Hochschule Wismar
Fakultät für Wirtschaftswissenschaften

Data Mining zur Identifikation potentieller Kunden

Masterarbeit zur Erlangung des Grades
Master of Science (M.Sc.)
der Hochschule Wismar

eingereicht von: Christin Lebing,
geboren am 12. Januar 1984 in Greifswald.
Studiengang: Fernstudiengang Wirtschaftsinformatik
Matrikel Nr. : 113027

Erstgutachter: Prof. Dr. Jürgen Cleve
Zweitgutachter: Dr. Matthias Berth

Wismar, den 15. Januar 2011

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ARFF	Attribute-Relation File Format
ASCII	American Standard Code for Information Interchange
CRISP-DM	CRoss Industry Standard Process for Data Mining
CRISP-DM	CRoss Industry Standard Process for Data Mining
DF	Document Frequency
DOI	Document Object Identifier
FN	False Negatives
FP	False Positives
GE	General Electrics
GmbH	Gesellschaft mit beschränkter Haftung
GML	Graph Modelling Language
GO	GeneOntology
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IfM	Institut fur Mittelstandsforschung
ISSN	International Standard Serial Number
KDD	Knowledge Discovery in Databases
KIE	Kennedy Institute of Ethics
KMU	Kleine und Mittlere Unternehmen
kNN	k Nearest Neighbors
MeSH	Medical Subject Headings
MTV	Model-Template-View
MVC	Model-View-Controller
NASA	National Aeronautics and Space Administration
NCBI	National Center for Biotechnology Information
NCR	National Cash Register
NIH	National Institutes of Health
NLM	National Library of Medicine
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
OCR	Optical Character Recognition
OHRA	Onderlinge ziektekostenverzekeringsfonds van Hoogere Rijks Ambtenaren
PII	Publisher Item Identifier
PMID	PubMed unique IDentifier
PoS	Part-of-Speech
SMO	Sequential Minimal Optimization

SPSS	Statistical Package for the Social Sciences
SQL	Standard Query Language
SVM	Support Vector Machines
TF*IDF	Term Frequency * Inverse Document Frequency
TIB	TechnischeInformationsBibliothek
TN	True Negatives
TP	True Positives
URL	Uniform Resource Locator
UTF-8	Unicode Transformation Format-8bit
XML	eXtensible Markup Language

1. Einleitung

„Daten allein sind keine Garantie für Erfolg, es kommt darauf an, was man daraus macht.“ [Leh09, S. 12]

In staatlichen Behörden, medizinischen Einrichtungen, in Banken und sogar im Supermarkt um die Ecke werden Daten gesammelt und gespeichert. Riesige Datenmengen entstehen, die für den Menschen nicht oder nur unter Nutzung von Werkzeugen überschaubar sind. Gegenwärtig wächst die Datenmenge weltweit um mehr als 60 Prozent pro Jahr. Dies entspricht einer Vervielfachung der heute (Stand: Mai 2010) vorhandenen Daten um einen Faktor von 44 bis zum Jahr 2020 [Vgl. Gmb10a, S. 1].
Wie viele nützliche Informationen sind in diesen Daten enthalten? Welche bisher unbekannten Muster können entdeckt werden? Das Data Mining versucht, genau diese Fragen zu beantworten. Zahlreiche Verfahren in diversen Anwendungsklassen stehen zur Verfügung. Sie werden unter anderem zur Analyse von Daten aus betriebswirtschaftlichen Transaktionen eingesetzt. Unternehmen erhoffen sich beispielsweise, daraus neue Erkenntnisse über das Kaufverhalten von Kunden zu gewinnen [Vgl. WF05, S. 26ff]. Welche Artikel werden zusammen gekauft? Welche Relationen bestehen zwischen Tageszeit, Standort und Art der verkauften Waren? Können Kunden sinnvoll in Gruppen eingeteilt werden? Welche Kunden sind besonders wichtig? Marketingstrategien können durch dieses Wissen optimiert und damit letztlich der Gewinn gesteigert werden.
Ziel der vorliegenden Arbeit war zu zeigen, wie Verfahren des Data Mining zur Identifikation potentieller Kunden eingesetzt werden können. Exemplarisch wurden Klassifikationsverfahren verwendet, um Publikationen aus der öffentlich-zugänglichen PubMed-Datenbank zu ermitteln, deren Autoren potentielle Kunden – zunächst für die DECODON GmbH – sind.
Die Ergebnisse der Klassifikation waren in einer Webanwendung darzustellen. Vertriebsmitarbeiter sollten in der Anwendung schnell einen Überblick über die für sie relevanten Publikationen sowie deren Autoren erhalten. Die Applikation sollte lose an die Adressdatenbank von DECODON gekoppelt werden. Einerseits war dies erwünscht, um den Nutzen für den Vertrieb erhöhen – die einfache Überführung in ein im Tagesgeschäft eingesetztes System ist möglich. Andererseits sollte die Anwendung mit geringem Aufwand für die Nutzung in anderen Unternehmen nutzbar gemacht werden können.
Die erläuterte Zielstellung erfordert die Kenntnis von Methoden und Konzepten aus Betriebswirtschaftslehre und Informatik. Um zu verstehen, weshalb die Sicherung des Unternehmenserfolgs die stetige Identifikation neuer potentieller Kunden erfordert, sind Kenntnisse der Betriebswirtschaftslehre erforderlich. Die Informatik liefert die Konzepte und Methoden für die Entwicklung der Webanwendung. Eine Verschmelzung der Erkenntnisse aus beiden Disziplinen wird in dem Fachgebiet der Wirtschaftsinformatik behandelt. Die vorliegende Arbeit ist folglich diesem Fachgebiet zuzuordnen.

Der **einleitende Teil (Kapitel 1 und 2)** dieser Arbeit enthält im Anschluss an das aktuelle Kapitel eine Einführung in das Data Mining als Disziplin für die Wissensextraktion aus großen Datenmengen (Kapitel 2).
Der Aufbau des **Hauptteils (Kapitel 3 bis 8)** orientiert sich an dem in [Cha+00] vorgeschlagenen Cross Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM ist ein Referenzmodell für die Durchführung und Dokumentation von Data-Mining-Projekten. Die erste Phase des Modells dient dem Verständnis der Aufgabenstellung aus betriebswirtschaftlicher Sicht. Dementsprechend werden die betriebswirtschaftlichen Grundlagen in Kapitel 3 erläutert. Anschließend werden – gemäß zweiter CRISP-DM-Phase – die zu analysierenden Daten in Kapitel 4 untersucht. Das daraus resultierende tiefere Verständnis der Daten ermöglicht im folgenden Schritt – der Datenvorverarbeitung – die Auswahl der richtigen Vorverabeitungsmethoden. Nach erfolgreicher Vorverarbeitung (Vgl. Kapitel 5) können die Daten in der vierten Phase des CRISP-DM zur Generierung von Modellen verwendet werden. Dieser Teil des Prozesses wird als Modellbildung bezeichnet und in Kapitel 6 erläutert. Die entdeckten Muster und Modelle müssen hinsichtlich ihrer Gültigkeit, Neuartigkeit, Nützlichkeit und Verständlichkeit bezüglich der Zielsetzung überprüft werden. Diese Evaluierung sowie die Analyse des durchgeführten Prozesses ist Inhalt der fünften CRISP-DM-Phase – Kapitel 7 dieser Arbeit. Im letzten Kapitel des Hauptteils dieser Arbeit wird – entsprechend der letzten Phase des Referenzmodell – gezeigt, wie die entdeckten Muster in einer Webanwendung nutzbar gemacht wurden (Kapitel 8).
Im **Schlussteil (Kapitel 9 und 10)** werden zunächst einige mit dieser Arbeit verwandte Arbeiten vorgestellt (Kapitel 9). Anschließend fasst das zehnte Kapitel die gewonnenen Erkenntnisse zusammen und zeigt mögliche Fortführungen.
Der Aufbau dieser Arbeit ist in Abbildung 1.1 zusammenfassend dargestellt.

