### *Mogelijkheden voor verder onderzoek*

In het praktijkonderzoek werden momenteel nog niet alle methoden onderzocht voor het koppelen van features die eenzelfde producteigenschap beschrijven. Het is zeker interessant om deze methoden ook aan een praktijkonderzoek te onderwerpen binnen het domein van productbeoordelingen.

# 8 Samenvatting

We leven in een informatietijdperk waarin we veel meer data verzamelen dan we ooit manueel kunnen verwerken. De informatie- en economische waarde die verborgen zit in tal van gegevensbronnen is gigantisch. Meer en meer bedrijven zijn zich hier bewust van en zijn op zoek naar methoden om het potentieel dat begraven ligt in gegevensbronnen en databanken naar boven te brengen. Eén bron van informatie die de laatste jaren enorm in omvang is toegenomen zijn online productbeoordelingen. Op tal van webwinkels zijn honderden tot wel duizenden productbeoordelingen terug te vinden. Deze productbeoordelingen bieden een unieke kans aan organisaties om de meningen en verwachtingen van consumenten te leren kennen. De uitdaging bestaat er nu uit om de onderliggende informatie naar boven te brengen in een compact en bruikbaar formaat.

Het traditionele antwoord hierop is het gebruik van datamining. Datamining tracht om aan de hand van verschillende modellen en technieken de onderliggende verbanden in een grote verzameling data naar boven te brengen. Een van de grootste beperkingen van dataminingtechnieken is dat deze nood hebben aan een gestructureerde input met variabelen. Veel van de informatie die we opslaan is echter tekst, en is niet meteen bruikbaar in dataminingtoepassingen. Text Mining is een relatief nieuw domein dat tracht met deze beperking om te gaan. Text Mining bouwt hiervoor verder op een ruime theoretische basis vanuit Natural Language Processing en Information Retrieval. In deze masterproef wordt kort de achtergrond van Text Mining geschetst samen met enkele toepassingen.

Een van de voornaamste toepassingen van Text Mining is het classificeren van documenten. Aan de hand van een literatuurstudie wordt er een verkennend onderzoek gedaan naar de meest gebruikte classificatiemethoden binnen Tekst Mining. In een tweede fase wordt aan de hand van een kort praktijkonderzoek nagegaan wat de bruikbaarheid is van de verschillende classificatiemethoden op productbeoordelingen. Er wordt hier vooral nagegaan of een productbeoordeling als positief of negatief geclassificeerd kan worden (*sentiment analysis*). Omdat de toepassingen die gebruik maken van Text Mining technieken voor de analyse van productbeoordelingen vaak in een online omgeving worden uitgevoerd wordt de performantie getest van een Naïve Bayes classifier binnen een php/mysql omgeving.

Een tweede toepassing van Text Mining is Information Extraction. Dit is de stukjes informatie uit tekst selecteren die werkelijk interessant zijn voor een onderzoek. Specifiek gaan we hier op zoek naar technieken om producteigenschappen te herkennen in een verzameling productbeoordelingen. Via een literatuurstudie wordt er een verkennend onderzoek gedaan naar de meest voorkomende technieken voor het herkennen van zogenoemde ‘features’. Vervolgens wordt aan de hand van een praktijkonderzoek nagegaan of deze technieken ook gebruikt kunnen worden voor het herkennen van producteigenschappen.

# 9 Bijlagen

## 9.1 Lijst van afkortingen

| | |
|---|---|
| AI | Artificiële Intelligentie |
| CNB | Complement Naïve Bayes |
| HMM | Hidden Markov Model |
| IDF | Inverse Document Frequency |
| KDD | Knowledge Discovery in Databases |
| ML | Maximum Likelihood |
| MNB | Multinomial Naïve Bayes |
| NLP | Natural Language Processing |
| NMF | Non-Negative Matrix Factorization |
| PFE | Product Feature Extraction |
| POS | Part Of Speech |
| VMM | Visible Markov Model |

## 9.2 Glossarium

#### 9.2.1.1 Classifier

Een classifier tracht een nieuwe onbekende observatie toe te wijzen aan een bepaalde groep op basis van een verzameling bekende observaties waarvan de groep waartoe deze behoren gekend is. Een trainingset van data is dus een noodzakelijke voorwaarde voor een classifier.

#### 9.2.1.2 Supervised Learning

Supervised learning  maakt gebruik van een training set van bekende voorbeelden. Elk voorbeeld is een paar van een input object (vaak een vector) en een bekende output waarde. Het resultaat van een supervised learning algoritme is een functie  die een discrete waarde kan aannemen (*classifier*) of een continue waarde (*regressiefunctie*). De afgeleide functie tracht de correcte output te vinden voor een nieuw onbekend voorbeeld. (Wikipedia)

#### 9.2.1.3 Tokens

Tokens zijn een opeenvolging van karakters. Tokens kunnen delen van zinnen, delen van woorden, of andere patronen zijn.

#### 9.2.1.4 Features

Features zijn in deze masterproef tokens of een opeenvolging van tokens die een producteigenschap beschrijven. Deze kunnen expliciet zijn zoals bijvoorbeeld “lange batterijduur” of impliciet zoals “kan een week zonder opladen”.

#### 9.2.1.5 Unsupervised Learning

Unsupervised learing maakt geen gebruik van een gelabelde trainging set. Een unsupervised learning algoritme probeert een verborgen structuur te vinden in een verzameling data. (Wikipedia)

#### 9.2.1.6 Smoothing

Smoothing wordt in het domein van Natural Language Processing (NLP) vaak gebruikt op het probleem van ontbrekende data *(data sparsity)* aan te pakken. Wanneer er onvoldoende data beschikbaar is is de kans groot dat voor een bepaald token de maximum likelihood estimate 0 is. Dit ingangspunt is voor een NLP model foutief. Het is niet omdat een token niet voorkomt in de training data dat dit token onmogelijk kan voorkomen in test data.

In het geval dat er toch voldoende data beschikbaar is en ontbrekende data geen probleem vormt kan het aangewezen zijn om een complexer model te ontwikkelen. Bijvoorbeeld door woordcombinaties op te nemen in het model (n-grams). De kans dat een woordcominatie wel voorkomt in de test data en niet in de training data is hoger dan bij het gebruik van enkele tokens (unigrams). In dit geval dient smoothing weer toegepast te worden.

Smoothing kan in de meeste gevallen een positief effect hebben op het resultaat van een model zodat dat dit veel extra complexiteit toevoegt of rekenkracht vraagt. (Chen S.F., 1998)

#### 9.2.1.7 Inverse Document Frequency

Om documenten te indexeren is het eenvoudigweg ordenen van tokens naargelang hun frequentie vaak onvoldoende. Inverse Document Frequency (IDF) is een methode die een hogere waarde toekent aanmeer specifieke tokens en een lagere waarde aan veel gebruikte tokens (Konchady, 2006). De IDF $i_m$ waarde is omgekeerd proportioneel ten opzichte van het aantal documenten waarin een woord $m$ voorkomt. Voor elk woord $m$ wordt het aantal documenten $d_m$ opgezocht waarin het woord $m$ voorkomt. De IDF waarde kan vervolgens berekend worden door het totaal aantal documenten $N$ te delen door het aantal documenten waarin het woord $m$ voorkomt:

$$IDF = \frac{N}{d_m}$$

Wanneer een token veel voorkomend is en bijgevolg $d_m$ groot is benadert de IDF waarde 1. Wanneer een token slechts enkele keren voorkomt benadert de IDF waarde $N$. Vaak wordt een logaritme gebruikt om de IDF waarde af te vlakken:

$$f_m = \log(N) - \log(d_m) + 1$$

#### 9.2.1.8 POS tagging

Part Of Speech tagging of kortweg POS tagging is het proces om aan elk word in een tekst een grammaticale betekenis te geven. Een POS tagger tracht na te gaan of een woord een zelfstandig naamwoord, werkwoord, adjectief, etc. is op basis van de definitie van het woord als wel als de context.

De grote uitdaging voor een POS tagger is om de juiste betekenis aan een woord te geven afhankelijk van de context waarin het gebruikt wordt. Heel wat woorden kunnen immers een verschillende grammaticale betekenis hebben afhankelijk van de context.

De POS tagger die gebruikt wordt in deze masterproef maakt gebruik van WordNet om de grammaticale betekenis van een woord op te zoeken. WordtNet kan aangeven of een woord als zelfstandig naamwoord, werkwoord, bijvoeglijk naamwoord of bijwoord wordt gebruikt of een combinatie van de verschillende woordsoorten. Voor woorden met meer dan een grammaticale betekenis wordt in deze masterpoef gekozen voor de grammaticale betekenis met de hoogste kans. Ondanks deze op het eerste zicht simplistische toewijzingsregel kan de toewijzing en precisie halen tot wel 90% (Charniak, 1993).

#### 9.2.1.9 Hidden Markov Models

Een van de toepassingen van Hidden Markov Models is het selecteren van de meest waarschijnlijke state van een sequentie voor een gegeven observatie. Het is mogelijk om gegeven een token in een sequentie van tekst de waarschijnlijkheid te berekenen dat het tot een bepaalde state behoort (Konchady, 2006).

#### 9.2.1.10 N-grams

Een n-gram is en deel van een gegeven opeenvolging. In het geval van de analyse van productfeatures is een n-gram een deel van een zin. Unigrammen bestaan uit 1 woord (token), bigrammen uit twee opeenvolgende token, trigrammen uit drie opeenvolgende tokens,…

N-grams zijn in essentie Hidden Markov Models waarbij een bigram een eerste order Hidden Markov Model is, en trigram een tweede order Hidden Markov Model, enzoverder. Een bigram heeft een 'geheugen' van een stap in het verleden. De state van een token is enkel afhankelijk van de eerst voorgaande state. Bij een trigram is de state afhankelijk van de twee voorgaande states, enzoverder.

#### 9.2.1.11 WordNet

WordNet is een semantische database voor de Engelse taal (Wikipedia). Zelfstandige naamwoorden, werkwoorden, bijvoeglijke naamwoorden en bijwoorden worden ondergebracht in zogenaamde *'synsets'*. Dit zijn sets van synoniemen die een zelfde begrip beschrijven. De *synsets* worden met elkaar gelinkt via semi-semantische en lexicale relaties (Princeton University). De speciale structuur van WordNet maakt het tot een grote ondersteuning voor *Natutal Language Processing* toepassingen zoal *POS Tagging* of het leggen van semantische relaties.

#### 9.2.1.12 Dot-Product

Het dot-product (ook wel inwendig product, inproduct of scalair product genoemd) is een methode om twee vectoren met elkaar te vermenigvuldigen.

Wanneer er twee vectoren zijn, $a=(a_1, a_2, a_3, \ldots)$ en $b=(b_1, b_2, b_3, \ldots)$ dan kan het dot-product geschreven worden als:

$$a \bullet b = \sum_{i=1}^{n} a_i\, b_i = a_1 b_1 + a_1 b_1 + \cdots + a_n b_n$$

Het dot-product kan alternatief ook geschreven worden als:

$$a \bullet b = |a||c| \cos\theta$$

Waarbij $\theta$ de hoek is tussen twee vectoren.