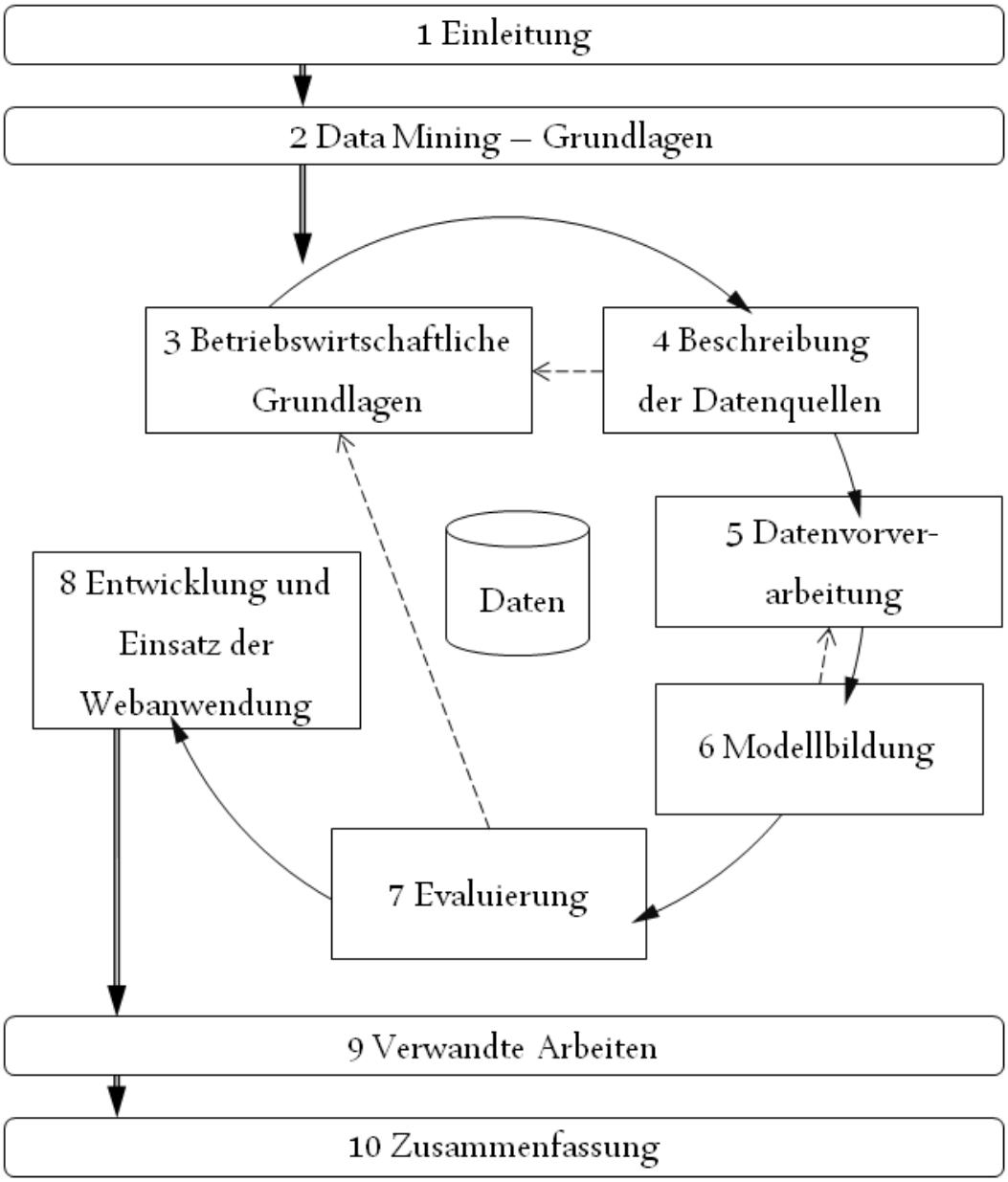

Abbildung 1.1. Aufbau der Arbeit, orientiert am CRISP-DM Referenzmodell und erweitert um einleitende und zusammenfassende Kapitel [In Anlehnung an Cha+00, S. 13].

2. Data Mining – Grundlagen

Der englische Begriff „mining“ wird mit „schürfen“, „graben“ oder „Bergbau“ übersetzt. „Data Mining“ ist demnach das Schürfen in Daten. Ähnlich wie im Bergbau ist beim Data Mining im Vorfeld nicht bekannt, was gefunden werden wird. Der Anwender steht vor einem „Berg von Daten“, in dem er „die Edelsteine der Informationsgesellschaft“ vermutet: Informationen und Wissen. Ohne Werkzeuge und technische Hilfsmittel können weder im Bergbau noch im Data Mining die Objekte der Begierde entdeckt werden.
Ziel des Data Mining ist folglich die *automatisierte Entdeckung von bisher unbekannten Zusammenhängen in riesigen Datenmengen, die die Generierung neuen Wissens ermöglichen* [Vgl. Her07, S. 456].
In diesem Kapitel wird zunächst die Verwendung des Begriffs „Data Mining“ im Zusammenhang mit dem Gesamtprozess der Wissensextraktion und bezüglich der zu analysierenden Daten erörtert. Anschließend erfolgt in Abschnitt 2.2 eine Betrachtung der Interdisziplinarität des Data Mining. Referenzmodelle für die Durchführung von Data-Mining-Prozessen werden in 2.3 beschrieben. Abschnitt 2.4 befasst sich mit den Anwendungsklassen des Data Mining. In Hinblick auf die Zielstellung dieser Arbeit werden in den letzten beiden Abschnitten Strategien für das Training und Bewertungsmaße für die Verfahren der Anwendungsklasse „Klassifikation“ erläutert.

2.1. Begriffsabgrenzung

Der Gesamtprozess (Vgl. Abschnitt 2.3) zur Extraktion von Wissen aus großen Datenmengen kann vereinfacht - wie in Abbildung 2.1 dargestellt - in die Phasen Datenvorverarbeitung, Modellbildung und Nachbereitung gegliedert werden.

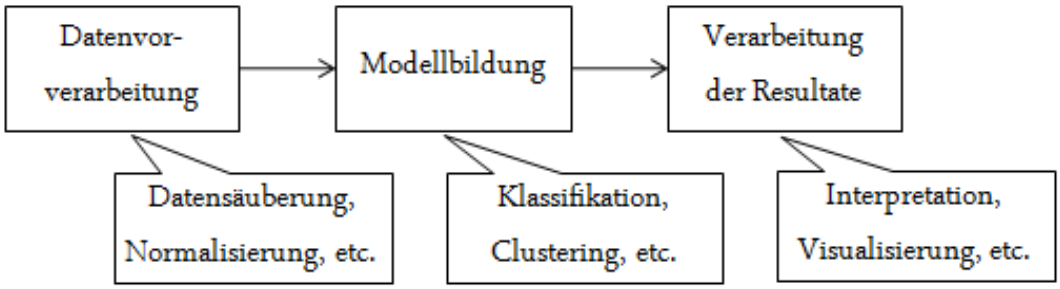

Abbildung 2.1. Vereinfachte Darstellung eines Data-Mining-Prozesses [In Anlehnung an TSK06, S. 3]

Fayyad et al. nennen in ihrer Arbeit über das „Knowledge Discovery in Databases (KDD)“-

Prozessmodell für Data Mining (Vgl. Abschnitt 2.3) die modellbildende Phase „Data Mining“ und den Prozess selbst „Knowledge Discovery in Databases“ [Vgl. FPSS96]. Chapman et al. – die Entwickler des CRISP-DM-Prozessmodells (Vgl. Abschnitt 2.3) – bezeichnen hingegen die modellbildende Phase als solche und verwenden den Begriff „Data Mining“ für den gesamten Prozess [Vgl. Cha+00].
Die Extraktion von Wissen ist nur bei erfolgreicher Durchführung aller Phasen des Prozesses möglich. Die Anwendung der modellbildenden Verfahren ohne Datenvorverarbeitung wäre nicht realisierbar. Letztere ist nicht nur der zeitintensivste Teil des Gesamtprozesses sondern auch besonders erfolgsentscheidend. Daher wird ihr mit Kapitel 5 in dieser Arbeit ein eigenes Kapitel gewidmet. Die Nachbearbeitung ist für die Verwertung der Ergebnisse der Modellbildung unerlässlich. Im weiteren Verlauf dieser Arbeit werden alle Teilschritte des Gesamtprozesses als unverzichtbare Bestandteile des Data Mining verstanden und der Begriff entsprechend verwendet.
Unabhängig von der Begriffsverwendung im Zusammenhang mit dem Gesamtprozess wird zusätzlich zwischen **Data Mining im engeren und weiteren Sinne** unterschieden. Data Mining im engeren Sinne bezeichnet die Analyse von in Datenbanken strukturiert abgelegten Daten. Data Mining im weiteren Sinne umfasst überdies sowohl die Wissensextraktion aus unstrukturierten Daten (Text Mining) als auch aus semi-strukturierten Daten aus dem Internet (Web Mining) [Vgl. Cle10, S. 12].
Die Bereiche unterscheiden sich zudem hinsichtlich der Explizität des Wissens: Die Wissensextraktion aus riesigen Datenbanken ist für den Menschen ohne Werkzeuge schwierig bis unmöglich. Das Wissen ist lediglich implizit in den Daten enthalten. Bisher unbekannte Informationen und Zusammenhänge müssen noch entdeckt werden. Menschen sind in der Lage, diese Informationen und sogar Wissen aus Texten zu extrahieren. Automatisierung ist dennoch notwendig: Aufgrund der Vielzahl von Texten und des daraus resultierenden immensen Zeitaufwandes ist vollständige manuelle Bearbeitung nicht möglich [Vgl. WF05, S. 352].
Abbildung 2.2 zeigt die Unterscheidung der Teilbereiche des Data Mining im weiteren Sinne nach der Strukturiertheit der Daten und der Explizität des Wissen. Die Größe der Boxen

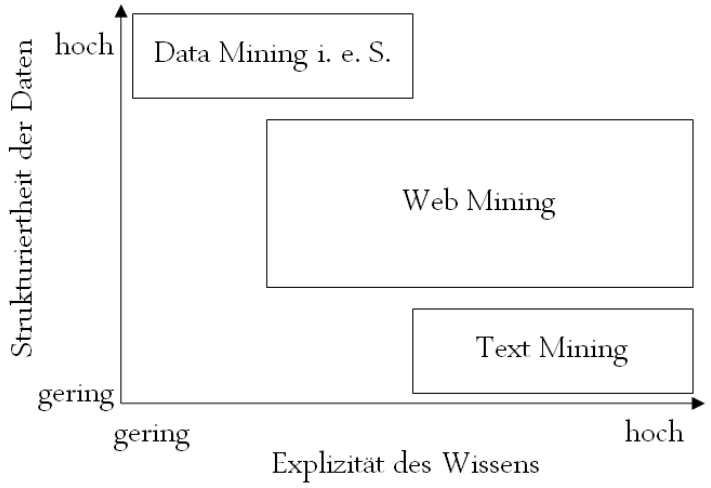

Abbildung 2.2. Anwendungsbereiche des Data Mining im weiteren Sinne (Eigene Darstellung).

resultiert aus der Ausdehnung bezüglich der Achsen: Text Mining befasst sich mit nahezu vollständig unstrukturierten Daten. Lediglich die Untergliederung in Titel und Haupttext kann in der Regel festgestellt werden. Das in Texten enthaltene Wissen ist für den menschlichen Experten leicht zu extrahieren. Folglich ist die Box „Text Mining“ relativ klein. Web Mining befasst sich mit unterschiedlich stark strukturierten Daten: In der Hypertext Markup Language (HTML) verfasste Webseiten, Weblogs, Log-Dateien und viele weitere Dokumente werden analysiert. Die Extraktion von Informationen aus HTML-Seiten ist für den Menschen relativ einfach. Das Wissen in Log-Dateien ist nur implizit vorhanden. Folglich ist die Ausdehnung der Box des „Web Mining“ größer.
Das Data Mining im weiteren Sinne wird in Zukunft durch weitere Kategorien erweitert werden. Beispielsweise setzt sich „Speech Mining“ als Bezeichnung für die Wissensextraktion aus Audiodateien (z. B. aufgezeichnete Reden von Politikern, Nachrichtensendungen usw.) allmählich durch [Vgl. Cam+07].
Ist im weiteren Verlauf dieser Arbeit von „Data Mining“ die Rede, ist stets das Data Mining im weiteren Sinne gemeint. Beziehen sich Aussagen lediglich auf die Wissensextraktion aus strukturierten Daten, wird explizit vom Data Mining im engeren Sinne gesprochen.

2.2. Bezüge zu anderen Disziplinen

Das Data Mining ist aus einer Vielzahl von Disziplinen entstanden. Aus der Informationstheorie werden beispielsweise Erkenntnisse über die Entropie verwendet: Aizawa nutzt diese zur Weiterentwicklung der in Abschnitt 5.3 erläuterten Verfahren zur Dimensionsreduktion [Aiz00]. Bewertungsmaße aus dem Information Retrieval (Vgl. Abschnitt 2.6.2) finden ebenso Anwendung [TSK06, S. 297]. Konzepte aus dem Gebiet der Datenbanken bilden die Grundlage für das Data Mining im engeren Sinne [Cle10, S. 13]. Sie können unterstützend bei der Analyse semi- und unstrukturierter Daten eingesetzt werden [Vgl. HNP05, S. 22]. Ebenso finden die Verfahren und Konzepte des Data Mining im Zusammenhang mit Data Warehouses vielfach Anwendung [Vgl. Her07, S. 476ff]. Das Text Mining nutzt die Erkenntnisse aus den Sprachwissenschaften [Vgl. FS07, S. 57ff]. Grundkenntnisse über die im Internet verwendeten Auszeichnungssprachen wie HTML sind für das Web Mining unerlässlich.
Darüber hinaus wurden Verfahren aus anderen Disziplinen für die Zielsetzungen des Data Mining erweitert: Platt stellt beispielsweise in [Pla99] einen verbesserten Algorithmus für die Lösung des bei der Modellbildung für Support Vector Machines entstehenden Optimierungsproblems vor (Vgl. Abschnitt 6.1.3).
Besonders eng sind die Bezüge des Data Mining zur Statistik und zum maschinellen Lernen [TSK06, S. 6], [FPSS96, S. 39]. Die Disziplinen sind derart eng miteinander verwoben, dass einige Verfahren nicht eindeutig einer zugeordnet werden können. Support Vector Machines werden beispielsweise dem maschinellen Lernen [Vgl. Alp10, S. 309ff], der Statistik [Vgl. Ber08, S. 301ff] und dem Data Mining [Vgl. TSK06, S. 256ff] zugeschrieben. In den folgenden beiden Abschnitten werden die wichtigsten Aspekte der Verbindungen zu Statistik und maschinellem Lernen erläutert.

2.2.1. Statistik

Bei der Erhebung von Daten für statistische Auswertungen muss häufig auf eine Vollerhebung[1] verzichtet werden. Gründe hierfür sind unter anderem

- der immense Zeitaufwand,
- die sehr hohen Kosten,
- der erforderliche Personaleinsatz,
- die Nicht-Durchführbarkeit aufgrund technischer, rechtlicher oder sonstiger Restriktionen sowie
- der proportional zur Datenmenge wachsende Aufwand für die Überpüfung und Korrektur der Daten

[Vgl. Voß+04, S. 48]. Die Beschränkung auf Teilerhebungen ist unerlässlich. Eine erhobene Teilmenge muss groß genug sein, um Rückschlüsse auf die Grundgesamtheit ziehen zu können – sie muss repräsentativ sein. Die Stichprobentheorie[2] – ein äußerst umfangreiches Teilgebiet der Statistik – dient der Bestimmung des erforderlichen Stichprobenumfangs in Abhängigkeit von der Zielstellung. Die Erkenntnisse daraus werden im Data Mining verwendet.
Die Stichprobentheorie ist nur ein Teilgebiet der Statistik, das im Data Mining Anwendung findet. Darüber hinaus werden beispielsweise Theorien zum Schätzen und Testen[3] genutzt [Vgl. TSK06, S. 6].
Auch zur Abschätzung der Erfolgsaussichten [Vgl. Cle10, S. 14] und für den Vergleich der Performanz von Methoden des Data Mining [Vgl. WF05, S. 153ff] werden Verfahren der Statistik eingesetzt.

2.2.2. Maschinelles Lernen

Als maschinelles Lernen wird die Fähigkeit einer Maschine verstanden, durch Sammeln von Erfahrungen sukzessive neues Wissen aufzubauen, um gestellte Aufgaben besser lösen zu können [Vgl. Mit97, S. 2]. Dieses Teilgebiet der Künstlichen Intelligenz ist bereits sehr ausgereift: Nach erfolgreicher Lernphase ist die Leistung für einige Problemstellungen mit der menschlicher Experten vergleichbar. Die Geschwindigkeit ist sogar deutlich höher [Vgl. Seb01, S. 2]. Forschungsgegenstand des maschinellen Lernens ist die (Weiter-) Entwicklung von Lernstrategien für Maschinen.
Eine erfolgreiche Lernstrategie ist das **überwachte Lernen (Supervised Learning)**. Bei dieser Variante werden in der Trainingsphase Eingabedaten mit den für diese Daten korrekten Ausgabedaten gegeben. Auf Grundlage dieser Trainingsdaten kann ein Modell aufgebaut werden, dass auf neue Daten angewandt werden kann. Überwachtes Lernen wird beispielsweise bei der Klassifikation (Vgl. Abschnitt 2.4.5) eingesetzt [Vgl. JC08, S. 204].