# 9.3 Volledige broncode naïve bayes classifier (class_naive_bayes.php)

```
<?php
include('inc_db.php');

class NaiveBayes{

	// list of basewords
	var $basewords="a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9
	about add ago after all also an and another any are as at be because been before
	being between both but by came can come could did do does due each else end
	far for from get got has had have he her here him himself his how if into is
	it its just let lie like low make many me might must my never no
	nor not now of off old on only or other out over per pre put re said same see
	she should since so some still such take than that the their theirs them themselves then there these
	they this those through to too under up use very via want was way we well were
	what when where which while who will with would yes yet you your";

	// DATABASE FUNCTIONS


	// add training example to the database
	// $catgeory is the categoryID
	function trainText($classifier,$categeory,$traindata){
		// convert the training data to tokens
		$classify = $this->textToTokens($traindata);

		// check if the token allready exists
		foreach($classify as $key => $value){
		$sql_check=mysql_query("SELECT frequencyID , frequency
							FROM nb_classifier_frequencies
							WHERE categoryID='".$categeory."'
							AND token='".addslashes($key)."'") or die(mysql_error());
		// update frequency if token exists for this catgeory
		if(mysql_num_rows($sql_check)<>0){
			$updatedfrequency=0;
			$old=mysql_fetch_assoc($sql_check);
			$updatedfrequency=$old['frequency']+$value;
			mysql_query("UPDATE nb_classifier_frequencies
							SET frequency='".$updatedfrequency."'
							WHERE frequencyID='".$old['frequencyID']."'") or die(mysql_error());
			}
		// else create new entry for token with frequency 1
		else{
			$wordtype_n=0;
			$wordtype_v=0;
			$wordtype_a=0;
			$wordtype_s=0;
			$wordtype_r=0;
			// check if token exists in wordnet
			$sql_checkwordnet=mysql_query("SELECT * FROM words WHERE lemma='".addslashes($key)."'")
			                                                                    or die(mysql_error());
				// if token exists in the wordnet dictionary look up the wordtypes
				// n=Noun, v=Verb, a=Adjective, s=Adjective Sattelite, a=Adverb
				if(mysql_num_rows($sql_checkwordnet)<>0){
					$sql_wordtype=mysql_query("SELECT pos FROM dict_table WHERE
					lemma='".addslashes($key)."' GROUP BY pos") or die(mysql_error());
					while($pos=mysql_fetch_assoc($sql_wordtype)){
						switch ($pos['pos']){
							case 'n':
								$wordtype_n=1;
								break;
							case 'v':
								$wordtype_v=1;
								break;
							case 'a':
								$wordtype_a=1;
```

```
                                                                        break;
                                                                case 's':
                                                                        $wordtype_s=1;
                                                                        break;
                                                                case 'r':
                                                                        $wordtype_r=1;
                                                                        break;
                                                                }
                                                        }
                                                }
                                        // add token to the database
                                        mysql_query("INSERT INTO nb_classifier_frequencies (classifierID, categoryID, token, frequency,
                                                wordtype_n, wordtype_v, wordtype_a, wordtype_s, wordtype_r)
                                                VALUES ('".$classifier."', '".$categeory."', '".addslashes($key)."', '".$value."', '".$wordtype_n."',
                                                '".$wordtype_v."', '".$wordtype_a."', '".$wordtype_s."', '".$wordtype_r."')")
                                                or die(mysql_error());
                                        }
                        }
return true;
}

// TEXT FUNCTIONS

// split a text into words
function splitText($text){
        $rawTokens=split("[^-_A-za-z0-9]", $text);
        return $rawTokens;
}

/** clean a string from the diacritics

        @author Antoine Bajolet [phpdig_at_toiletoine.net]
        @author SPIP [http://uzine.net/spip/]

        @return string clean string
        @param  string string with accents
*/
function cleanString($string)
{
        $diac =
                /* A */   chr(192).chr(193).chr(194).chr(195).chr(196).chr(197).
                /* a */   chr(224).chr(225).chr(226).chr(227).chr(228).chr(229).
                /* O */   chr(210).chr(211).chr(212).chr(213).chr(214).chr(216).
                /* o */   chr(242).chr(243).chr(244).chr(245).chr(246).chr(248).
                /* E */   chr(200).chr(201).chr(202).chr(203).
                /* e */   chr(232).chr(233).chr(234).chr(235).
                /* Cc */  chr(199).chr(231).
                /* I */   chr(204).chr(205).chr(206).chr(207).
                /* i */   chr(236).chr(237).chr(238).chr(239).
                /* U */   chr(217).chr(218).chr(219).chr(220).
                /* u */   chr(249).chr(250).chr(251).chr(252).
                /* yNn */ chr(255).chr(209).chr(241);
        return strtolower(strtr($string, $diac, 'AAAAAAaaaaaaOOOOOOooooooEEEEeeeeCcIIIIiiiiUUUUuuuuyNn'));
}

// convert a string to a token
function stringToToken($string)
{
        $rawToken = trim($string); // remove whitespace
        $rawToken = strtolower($rawToken); // convert rawToken to lowercase
        $rawToken = $this->cleanString($rawToken);
        return $rawToken;
}

// extract all tokens from a text
// and their frequencies
function textToTokens($text){
        $tokens=array(); // final token array

        $rawTokens = $this->splitText($text); // break down the text
```

```
		foreach($rawTokens as $rawToken){
			$rawToken = $this->stringToToken($rawToken); // convert string to token

			// check if rawtoken exists in wordnet
			// if not perform stemming
			// check is used to reduce faults of the stemming algorithm
			$sql_wordnet=mysql_query("SELECT * FROM words WHERE lemma='".addslashes($rawToken)."'")
				or die(mysql_error());
			if(mysql_num_rows($sql_wordnet)==0){
			$rawToken = $this->stem($rawToken); // stem token
			}

			// check if token is between 3 and 15 characters
			if(strlen($rawToken)>=3 && strlen($rawToken)<=15){

				// check if rawToken is a baseword
				if(strpos($this->basewords, $rawToken)){
				//echo'<div>excluded: '.$rawToken.'</div>';
				}
				// if not register occurence of the token
				else{
					if(!isset($tokens[$rawToken])){
					$tokens[$rawToken]=1; // create new final token ans set wordcount to none
					}
					else{
					$tokens[$rawToken]++; // increase token count