[1] Bei dieser Form der Datenerhebung findet keine Beschränkung statt. Die Grundgesamtheit wird vollständig berücksichtigt [Vgl. Voß+04, S. 48].
[2] Eine Einführung gibt z. B. [Boc98]
[3] Einen guten Einstieg in die Schätz- und Testtheorie gibt z. B. [Rü99]

Beim **unüberwachten Lernen (Unsupervised Learning)** werden die Ausgangsdaten ohne weitere Informationen – z. B. über die Klassenzugehörigkeit der Objekte – zur Verfügung gestellt. Derart lernende Verfahren werden eingesetzt, um bisher unbekannte Zusammenhänge und Strukturen zu entdecken. Das in Abschnitt 2.4.1 erläuterte Clustering nutzt beispielsweise diese Strategie [Vgl. Alp10, S. 11ff].
Die dritte Lernstrategie ist das **bestärkende Lernen (Reinforcement Learning)**. Jede Aktion des Lerners – als Agent bezeichnet – überführt dessen Umwelt in einen neuen Status. Der Agent erhält für die Aktion eine positive oder eine negative Belohnung. Ziel des Agenten ist die Optimierung seines Verhalten, so dass er weniger negative und mehr positive Belohnungen erhält [Vgl. KLM96]. Abbildung 2.3 zeigt eine abstrahierte Darstellung dieser Lernstrategie. Im Data Mining wird diese Strategie - soweit bekannt - nicht eingesetzt. Aufgrund der rasanten

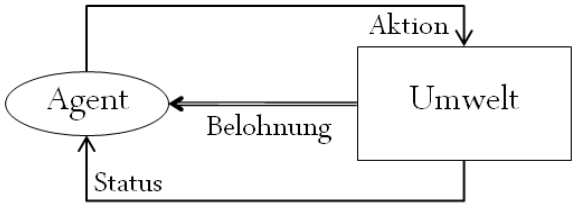

Abbildung 2.3. Ablauf des bestärkenden Lernens [In Anlehnung an Alp10, S. 448].

Weiterentwicklung von Data Mining und Maschinellen Lernen [Vgl. Alp10, S. xxxv] ist die Anwendung aber denkbar.

2.3. Prozessmodelle

Für die erfolgreiche Durchführung von Data Mining ist die Planung des Prozesses entscheidend. Ein weit verbreitetes Referenzmodell ist Knowledge Discovery in Databases (KDD). KDD wurde von Fayyad et al. entwickelt. Entgegen der Bezeichung ist die Anwendung dieses Referenzmodells nicht auf Data Mining im engeren Sinne beschränkt. Das Referenzmodell eignet sich auch für das Text Mining [Vgl. HNP05, S. 20ff].
Der Data-Mining-Prozess wird in fünf Phasen gegliedert [Vgl. FPSS96]:

1. **Selektion der Daten**

   Aus den verfügbaren Daten werden die zu analysierenden Zieldaten selektiert (Vgl. Abschnitt 5.1). Technische und rechtliche Restriktionen müssen berücksichtigt werden.

2. **Bereinigung der Daten**

   Diese Phase dient der Behandlung von fehlenden oder fehlerhaften Werten. Ziel sind möglichst fehlerfreie und damit besser verwendbare Daten. In Abschnitt 5.2 werden für die Aufgabenstellung relevante Aspekte dieser Phase erläutert.

3. **Datentransformation**

Für die Analyse der Daten ist häufig z. B. die Anpassung der Datentypen und -strukturen erforderlich. Oftmals müssen Daten aggregiert oder Attribute kombiniert / separiert werden [Vgl. Cle10, S. 21]. Die Datentransformation ist Gegenstand von Abschnitt 5.4.

4. **Data Mining**

   In dieser Phase werden Verfahren (Vgl. Abschnitt 2.4) ausgewählt und angewendet. Sind Menge oder Qualität der Daten nicht ausreichend, ist eine Wiederholung der vorherigen Phasen erforderlich.

5. **Interpretation und Evaluierung**

   Die in der vierten Phase (Vgl. Abschnitt 6.2 und Kapitel 7) gewonnenen Muster und Modelle müssen interpretiert und evaluiert werden. Sind die Erkenntnisse nicht ausreichend interessant, neu, einzigartig oder nützlich, müssen die vorherigen Phasen analysiert und ggf. wiederholt werden.

Der gesamte KDD-Prozess ist in Abbildung 2.4 dargestellt.

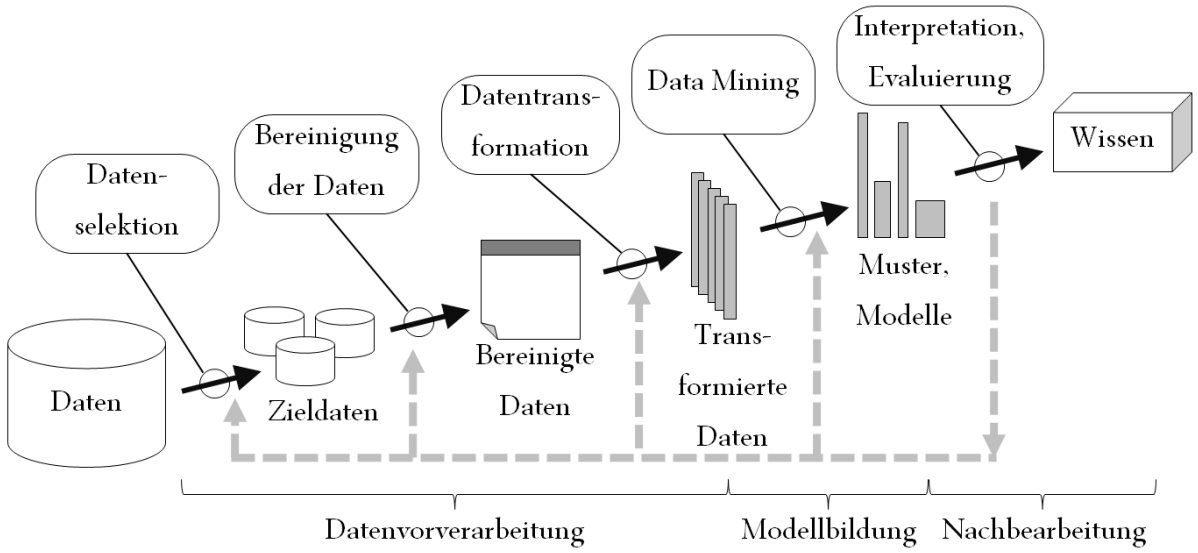

Abbildung 2.4. Knowledge Discovery in Databases - Referenzmodell für den Data-Mining-Prozess nach Fayyad et al. [In Anlehnung an: FPSS96, S. 41].

Ein weiteres Referenzmodell für den Data-Mining-Prozess ist der Cross Industry Standard Process for Data Mining (CRISP-DM). Es wurde von den Vertretern der Unternehmen SPSS, NCR, Daimler-Benz und OHRA entwickelt.
Werden die ersten drei Phasen des KDD-Prozesses unter dem Oberbegriff Datenvorverarbeitung zusammengefasst, die Data-Mining-Phase in Modellbildung umbenannt sowie Evaluierung und Interpretation als Nachbearbeitung bezeichnet, entspricht das KDD-Modell den Schritten drei, vier und fünf des CRISP-DM. Zusätzlich werden im CRISP-DM-Referenzmodell zwei Teilschritte zu Beginn und eine zum Ende des Prozesses eingeführt.
Zu Beginn werden die folgende Schritte ergänzt:

1. **Verständnis der Aufgabe**

   In diesem ersten Schritt werden die betriebswirtschaftlichen Grundlagen und Ziele erörtert (Vgl. Kapitel 3). Die daraus resultierende Zielstellung für das Data Mining wird konkretisiert.

2. **Verständnis der Daten**

   Die Sammlung der Ausgangsdaten wird ebenso beschrieben wie die Daten selbst. Ziel ist ein möglichst umfangreiches Verständnis der Daten und ihrer Attribute sowie die Analyse der Datenqualität (Vgl. Kapitel 4).

Nach der Nachbearbeitung – im CRISP-DM Evaluierung genannt - folgt im CRISP-DM-Referenzmodell das **Deployment**. In dieser Phase werden

- Pläne für die Nutzung der Ergebnisse,
- Pläne für die (Weiter-)Entwicklung und Wartung und
- ein Abschlussbericht

erstellt (Vgl. Kapitel 8). Abbildung 2.5 zeigt das vollständige Referenzmodell.

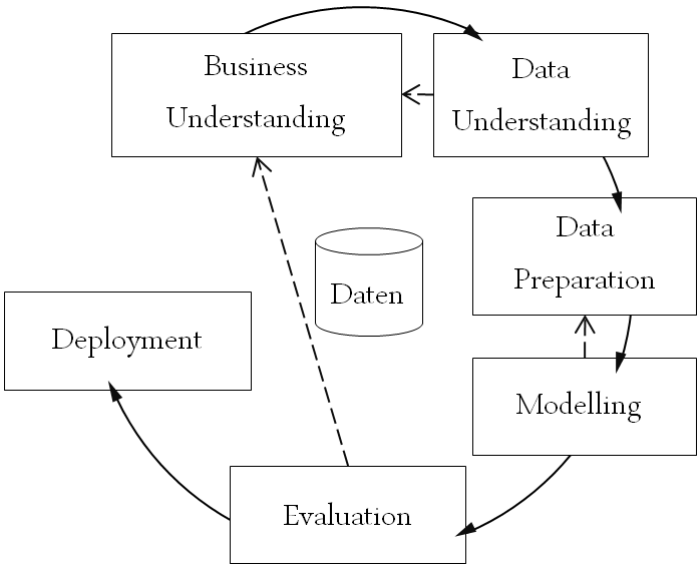

Abbildung 2.5. Cross Industry Standard Process for Data Mining (CRISP-DM) [Quelle: Cha+00, S. 13].

2.4. Anwendungsklassen

Die Anwendungsklassen des Data Mining werden in beschreibende und vorhersagende Verfahren untergliedert. Beschreibende Verfahren liefern neue Informationen und bisher unbekanntes Wissen über Ausreißer, Trends und Strukturen in den Eingabedaten. Vorhersagende

Verfahren erhalten Trainingsdaten, aus denen sie Modelle entwickeln. Diese Modelle werden genutzt, um Vorhersagen für neu präsentierte Daten zu machen [Vgl. TSK06, S. 7].
In den folgenden Abschnitten werden zunächst drei Anwendungsklassen der beschreibenden Verfahren vorgestellt: Clustering, Assoziationsanalyse und Anomaliedetektion. Anschließend werden in den Abschnitten 2.4.4 und 2.4.5 die vorhersagenden Anwendungsklassen Regression und Klassifikation betrachtet.

2.4.1. Clustering

Clustering-Verfahren werden zur Entdeckung von Strukturen in großen Datenmengen verwendet. Die Methoden lernen unüberwacht. Distanzmaße werden zur Bestimmung der Ähnlichkeit der Objekte bzw. Datensätze verwendet [Vgl. Cle10, S. 15].
Beim **partitionierendem Clustering** werden die Daten derart gruppiert, dass jedes Objekt bzw. jeder Datensatz genau einem Cluster angehört. Die Cluster sind somit eindeutig voneinander verschieden. Abbildung 2.6 zeigt eine vereinfachte Darstellung des partitionierenden Clusterings.

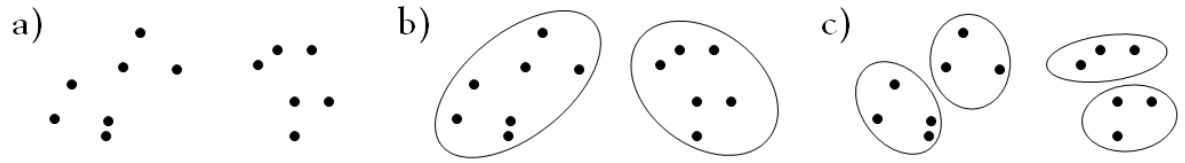

Abbildung 2.6. Partitionierendes Clustering. Darstellt sind die a) ursprünglichen Datenobjekte sowie zwei Beispiele, b) und c), für partitionierendes Clustering (In Anlehnung an: [Her07, S. 459]).

Bekannte Verfahren des partitionierenden Clustering sind das k-means- bzw. k-medians-Verfahren oder selbstorganisierende Neuronale Netze [Vgl. Cle10, S. 53].
Für das **hierarchische Clustering** werden zwei Ansätze unterschieden:

Top-Down-Verfahren (Divisives Clustering) Bei diesen Verfahren sind zunächst alle Objekte in einem Cluster zusammengefasst. Dieses Cluster wird iterativ geteilt bis in jedem Cluster nur noch ein Objekt ist. Die einzelnen Cluster sind hierarchisch verbunden.

Bottom-Up-Verfahren (Agglomeratives Clustering) Ausgangspunkt sind Cluster, die aus je einem Objekt bestehen. Basierend auf der Ähnlichkeit der Cluster wird aus den Clustern iterativ eine hierarchische Struktur aufgebaut.

Ergebnis beider Varianten ist eine hierarchische Strukturierung der Objekte. Jedes Cluster kann über- und untergeordnete Cluster haben. Für die zusammenfassende Darstellung aller Cluster wird häufig ein Baum verwendet [Vgl. Her07, S. 460f].
Neben der Differenzierung von hierarchischem und partitionierendem Clustering wird zwischen exklusivem, überlappendem und fuzzy Clustering unterschieden. Exklusive Verfahren ordnen jedes Objekt genau einem Cluster zu (Vgl. Abbildung 2.7 a)). Bei überlappenden Methoden können Objekte mehreren Clustern angehören (vgl. Abbildung 2.7 b)). Beim **fuzzy**

Clustering (Vgl. Abbildung 2.7 c)) wird zu jedem Objekt für jedes Cluster ein Gewicht angegeben. Dieses sagt aus, wie sicher das Objekt zum Cluster gehört. In der Praxis ist das Ergebnis dieser Form des Clustering nicht von den Resultaten exklusiver Verfahren zu unterscheiden: Das Objekt wird dem Cluster zugeordnet, für das das höchste Gewicht ermittelt wurde [Vgl. TSK06, S. 492f].

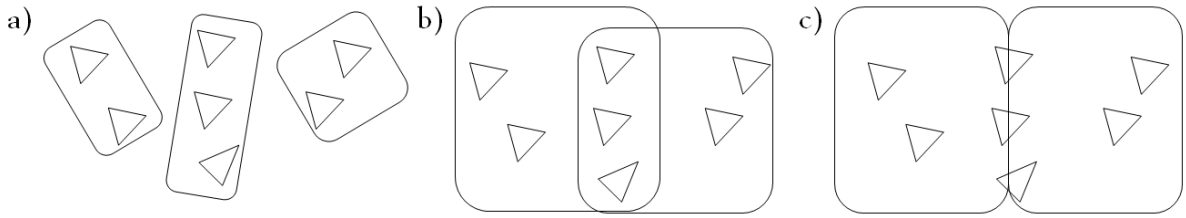

Abbildung 2.7. Varianten des Clustering: a) Exklusives, b) überlappendes und c) fuzzy Clustering (Eigene Darstellung).

Bisher wurde implizit angenommen, dass alle Objekte Clustern zugeordnet werden (= vollständiges Clustering). Werden beispielsweise Ausreißer ignoriert, ist vom partiellen Clustering die Rede.