					}
				}

			}
		}

		return $tokens;
}

// count the number of tokens in all of the classifier training data
// = SUM( unique token * frequency )
function tokenInClassifier($classifier,$token){
		$sql=mysql_query("SELECT SUM(frequency) AS total FROM nb_classifier_frequencies
				WHERE classifierID='".$classifier."' AND token='".addslashes($token)."'") or die(mysql_error());
		$total=mysql_fetch_assoc($sql);
		return $total['total'];
		}

// count number of tokens in a category
function tokenInCategory($category,$token){
		$sql=mysql_query("SELECT SUM(frequency) AS total FROM nb_classifier_frequencies
				WHERE categoryID='".$category."' AND token='".addslashes($token)."'") or die(mysql_error());
		$total=mysql_fetch_assoc($sql);
		return $total['total'];
		}

// count the number of categories in a classifier
function countCategories($classifier){
		$sql=mysql_query("SELECT * FROM nb_classifier_categories WHERE classifierID='".$classifier."'")
				or die(mysql_error());
		$count=mysql_num_rows($sql);
		return $count;
		}

// count the numer of token occurences in a category other than $category for a given classifier
// used for Complement Naive Bayes
function tokenInOtherCategories($token,$classifier,$category){
		$sql=mysql_query("SELECT SUM(frequency) AS total FROM nb_classifier_frequencies
				WHERE categoryID<>'".$category."' AND classifierID='".$classifier."'
				AND token='".addslashes($token)."'") or die(mysql_error());
		$total=mysql_fetch_assoc($sql);
		return $total['total'];
		}

// count the total number of token occurences for categories other than $category
```

```
// used for Complement Naive Bayes
function countOccurencesInOtherCategories($classifier,$category){
	$sql=mysql_query("SELECT SUM(frequency) AS total FROM nb_classifier_frequencies
				WHERE categoryID<>'".$category."' AND classifierID='".$classifier."'") or die(mysql_error());
	$total=mysql_fetch_assoc($sql);
	return $total['total'];
	}

// return the categoryID's for a given classifier
function categories($classifier){
	$categories=array();
	$sql=mysql_query("SELECT categoryID FROM nb_classifier_categories WHERE classifierID='".$classifier."'")
	                                                                                    or die(mysql_error());
	while($cat=mysql_fetch_assoc($sql)){
		array_push($categories,$cat['categoryID']);
		}

	return $categories;
	}

// MULTINOMIAL NAIVE BAYES
// input is an array with the token as key and the
// frequency of occurence in the text to be classified as the value
function MNB($tokenArray,$classifier){
	// initialize variables
	$result=array();
	$result['token']=array();
	$result['MNB']=array();
	$result['MNBlog']=array();
	$result['percentage']=array();
	$countCategories=$this->countCategories($classifier);
	$categories=$this->categories($classifier);

	// set the initial MNB result of each category to 1
	foreach($categories as $cat){
		$result['MNB'][$cat]=1;
		}

	foreach($tokenArray as $token => $frequency){
		$result['token'][$token]=array();

		foreach($categories as $cat){
			// fetch the MNB result for each category for the given token
	$prob=($this->tokenInCategory($cat,$token)+1)/($this->tokenInClassifier($classifier,$token)+$countCategories);
			$result['token'][$token][$cat]=$prob;

			// update the final MNB result for the category
			// update once for every occurence of the token in the text to be classified
			for($i=1;$i<=$frequency;$i++){
				$result['MNB'][$cat]=$result['MNB'][$cat]*$prob;
				}
			}

		}

	// convert the MNB likelihood to MNB log likelihood
	// and select the most likely category
	$totalMNB=0;
	foreach($result['MNB'] as $cat => $value){
		// concert to MNB log likelihood
		$result['MNBlog'][$cat]=log($value);
		// select most likely category
		if(!isset($result['category'])){ $result['category']=$cat; }
		if($result['MNB'][$cat]>$result['MNB'][$result['category']]){ $result['category']=$cat; }
		// calculate sum on MNB's
		// used to calculate percentages
		$totalMNB=$totalMNB+$value;
		}

	// calculate percentages
	foreach($result['MNB'] as $cat => $value){
```

```
                $result['percentage'][$cat]=$value/$totalMNB;
                }

        return $result;
        }

//
function classifyMNB($text,$classifier){
        $tokens=$this->textToTokens($text);
        $result=$this->MNB($tokens,$classifier);
        return $result;
        }



// COMPLEMENT NAIVE BAYES
// input is an array with the token as key and the
// frequency of occurence in the text to be classified as the value
function CNB($tokenArray,$classifier){
        // initialize variables
        $result=array();
        $result['token']=array();
        $result['CNB']=array();
        $countCategories=$this->countCategories($classifier);
        $categories=$this->categories($classifier);

        $occurencesInOtherCategories=array();
        foreach($categories as $cat){
                $occurencesInOtherCategories[$cat]=$this->countOccurencesInOtherCategories($classifier,$cat);
                }

        // set the initial CNB count of each category to 0
        foreach($categories as $cat){
                $result['CNB'][$cat]=0;
                $result['CNBlog'][$cat]=0;
                }

        foreach($tokenArray as $token => $frequency){
                $result['token'][$token]=array();
                $subselect=0;

                foreach($categories as $cat){
                        // fetch the CNB result for each category for the given token
$prob=($this->tokenInOtherCategories($token,$classifier,$cat)+1)/($occurencesInOtherCategories[$cat]+$countCategories);
                        $result['token'][$token][$cat]=$prob;

                        // select category with lowest CNB
                        if($subselect==0){ $subselect=$cat; }
                        elseif($result['token'][$token][$cat]<$result['token'][$token][$subselect]){ $subselect=$cat; }

                        $result['CNBlog'][$cat]=$result['CNBlog'][$cat]+log($prob);

                        }

                // update the final CNB count for the category
                // update once for every occurence of the token in the text to be classified
                for($i=1;$i<=$frequency;$i++){
                        $result['CNB'][$subselect]++;
                        }

                }

        // select category with the highes CNBlog
        foreach($categories as $cat){
                if(!isset($result['category'])){ $result['category']=$cat; }
                elseif(abs($result['CNBlog'][$cat])>abs($result['CNBlog'][$result['category']])){ $result['category']=$cat; }
                }

        return $result;
        }
```

```
        //
        function classifyCNB($text,$classifier){
                $tokens=$this->textToTokens($text);
                $result=$this->CNB($tokens,$classifier);
                return $result;
                }


?>
```