2.4.2. Assoziationsanalyse

Die Assoziationsanalyse ist aus der Warenkorbanalyse entstanden. Die Warenkorbanalyse dient dem Entdecken von Beziehungen in Transaktionsdaten aus Supermärkten. Aus den Beziehungen werden Regeln für die Vorhersage des Kaufverhaltens generiert [Vgl. TSK06, S. 328]. Regeln haben eine Bedingung – mit „WENN“ eingeleitet – sowie eine daraus resultierende Folge, die mit „DANN“ beginnt. Ein Regel könnte lauten:

„WENN A und B gekauft werden, DANN wird auch C gekauft.“

Das Lernen findet bei der Assoziationsanalyse unüberwacht statt, d. h. das Verfahren erhält als Eingabe lediglich die Transaktionen. Aus diesen werden Frequent Itemsets – häufig vorkommende Mengen von z. B. Artikeln – ermittelt. Für diese Ermittlung wird der Support verwendet. Er gibt den Anteil der Datensätze mit dem jeweiligen Frequent Itemset an allen Datensätzen an:

$$\text{Support} = \frac{\text{Anzahl der Datensätze, die Frequent Itemset X enthalten}}{\text{Anzahl aller Datensätze}}$$

Vor Beginn einer Assoziationsanalyse wird ein Schwellwert für den Support festgelegt. In einem iterativen Prozess wird dann zunächst das Vorkommen jedes einzelnen Elements gezählt. Die Elemente, deren Support oberhalb des Schwellwerts liegt, werden weiter berücksichtigt. Diese Frequent Itemsets der Länge 1 werden zu neuen Itemsets der Länge 2 zusammengefügt. Von diesen werden erneut nur diejenigen mit ausreichendem Support in die nächste Iteration mitgenommen. Das Verfahren wird so lange fortgeführt, bis keine weiteren Frequent Itemsets größerer Länge mehr gefunden werden können. Aus den Itemsets werden dann Regeln erzeugt. Für jede dieser Regeln wird deren Konfidenz wie folgt berechnet:

$$\text{Konfidenz} = \frac{\text{Anzahl der Datensätze mit allen Elementen der Regel}}{\text{Anzahl der Datensätze mit den Elementen der Bedingung}}$$

Die Konfidenz einer Regel muss gleich oder größer als ein zu Beginn festgelegter Schwellwert sein, damit die Regel in die Ergebnismenge aufgenommen wird [Vgl. Her07, S. 464ff]. Eine vereinfachte Darstellung dieser beiden Phasen der Assoziationsanalyse – Ermittlung der Frequent Itemsets und Erzeugen von Regeln – wird in Abbildung 2.8 gezeigt.

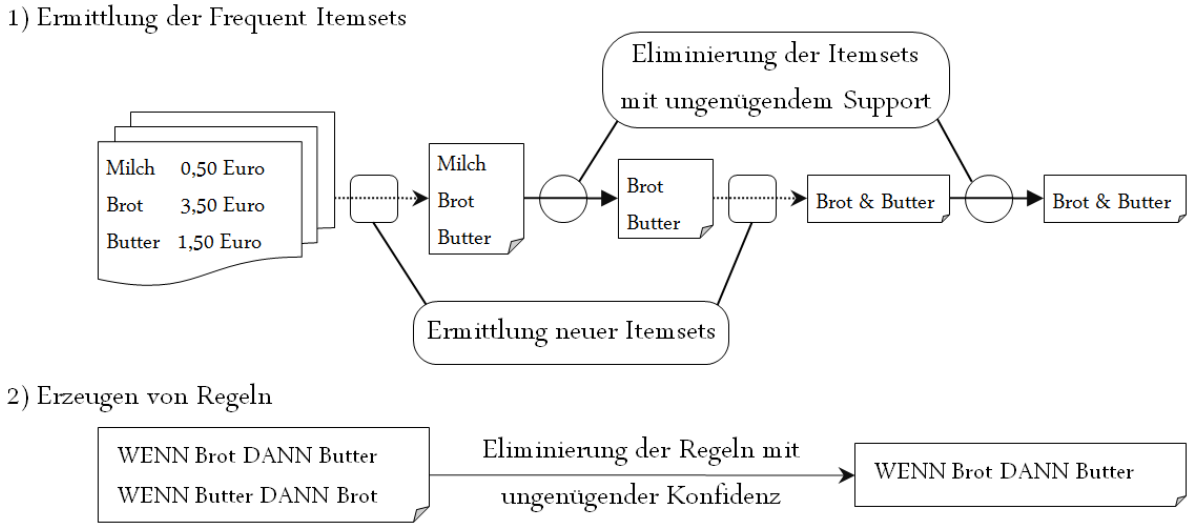

Abbildung 2.8. Vereinfachte Darstellung des Ablaufs der Assoziationsanalyse (Eigene Darstellung).

2.4.3. Detektion von Anomalien

In vielen Anwendungsfällen sind nicht die Struktur oder die Zusammenhänge zwischen den Daten zu ermitteln. Vielmehr ist von Interesse, welche Datensätze signifikant von der Mehrheit abweichen. Die Entdeckung von derartigen Ausreißern wird als Anomaliedetektion bezeichnet.
Eine geeignete Darstellungsform für Ausreißer sind die „Chernoff Faces“. Chernoff nutzt Gesichter für die Darstellung hochdimensionaler Datenmengen. Jeder Teil eines Gesichtes wird stellvertretend für ein Attribut der Datenmenge dargestellt. Größe bzw. Form des Gesichtsteils visualisieren die Attributsausprägung [Vgl. Che73]. Abbildung 2.9 zeigt beispielhaft einen Ausreißer (das fünfte Gesicht von links) in einer ansonsten sehr homogenen Menge.
Für die Detektion von Anomalien werden **grafische** (Vgl. Abbildung 2.9), **statistische** und **distanzbasierte** Ansätze verwendet [Vgl. Her07, S. 471f]. Anwendung finden die Verfahren der Anomaliedetekion beispielsweise im Bankensektor bei der Entdeckung von Kreditkartenmissbrauch oder in der medizinischen Diagnostik.

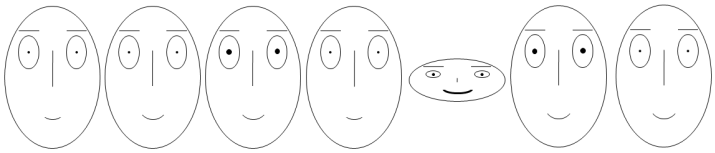

Abbildung 2.9. Datenmenge mit einem Ausreißer (fünftes Gesicht von links) – visualisiert mit Chernoff-Gesichtern (Eigene Darstellung).

2.4.4. Regression

Die Regression ist ein Verfahren der numerischen Vorhersage [Vgl. Cle10, S. 26]. Aus einer gegebenen Menge von numerischen Daten wird eine Funktion approximiert, die möglichst optimal die gegebenen Daten widerspiegelt. Abbildung 2.10 zeigt eine vereinfachte Darstellung einer Regressionsfunktion. Mit der erhaltenen Funktion können beispielsweise Trends oder Klassen für neue Objekte vorhergesagt werden.

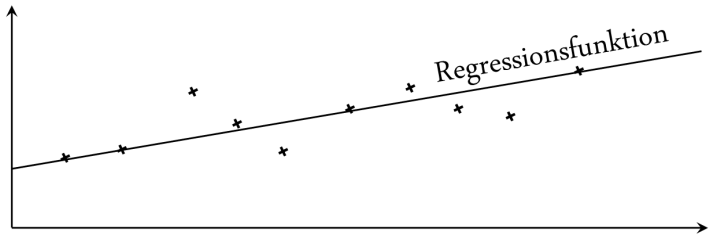

Abbildung 2.10. Regressionsfunktion (Eigene Darstellung).

Regressionsverfahren können auch für die Klassifikation eingesetzt werden. Voraussetzung ist, dass die Attribute ausschließlich numerisch sind. Für eine Instanz i mit den Attributsausprägungen a^i_1, a^i_2, ..., a^i_k kann eine Klasse $c^i_{vorhergesagt}$ unter Nutzung der Gewichte w_0, w_1, ..., w_k mit einer Regressionsfunktion $f(w)$ vorhergesagt werden:

$$c^i_{vorhergesagt} = w_0 + w_1 a^i_1 + w_2 a^i_2 + ... + w_k a^i_k = \sum_{j=0}^{k} (w_j a^i_j). \tag{2.1}$$

Klasse $c^i_{vorhergesagt}$ kann von der tatsächlichen Klasse $c^i_{tatsächlich}$ der jeweiligen Instanz abweichen. Unter der Annahme, dass die beste Regressionsfunktion die besten Vorhersagen für die Trainingsmenge liefert, wird die Summe der Abweichungen aller Klassenvorhersagen

$$\sum_{i=1}^{n} (c^i_{tatsächlich} - \sum_{j=0}^{k} (w_j a^i_j))^2 \tag{2.2}$$

durch Anpassung der Gewichte minimiert [Vgl. WF05, 119f].
Neben der linearen Regression werden beispielsweise Regressionsbäume der Regression zugeordnet [Vgl. Cle10, S. 26]. Anwendung findet das Grundprinzip der Funktionsapproximation ebenso beim statistischen Lernen [Vgl. Ber08].

2.4.5. Klassifikation

Ziel der Klassifikation ist das Erzeugen eines Modells, welches für die Vorhersage der Klassenzugehörigkeit neuer Objekte genutzt werden kann [TSK06, S. 146]. Zu diesem Zweck werden einem Klassifikationsverfahren zunächst Objekte mit ihrer korrekten Klassifizierung präsentiert. Diese initialen Daten werden als Trainingsmenge bezeichnet. Auf Basis der Trainingsmenge wird ein Modell erzeugt, dass Regeln für die Klassenzuordnung enthält. Diese erste Phase der Induktion wird gefolgt von der Deduktion. Dabei wird das erlernte Modell auf neue Objekte zur Vorhersage ihrer Klassenzugehörigkeit angewandt. Der Ablauf eines Klassifikationsverfahrens ist in Abbildung 2.11 dargestellt.

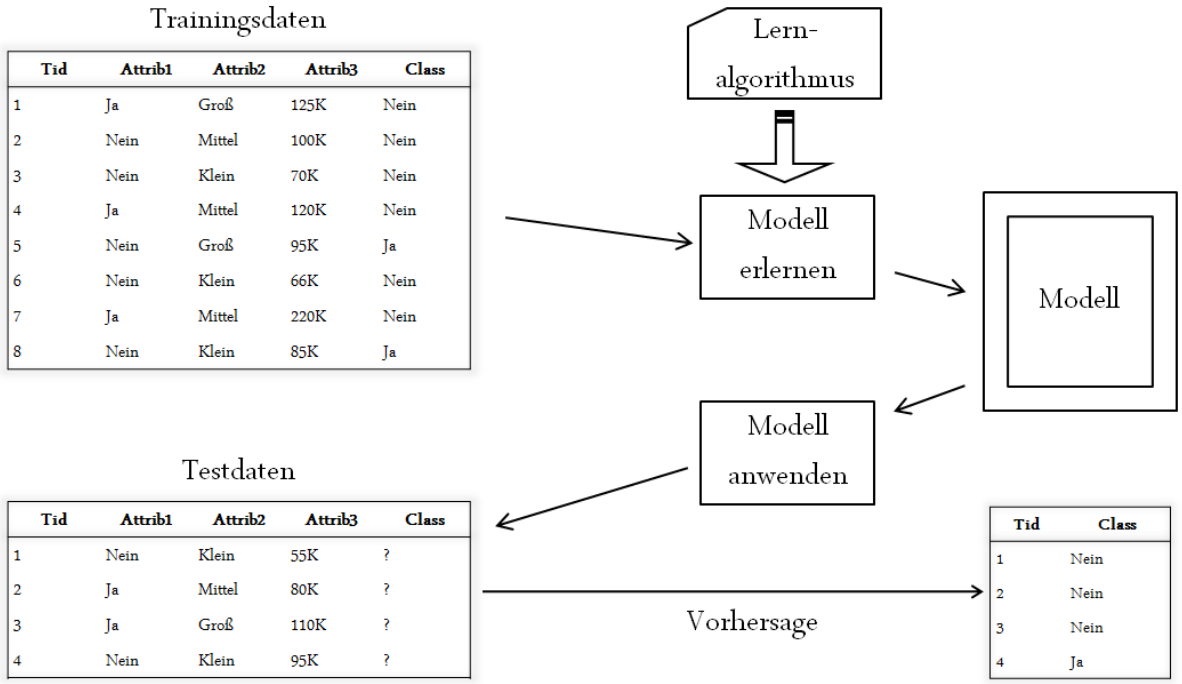

Abbildung 2.11. Vereinfachte Darstellung des Ablaufs der Klassifikation (In Anlehnung an: [TSK06, S. 148]).

In dieser Arbeit werden ausschließlich Verfahren der Klassifikation angewandt. Die verwendeten Methoden - Naïve Bayes, k-Nearest-Neighbors und Support Vector Machines – werden in Abschnitt 6.1 erläutert.

2.5. Strategien für Test und Training

Die Instanzmenge für überwacht-lernende, modellbildende Verfahren muss in eine Trainings- und eine Testmenge zerlegt werden. Für das weitere Verständnis relevante Zerlegungsstrategien werden in den folgenden Abschnitten erläutert. Weitere werden unter anderem Methoden bei [WF05], [WK91] und [Che99] beschrieben.

2.5.1. Holdout und Stratifikation

Beim **Holdout** wird die Instanzmenge in zwei Mengen geteilt: Eine Trainingsmenge, die zum Lernen präsentiert wird, und eine Testmenge, für deren Elemente Vorhersagen gemacht werden. Die Teilung erfolgt zufällig. Häufig wird ein Drittel der Instanzmenge als Testmenge zurückgehalten. Die anderen zwei Drittel werden der Trainingsmenge zugeordnet [Vgl. WF05, S. 149].
Eine Weiterentwicklung der Holdout-Strategie ist die **Stratifikation**. Bei der Trennung in Trainings- und Testmenge wird darauf geachtet, dass die Verteilung der Klassen in beiden Mengen der Verteilung in der Instanzmenge entspricht. Für alle Klassen k gilt für die Häufigkeit der Elemente h_k:

$$h_k^{\text{Instanzmenge}} = h_k^{\text{Trainingsmenge}} = h_k^{\text{Testmenge}}$$

[Vgl. Cle10, S. 60]. In der Praxis ist die exakte Gleichheit der drei Häufigkeiten nicht immer herstellbar, wird aber angestrebt.
Diese Vorgehensweise ist aufgrund zweier wesentlicher Vorteile empfehlenswert:

- *Vermeidung der Verfälschung der Ergebnisse* durch ungünstige bzw. besonders gute Verteilung der Klassen in Trainings- und Testmenge. Sind beispielsweise die Elemente einer der Klassen ausschließlich in der Testmenge vorhanden, kann das Verfahren in der Trainingsphase kein Wissen zu dieser Klasse generieren. Kommen Elemente einer Klasse nicht in der Testmenge vor, können keinerlei Aussagen über die Performanz bezüglich dieser Klasse gemacht werden.

- Entspricht die Verteilung in der Instanzmenge der (vermuteten) realen Verteilung in der Grundgesamtheit, ermöglicht die Stratifikation eine *verlässlichere Prognose bezüglich der Performanz außerhalb der Testumgebung*.

2.5.2. Kreuzvalidierung

Eine weitere Strategie ist die **Kreuzvalidierung**. Bei diesem Verfahren wird die Instanzmenge zunächst in n gleich große Teilmengen stratifiziert. Anschließend wird die erste der n Teilmengen zur Testmenge erklärt, alle anderen zur Trainingsmenge zusammengefasst. In einer zweiten Iteration wird die zweite Teilmenge zum Testen und die verbleibenden für das Training verwendet. Das Verfahren wird fortgeführt, bis jede der Mengen genau einmal Testmenge war. Das entspricht n Iterationen.
Empfohlen wird $n = 10$ (Vgl. [Sal97, S. 325], [WF05, S. 150], [FS07, S. 79]). Diese Variante wird als 10-fache Kreuzvalidierung bezeichnet.
Ist n gleich der Anzahl aller Elemente in der Instanzmenge, beinhaltet jede der n Mengen lediglich ein Element. Folglich wird in jeder Iteration genau ein Element nicht zum Training verwendet. Diese Variante der Kreuzvalidierung wird daher als **Leave-one-out**-Verfahren bezeichnet.

Tabelle 2.1 stellt diese beiden häufigsten Varianten – Leave-one-out und 10-fache Kreuzvalidierung – gegenüber.

	Leave-one-out	10-fache Kreuzvalidierung
Elemente in der Trainingsmenge	n-1	90% *
Elemente in der Testmenge	1	10% *
Wiederholungen	n	10

* der Instanzmenge

Tabelle 2.1. Varianten der Kreuzvalidierung (In Anlehnung an [Voß+04, S. 598])

Zur Evaluierung der Modellbildung (Vgl. Abschnitt 6.2) wurde – wie bei [Sal97, S. 325], [WF05, S. 150] und [FS07, S. 79] empfohlen – die zehnfache Kreuzvalidierung verwendet.

2.6. Bewertungsmaße

Nach der Anwendung von Data-Mining-Verfahren ist deren Bewertung erforderlich. Zahlreiche Maße stehen zur Verfügung. Aufgrund der Dynamik des Forschungsgebietes werden zudem stetig neue entwickelt.
Im Rahmen dieser Arbeit werden ausschließlich Klassifikationsverfahren verwendet. Folglich werden lediglich die für die Bewertung der verwendeten Verfahren relevanten Maße vorgestellt. Weitere werden unter anderem bei [TSK06], [Cle10] oder [WF05] erläutert.