# 9.4 Voorbeeld classificatie[16]

### *Grand Theft Auto IV Platinum Hits - Xbox 360*


### *Product reviews*

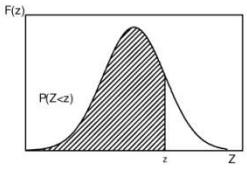
| | *Rating* | *MNB* | *CNB* |
|---|---|---|---|
| **Waste of time**<br>It is the most boring GTA by far. If you want fun Saints Row 2 is the way to go. Check out Yahtzees Saints Row Review it say it all. | 1.00 | negative | negative |
| **One of the best out there**<br>This is one of the best games you can buy.The missions are fun and classic and each character has a different attitude to everything. When not playing missions there are many fun things to do, just look for them. The graphics arent great, but it is easy to overlook that. You have to buy this game, it is amazing! | 5.00 | positive | positive |
| **good game to play when bored**<br>GTA 4 is a good overall game but thats about it. unless you like the prevouis games than some may not find this a game good, or if your used to first shooters like cod4 than you will find the fighting portion of the game disappointing. this game does have awesome graphics and the missions will keep you busy for awhile but the driving and fighting part of the game hurts the overall rating | 2.00 | positive | positive |
| **Really awesome game**<br>Personally, I think this game is awesome. The best one out of the series so far. There might be some disappointments such as no airplanes, no military, no car customizing, no trains, and etc, but I think that all the realism and great graphics that they put into the game made up for some of the loss. The cheats are harder to put in since you have to enter through his cell phone, but I like that better because I'm such a cheat junkie, and with all the time you have to spend going on his phone and dialing the numbers in, it just isn't worth using cheats in this game. I would deff. recommend this game to a friend. | 5.00 | positive | positive |
| **OMG! YES**<br>i just got my game @ bestbuy i only had an hour to play before work and man im @ work and i want more, i was left at a cliff hanger lol this game is awsome good graphics too and the gameplay is phenomenal so yea, pick up ur copy and try it | 5.00 | positive | positive |
| **Awesome!**<br>Tons of fun to play! I also have the game add-ons, and those are killer also! A must buy! | 5.00 | positive | positive |
| **Yet again, An awesome game!**<br>Ever since I was about 8 I discovered Grand Theft Auto. It Actually started with The Second Grand Theft Auto. One of my brothers friends, played it. I never asked to play, until one day my brother brought home Grand Theft Auto 3. I asked him if I could try. I did, Went absolutely insane with the gun. It is horrible actually. I just shot every random person. Than he bought Vice City. I loved it too. Years later, I met a friend in 7th grade, I went to his house, was playing a video game, He was playing Vice City. I screamed and begged him for me to try it. I haven't played since I was 9. than He also had San Andrea's. Than the same friend reserved Grand Theft Auto 4. I was freaking out, especially when he got it. I played it with him. Once I think I played it for 3 days straight. Today, I bought my own, and I'm playing it as I type this. I can seriously either pass missions, or if I'm bored I'll go into a park, and beat people up. Its a good, good game just, Its a little addictive. Especially when It's summer. | 5.00 | negative | positive |
| **very good**<br>The very good return to the grand theft auto series great graphics, storyline, weapons everything and i never get tired of it. | 5.00 | positive | positive |
| **Best GTA Game**<br>Best GTA game so far. Definetly worth buying. Good graphics and really fun with some suprising new features. 5/5 star game | 5.00 | positive | positive |
| **Great but easy** | 4.00 | positive | positive |

[16] De tabel is beperkt opgenomen omwille van plaatsbesparing en is ter illustratie van het classificatieresultaat van de PHP-classifier op basis van MNB of CNB

| | | | |
|---|---|---|---|
| GTA is an amazing series of games. This particular GTA was entertaining and extremly realistic. I loved playing the story mood as well as I liked playing on Xbox Live. Unfortunatly the story mood was a tad bit easy and didn't fully challenge me. But overall 4 star game. Buy and Enjoy! | | | |
| **THE BEST GTA EVER**<br>This is a great game with super graphics. Extremely fun missions. I LOVE IT!!! Not recommended for children under 17! Very strong se xxx ual content, drug and alcohol usage and profanity. Coge | 5.00 | positive | positive |
| **buy it**<br>this game is awsome but the only thing that sucks is the car handling and bad for kids under 17 but im younger and i still play it because i like gorey games | 5.00 | positive | positive |
| **gonna be the best game out there**<br>i bought every GTA game there is for ps ..xbox.. and now xbox360.. i got everyone of tha been out and i watch there graphics and abilitys grow on this game and everytime it gets better ... i seen it get cycles to boats to racing helicopters now online..!! it crazy there online game play will not be i see u shoot shoot .. it gonna be duck and hide plane races and ur supposed to get all free roaming abilitys to every location online to and tha takes alot of work.. tha why it took them so long to develope this game .. it gonna be the best out there u get tons of side missions .. very nice rides in this one .. more open areas with better graphics.. u get races and everything .. if u like games with a storyline plus fun side missions plus loads of free roaming.. this is def the game u just HAVE TO BUY!! BUY!! BUY!! BUY!1 if u dont get it now ur missing out big time .. get it before it to late!! | 5.00 | positive | positive |
| **Great Game**<br>Great game. Would recommend it to anyone who remotely likes the GTA series. | 5.00 | positive | positive |
| **better than Halo**<br>This is the best ever grand theft auto. I can not put it down. I love it one GREAT game | 5.00 | positive | positive |
| **Arugably the greatest game... ever.**<br>I purchased this game a couple of months ago, played through it twice, and have only good things to say about it! The storyline is unlike any other storyline that I've seen in a game before. It's original, fun, and there are even some twists and turns that you'll come across throughout the game. There are MANY different characters that you'll meet throughout the story, and you'll even have a chance to kill some of them. Hehe. The gameplay is fantastic. The driving is realistic, though this can prove to be a pain in the rear sometimes. The voice acting is astounding; you'll find yourself so immersed in the game, that you'll start thinking that you're IN Liberty City, which also looks extroadinary. There are tons of mini-games, such as going bowling and playing darts. (Niko's also able to obtain girlfriends within the game; some of them have special perks which can benefit you as you play!) The variety of weapons, guns and melee weapons, isn't as broad as the previous 'Grand Theft Auto' titles, but this only makes the game more realistic. Of course, you still have the trusty ol' RPG that you can take down choppers with! Overall, I give this game a 5/5, not because I'm following a trend of 5/5's, but because the game DESERVES that rating. It was well worth the wait to purchase this game, and I must say, it was probably the best $60 I've ever spent. Trust me when I say this: If you go out and buy this game today, you will NOT be dissapointed. | 5.00 | positive | positive |
| **Worth the Money Spent!**<br>I purchased this game for my son and he had been anxiously awaiting its release. According to him, it highly surpasses all the other GTA games! I purchased it 3 days ago when it was released and he has spent every available minute playing it. The graphics on this game are awesome! | 5.00 | negative | positive |
| **BEST GAME EVER**<br>Sure, Grand Theft Auto is Grand Theft Auto.. but hey, this game completely blows the other ones away. The graphics, the game play, the real factor. This is truely an amazing buy. If your looking for that "great game" that never ends and satisfies all your needs, this is it. Buy it now, believe me. | 5.00 | positive | positive |
| **Best GTA Yet!**<br>This game is amazing, its graphics are outstanding and the storyline is great! this game will provide you with a minimum of 30-45 hours of storyline play time. it can be thousands of hours of playtime just going around killing people or having the cops chase you. I would deffinatly recommend this game to many others and have bought it as a gift for my brother already. this game is worth every penny you spend on it. | 5.00 | positive | positive |
| **Very Fun, Great Graphics, Never Gets Boring!**<br>GTA 4 is probably most known for its stunning graphics and amazing story. You play as Niko Belic a Russian imigrant. You will go through some crazy stuff as you try to save yuor cousin roman, and yourself. The setting takes place in Liberty City which is clearly a remake of New York City. Its got everything that NY has. For example The Stastue of Liberty is The Stastue of Happieness. Anyway, you explore 3 different islands. I am not really sure what their names are though. This game does take take realistic to a whole new level by having a 5 out of 5 graphic team. Also by the killing affects. For example you can run over people and they fly all over the place. Also if there is a car on fire and you get near the fire your clothes are set a flame and you roll on the ground. Overall GTA4 is a great game. There is only 1 problem, you can't save games at any time, you have to go to the safe house. =( | 5.00 | positive | positive |
| **Sweet Game**<br>Stoked to play,played it for about 12 hours straight when I put it in the system,everything about this game was worth the wait,everything from the patch work on the streets of liberty city to the cracks in the walls when you hit them head on to the bullet holes left in your car after a shootout to the brake calipers on the comet.....this game is phenomenal,I read these negative reviews mainly on repetitive gameplay and hard driving....well folks the driving is more realistic and when you think about it everything is repetitive in a way so you cant rip on something for that but i can understand why some would find the driving hard, but I find it very very easy, I highly recommend this game to anyone 12-90 yrs of age | 5.00 | positive | positive |

# 9.5 Tabel standaard normale verdeling

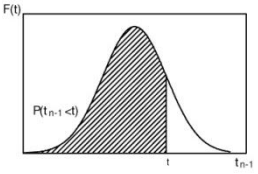
Tabel 9.1: Standaard normale verdeling




| | Tweede decimaal van $z$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $z$ | .00 | .01 | .02 | .03 | .04 | .05 | .06 | .07 | .08 | .09 |
| 0.0 | .5000 | .5040 | .5080 | .5120 | .5160 | .5199 | .5239 | .5279 | .5319 | .5359 |
| 0.1 | .5398 | .5438 | .5478 | .5517 | .5557 | .5596 | .5636 | .5675 | .5714 | .5753 |
| 0.2 | .5793 | .5832 | .5871 | .5910 | .5948 | .5987 | .6026 | .6064 | .6103 | .6141 |
| 0.3 | .6179 | .6217 | .6255 | .6293 | .6331 | .6368 | .6406 | .6443 | .6480 | .6517 |
| 0.4 | .6554 | .6591 | .6628 | .6664 | .6700 | .6736 | .6772 | .6808 | .6844 | .6879 |
| 0.5 | .6915 | .6950 | .6985 | .7019 | .7054 | .7088 | .7123 | .7157 | .7190 | .7224 |
| 0.6 | .7257 | .7291 | .7324 | .7357 | .7389 | .7422 | .7454 | .7486 | .7517 | .7549 |
| 0.7 | .7580 | .7611 | .7642 | .7673 | .7704 | .7734 | .7764 | .7794 | .7823 | .7852 |
| 0.8 | .7881 | .7910 | .7939 | .7967 | .7995 | .8023 | .8051 | .8078 | .8106 | .8133 |
| 0.9 | .8159 | .8186 | .8212 | .8238 | .8264 | .8289 | .8315 | .8340 | .8365 | .8389 |
| 1.0 | .8413 | .8438 | .8461 | .8485 | .8508 | .8531 | .8554 | .8577 | .8599 | .8621 |
| 1.1 | .8643 | .8665 | .8686 | .8708 | .8729 | .8749 | .8770 | .8790 | .8810 | .8830 |
| 1.2 | .8849 | .8869 | .8888 | .8907 | .8925 | .8944 | .8962 | .8980 | .8997 | .9015 |
| 1.3 | .9032 | .9049 | .9066 | .9082 | .9099 | .9115 | .9131 | .9147 | .9162 | .9177 |
| 1.4 | .9192 | .9207 | .9222 | .9236 | .9251 | .9265 | .9279 | .9292 | .9306 | .9319 |
| 1.5 | .9332 | .9345 | .9357 | .9370 | .9382 | .9394 | .9406 | .9418 | .9429 | .9441 |
| 1.6 | .9452 | .9463 | .9474 | .9484 | .9495 | .9505 | .9515 | .9525 | .9535 | .9545 |
| 1.7 | .9554 | .9564 | .9573 | .9582 | .9591 | .9599 | .9608 | .9616 | .9625 | .9633 |
| 1.8 | .9641 | .9649 | .9656 | .9664 | .9671 | .9678 | .9686 | .9693 | .9699 | .9706 |
| 1.9 | .9713 | .9719 | .9726 | .9732 | .9738 | .9744 | .9750 | .9756 | .9761 | .9767 |
| 2.0 | .9772 | .9778 | .9783 | .9788 | .9793 | .9798 | .9803 | .9808 | .9812 | .9817 |
| 2.1 | .9821 | .9826 | .9830 | .9834 | .9838 | .9842 | .9846 | .9850 | .9854 | .9857 |
| 2.2 | .9861 | .9864 | .9868 | .9871 | .9875 | .9878 | .9881 | .9884 | .9887 | .9890 |
| 2.3 | .9893 | .9896 | .9898 | .9901 | .9904 | .9906 | .9909 | .9911 | .9913 | .9916 |
| 2.4 | .9918 | .9920 | .9922 | .9925 | .9927 | .9929 | .9931 | .9932 | .9934 | .9936 |
| 2.5 | .9938 | .9940 | .9941 | .9943 | .9945 | .9946 | .9948 | .9949 | .9951 | .9952 |
| 2.6 | .9953 | .9955 | .9956 | .9957 | .9959 | .9960 | .9961 | .9962 | .9963 | .9964 |
| 2.7 | .9965 | .9966 | .9967 | .9968 | .9969 | .9970 | .9971 | .9972 | .9973 | .9974 |
| 2.8 | .9974 | .9975 | .9976 | .9977 | .9977 | .9978 | .9979 | .9979 | .9980 | .9981 |
| 2.9 | .9981 | .9982 | .9982 | .9983 | .9984 | .9984 | .9985 | .9985 | .9986 | .9986 |
| 3.0 | .9987 | .9987 | .9987 | .9988 | .9988 | .9989 | .9989 | .9989 | .9990 | .9990 |

# 9.6 Tabel T-verdeling

Tabel 9.2: t-verdeling




| | P | | | | | | |
|---|---|---|---|---|---|---|---|
| $vg$ | .750 | .900 | .950 | .975 | .990 | .995 | .999 |
| 1 | 1.000 | 3.078 | 6.314 | 12.706 | 31.821 | 63.656 | 318.289 |
| 2 | 0.816 | 1.886 | 2.920 | 4.303 | 6.965 | 9.925 | 22.328 |
| 3 | 0.765 | 1.638 | 2.353 | 3.182 | 4.541 | 5.841 | 10.214 |
| 4 | 0.741 | 1.533 | 2.132 | 2.776 | 3.747 | 4.604 | 7.173 |
| 5 | 0.727 | 1.476 | 2.015 | 2.571 | 3.365 | 4.032 | 5.894 |
| 6 | 0.718 | 1.440 | 1.943 | 2.447 | 3.143 | 3.707 | 5.208 |
| 7 | 0.711 | 1.415 | 1.895 | 2.365 | 2.998 | 3.499 | 4.785 |
| 8 | 0.706 | 1.397 | 1.860 | 2.306 | 2.896 | 3.355 | 4.501 |
| 9 | 0.703 | 1.383 | 1.833 | 2.262 | 2.821 | 3.250 | 4.297 |
| 10 | 0.700 | 1.372 | 1.812 | 2.228 | 2.764 | 3.169 | 4.144 |
| 11 | 0.697 | 1.363 | 1.796 | 2.201 | 2.718 | 3.106 | 4.025 |
| 12 | 0.695 | 1.356 | 1.782 | 2.179 | 2.681 | 3.055 | 3.930 |
| 13 | 0.694 | 1.350 | 1.771 | 2.160 | 2.650 | 3.012 | 3.852 |
| 14 | 0.692 | 1.345 | 1.761 | 2.145 | 2.624 | 2.977 | 3.787 |
| 15 | 0.691 | 1.341 | 1.753 | 2.131 | 2.602 | 2.947 | 3.733 |
| 16 | 0.690 | 1.337 | 1.746 | 2.120 | 2.583 | 2.921 | 3.686 |
| 17 | 0.689 | 1.333 | 1.740 | 2.110 | 2.567 | 2.898 | 3.646 |
| 18 | 0.688 | 1.330 | 1.734 | 2.101 | 2.552 | 2.878 | 3.610 |
| 19 | 0.688 | 1.328 | 1.729 | 2.093 | 2.539 | 2.861 | 3.579 |
| 20 | 0.687 | 1.325 | 1.725 | 2.086 | 2.528 | 2.845 | 3.552 |
| 21 | 0.686 | 1.323 | 1.721 | 2.080 | 2.518 | 2.831 | 3.527 |
| 22 | 0.686 | 1.321 | 1.717 | 2.074 | 2.508 | 2.819 | 3.505 |
| 23 | 0.685 | 1.319 | 1.714 | 2.069 | 2.500 | 2.807 | 3.485 |
| 24 | 0.685 | 1.318 | 1.711 | 2.064 | 2.492 | 2.797 | 3.467 |
| 25 | 0.684 | 1.316 | 1.708 | 2.060 | 2.485 | 2.787 | 3.450 |
| 26 | 0.684 | 1.315 | 1.706 | 2.056 | 2.479 | 2.779 | 3.435 |
| 27 | 0.684 | 1.314 | 1.703 | 2.052 | 2.473 | 2.771 | 3.421 |
| 28 | 0.683 | 1.313 | 1.701 | 2.048 | 2.467 | 2.763 | 3.408 |
| 29 | 0.683 | 1.311 | 1.699 | 2.045 | 2.462 | 2.756 | 3.396 |
| 30 | 0.683 | 1.310 | 1.697 | 2.042 | 2.457 | 2.750 | 3.385 |
| 40 | 0.681 | 1.303 | 1.684 | 2.021 | 2.423 | 2.704 | 3.307 |
| 60 | 0.679 | 1.296 | 1.671 | 2.000 | 2.390 | 2.660 | 3.232 |
| 120 | 0.677 | 1.289 | 1.658 | 1.980 | 2.358 | 2.617 | 3.160 |
| $\infty$ | 0.674 | 1.282 | 1.645 | 1.960 | 2.326 | 2.576 | 3.090 |

# 10 Bibliografie


Amazon.com. (2011, January 27). Amazon.com Announces Fourth Quarter Sales Up 36% tp $12.95 Billion. Seattle, USA.

Boser, B. B., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the fifth annual workshop on Computational learning theory.* New York, USA: ACM.

Buijs, A. (2000). *Kwantitatieve toepassingen in de bedrijfskunde.* Educatieve Partners Nederland.

Burges, C. J. (1998). *Data Mining and Knowledge Discovery,.* Boston: Kluwer Academic Publishers.

Buss, K. (n.d.). *Literature Review on Preprocessing for Text Mining.* STRL, De Montfort University.

Campbell, C. (2008, Feb. 5). Introduction to Support Vector Machines. University of Bristol .

Cattral, R., Oppacher, F., & Deugo, D. (2001). *Supervised and Unsupervised Data Mining with an Evolutionary Algorithm.* Ottawa, Canada: IEEE.

CBIG. (n.d.). *Business Intelliegnce Data Mining Techniques.* Chicago Business Intelligence Group.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., et al. (2000). *CRISP-DM 1.0 - Step-by-step data mining guide.* SPSS Inc.

Charniak, E. (1993). *Statistical language learning.* Cambridge, MA: The MIT Press.

Chen S.F., F. J. (1998). *An Empirical Study of Smoothin Techniques for Language Modelling.* Cambridge, Massachusetts: Harvard University.

Deloitte & Touche. (2007). *The View from the Glass House, Competing in a Transparent Marketplace.* USA: Deloitte Research.

Dörre, J., Gerstl, P., & Seiffert, R. (1999). *Text Mining: Finding Niggets in Mountains of Textual Data.* Germany: IBM.

Fayyad, U., Gregory, P.-S., & Padhraic, S. (1996). *From Data Mining to Knowledge Discovery in Databases.* American Association for Artificial Intelligence.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers.

Hasan, F. M., UzZaman, N., & Khan, M. (2008). *Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla.* Bangladesh: BRAC University, Center for Research on Bangla Language Processing.

Hearst, M. A. (1999). Untangling Text Data Mining. *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics,.* Berkeley: School of Information Management & Systems, University of California.

Hillier, f. S., & Lieberman, G. J. (2005). *Introduction to Operations Research, eight edition.* New York, USA: McGraw-Hill.

Hu, M., & Liu, B. (2004). *Mining and Summarizing Customer Reviews.* Chicago, Illinois: University of Illinois.

IDC. (2007). *The Expanding Digital Universe - A Forecast of Worldwide Information Growth Through 2010.* IDC.

Karlsson, F. (1990). Constraint grammar as a fram work for parsing running text. *Proceedings of the 13th In ternational Conference on Computational Linguistics* (pp. 168–173). Helsinki, Finland: Coling.

Konchady, M. (2006). *Text Mining Application Programming.* Thomson Learning.

Kröse, B., & Smagt, P. v. (1996). *An Introduction to Neural Networks.* The Netherlands, Amsterdam: University of Amsterdam.

Leech, G., Rayson, P., & Wilson, A. (2001). *Word Frequencies in Written and Spoken English: based on the British National Corpus.* London: Longman.

Lewis, D. (1991). Evaluating text categorization. In Proceedings of Speech and Natural Language Workshop. *Defense Advanced Research Projects Agency* (pp. 312–318). CA, USA: Morgan Kaufmann.

Lewis, D. (2001). Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring Semantic Similarity. *Coyote Papers 12*, 9-16.

Lovins, J. B. (1968, March and June). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics, vol.11*.

Mayfield, J., & McNamee, P. (2003). *Single N-gram Stemming.* Laurel, USA: The Johns Hopkins University Applied Physics Laboratory .

Nakagawa, H., & Mori, T. (2002). *A Simple but Powerful Automatic Term Extraction Method.* Tokyo, Japan: The University of Tokyo.

Ng, A. (n.d.). Lecture Notes on Support vector Machines. Stanford University.

Nielsen Company. (2009). *Nielsen Global Online Consumer Survey.* New York, USA: The Nielsen Company.

Ning, W. (2005). *Textmining and Organization in Large Corpus.* Kongens Lyngby.

Ogone. (2011, Januari 26). *PERSBERICHT - Groei Belgische e-commerce in hogere versnelling!* Retrieved from http://www.ogone.be/nl/About%20Ogone/News%20Events/Figures%202010.aspx

Oracle®. (n.d.). *Non-Negative Matrix Factorization*. Retrieved from Oracle® Data Mining Concepts: http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/algo_nmf.htm

Paynter, G. W., Cunningham, S. J., & Buchanan, G. (2000). *Scalable Browsing for Large Collections: A Case Study.* Hamilton, New Zealand: University of Waikato.

Princeton University. (n.d.). *What is WordNet?* Retrieved from WordNet: http://wordnet.princeton.edu/

Richardson, R., Smeaton, A., & Murphy, J. (1994). *Using WordNet as a Knowledge Base for Measuring Semantic Similarity between Words.* Dublin, Ireland: Dublin City University.

Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., & Jin, C. (2007). *Red Opal: Product-Feature Scoring from Reviews.* Pittsburgh: Carnegie Mellon University.

Schrenk, M. (2007). *Webbot, Spiderd & Screen Scrapers.* San Fransisco: No Starch Press.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, Vol. 34, No. 1*, 1–47.

Segaran, T. (2007). *Programming Collective Intelligence - Building Smart Web 2.0 Applications.* United States of America: O'Reilly Media.

Seymour, T., Frantsvog, D., & Kumar, S. (2011). History Of Search Engines. *International Journal of Management & Information Systems – Fourth Quarter 2011 Volume 15, Number 4*.

Sharp, M. (2001). *Text Mining.* Rutgers University, School of Communication, Information and Library Studies.

Singhal, A. (2001). *Modern Information Retrieval: A Brief Overview.* Google, Inc.

Stein, B., & Potthast, M. (2007). Putting Successor Variety Stemming to Work. *Selected Papers from the 30th Annual Conference of the German Classification Society (GfKl)* (pp. 367-374). Berlin: Springer.

Tapanainen, P., & Voutilainen, A. (1994). Tagging accurately - don't guess if you know. *Proceedings of the 4th Conference on Applied Natural Language Processing* (pp. 47–52). Stuttgart, Germany: Morgan Kauffman.

Werbos, P. (1990). Backpropagation Through t Time: What It Does and How to do It. *Proceedings of the IEEE, vol. 78, no. 10*, 1550-1560.

*Wikipedia*. (n.d.). Retrieved from MYSQL - Wikipedia: http://nl.wikipedia.org/wiki/MySQL

Wikipedia. (n.d.). *Hyperoniem*. Retrieved from Wikipedia: http://nl.wikipedia.org/wiki/Hyperoniem

Wikipedia. (n.d.). *Hyponiem*. Retrieved from Wikipedia: http://nl.wikipedia.org/wiki/Hyponiem

Wikipedia. (n.d.). *MySQL*. Retrieved from Wikipedia: http://nl.wikipedia.org/wiki/MySQL

Wikipedia. (n.d.). *Supervised learning*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Supervised_learning

Wikipedia. (n.d.). *Unsupervised learning*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Unsupervised_learning

Wikipedia. (n.d.). *WordNet*. Retrieved from Wikipedia: http://nl.wikipedia.org/wiki/WordNet

Witten, I. H. (2005). *Text mining.* Hamilton, New Zealand: Computer Science, University of Waikato.

WWH. (n.d.). *Probabilistic Ranking of Product Features from Customer Reviews (Pattern Recognition and Image Analysis)*. Retrieved from what-when-how: http://what-when-how.com/pattern-recognition-and-image-analysis/probabilistic-ranking-of-product-features-from-customer-reviews-pattern-recognition-and-image-analysis/

Zaïane, O. R. (1999). *Principles of Knowledge Discovery in Databases.* Alberta: University of Alberta, Department of Computing Science.

Zipf, G. (1949). *Human Behaviour and the Principle of Least Effort.* Oxford, England: Addison-Wesley Press.

# Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
**Nieuwe kennis halen uit productbeoordeling websites aan de hand van datamining**

Richting: **master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica-informatie- en communicatietechnologie**
Jaar: **2011**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Campsteyn, Merijn**