2.6.1. Erfolgs- und Fehlerrate

Ein sehr einfach zu berechnendes Bewertungsmaß ist die **Erfolgsrate**. Sie gibt den Anteil der richtig klassifizierten an allen klassifizierten Dokumenten an:

$$\text{Erfolgsrate} = \frac{\text{Anzahl richtig klassifizierter Dokumente}}{\text{Anzahl aller Dokumente}}.$$

Alternativ kann der Anteil von falsch klassifizierten Dokumenten an allen Dokumenten verwendet werden. Dieser Quotient wird als **Fehlerrate** bezeichnet und wie folgt berechnet:

$$\text{Fehlerrate} = 1 - \text{Erfolgsrate} = \frac{\text{Anzahl falsch klassifizierter Dokumente}}{\text{Anzahl aller Dokumente}}.$$

Erfolgs- und Fehlerrate sind einfach interpretier- und berechenbar. Sie eignen sich für die Bewertung von Klassifizierungsaufgaben, bei denen

- die Klassenanteile annähernd gleich verteilt und

- separate Betrachtungen des Erfolges einzelner Klassen entbehrlich

sind. Gehören beispielsweise 90% aller Elemente einer Datenmenge einer ersten Klasse und 10% einer zweiten an, kann dies zu folgendem Problem führen: Eine Erfolgsrate von 0,9 lässt keine Aussage über den Erfolg bezüglich der zweiten Klasse zu. Alle Elemente der zweiten Klasse könnten sowohl falsch als auch richtig klassifiziert worden sein.
Im nächsten Abschnitt werden Maße vorgestellt, die für die Bewertung bei unausgewogener Klassenverteilung bzw. für die Erfolgsbetrachtung bezüglich einer Klasse geeignet sind.

2.6.2. Recall, Precision und F-Measure

In vielen Anwendungsfällen sollen mittels Klassifikation Datensätze bzw. Objekte einer Klasse gefunden werden. Der Erfolg bezüglich aller anderen Klassen ist irrelevant. Lediglich die Bewertung bezüglich der relevanten Klasse ist bedeutsam.
Für die Dokumentation der Ergebnisse wird unterschieden in

False Positives (FP) – nicht-relevante Objekte als relevant klassifiziert,

False Negatives (FN) – relevante Objekte als nicht-relevant klassifiziert,

True Positives (TP) – relevante Objekte als relevant klassifiziert und

True Negatives (TN) – nicht-relevante Objekte als nicht-relevant klassifiziert

(Vgl. [WF05, S. 162], [TSK06, S. 296]). Dargestellt werden diese Anzahlen in einer **Konfusionsmatrix**:

		Vorhergesagte Klasse	
		positiv	negativ
Tatsächliche	positiv	TP	FN
Klasse	negativ	FP	TN

Tabelle 2.2. Konfusionsmatrix [In Anlehnung an WF05, S. 162]

Die Werte können für die Berechnung von Bewertungsmaßen verwendet werden. Beispielsweise kann ermittelt werden, wie genau das Klassifikationsverfahren in Bezug auf die relevante Klasse war: Der Anteil tatsächlich relevanter Objekte wird mit allen als relevant klassifizierten Objekten ins Verhältnis gesetzt. Dieses Maß wird **Precision** genannt.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.3}$$

Sind z. B. 90 Prozent aller als relevant klassifizierten Objekte tatsächlich relevant, beträgt die Precision 0,9.

Weiterhin kann mit dem Bewertungsmaß **Recall** ermittelt werden, wie viele der relevanten Objekte wiederentdeckt, d. h. als relevant klassifiziert, wurden:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4}$$

Wurden beispielsweise 90 Prozent aller relevanten Objekte als solche klassifiziert, beträgt der Recall 0,9.
In der Praxis wird die Verwendung möglichst weniger Maße favorisiert. Da Recall und Precision Beziehungszahlen sind, können sie mit dem harmonischen Mittel $\bar{x}_H$ gemittelt werden:

$$\bar{x}_H = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{2.5}$$

Das harmonische Mittel von Recall und Precision wird als **F-Measure** bezeichnet [Mak+99, S. 250]. Im Gegensatz zu Recall und Precision ist das F-Measure schwer zu interpretieren. Grundsätzlich tendiert das harmonische Mittelwert stärker zum jeweils kleineren Wert [TSK06, S. 297]. D. h. der schlechtere der beiden Werte ist nicht viel kleiner als das F-Maß. Ist der Wert des F-Measures hoch, sind auch die Werte von Recall und Precision hoch.

2.6.3. Vergleich von Bewertungsmaßen

Werden – wie in der vorliegenden Arbeit – mehrere Verfahren evaluiert, ist der Vergleich der ermittelten Bewertungsmaße erforderlich. In einigen Quellen wird die Verwendung des t-Test empfohlen [Vgl. TSK06, S. 192], [Vgl. WF05, S. 154]. Der t-Test ist ein parametrischer Test, da er eine Gauß'sche Normalverteilung der Zufallsvariable sowie die Unabhängigkeit der Testmengen für die zu vergleichenden Verfahren annimmt [Vgl. Sal97, S. 321]. Ermittelt wird, ob die Nullhypothese – die arithmetischen Mittel der Zufallsvariablen zweier Stichproben sind gleich – mit einer festzulegenden Irrtumswahrscheinlichkeit[4] abgelehnt werden kann. Ist dies möglich, wird angenommen, dass die Verteilungen in der Grundgesamtheit verschieden sind.
Salzberg rät in seiner Arbeit über die Grundsätze für den Vergleich von Klassifikatoren aus zweierlei Gründen von der Anwendung des t-Tests ab:

- Die Klassifikatoren werden mit **denselben Testmengen** getestet. Die Trainingsmengen unterscheiden sich zwischen den Iterationen der Kreuzvalidierung nicht vollständig. Die Mengen sind damit nicht - wie beim t-Test angenommen - unabhängig voneinander.
- Die Wahrscheinlichkeit einen **Fehler erster Art** zu machen, ist hoch: Die Resultate eines Verfahrens würden häufiger fälschlicherweise als signifikant besser bzw. schlechter interpretiert werden [Vgl. Sal97, S. 325].

[4] Die Irrtumswahrscheinlichkeit gibt die Wahrscheinlichkeit eines Fehlers erster Art an. D. h. die Nullhypothese abzulehnen obwohl diese wahr ist. Das Signifikanzniveau α gibt den maximalen Wert für die Irrtumswahrscheinlichkeit an [Vgl. Voß+04, S. 422].

Aus den gleichen Gründen ist die Varianzanalyse für den Vergleich mehrerer Verfahren nicht geeignet. Nicht-parametrische Tests sollten verwendet werden. Sie machen keine Annahmen über Parameter, liefern dadurch weniger signifikante und folglich weniger falsch-positive Testergebnisse (Fehler erster Art) (Vgl. [Alp10, S.508ff], [Sal97, S. 325]). Für den Vergleich zweier Verfahren ist der Wilcoxon-Rangtest anwendbar. Nullhypothese ist, dass die ermittelten Bewertungsmaße den gleichen Verteilungen entstammen - kein Verfahren ist besser oder schlechter [Vgl. Voß+04, S. 476]. Eine ausführliche Erläuterung dieses Tests sowie des für den Vergleich mehrerer Verfahren geeigneten Kruskal-Wallis-Tests findet sich unter anderem bei [Alp10].

3. Betriebswirtschaftliche Grundlagen und Rahmenbedingungen

3.1. Direktmarketing

Der Fortschritt in der Informations- und Kommunikationstechnologie eröffnet heutigen Unternehmen neue und vor allem beschleunigte Wege zur direkten Kontaktaufnahme mit (potentiellen) Kunden. Als Kanäle stehen Telefon, Fax, Email, sogenannte Social Networking Plattformen und viele weitere zur Verfügung. Im Direktmarketing werden Instrumente und Prozesse entwickelt und etabliert, die Unternehmen bei der optimalen Nutzung der vielfältigen Möglichkeiten der direkten Kontaktaufnahme unterstützen [Vgl. Wir05, S. 3]. Unter „direkt“ ist hier einerseits die Ausschaltung von (Zwischen-) Händlern zu verstehen. Andererseits ist die individualisierte Ansprache mit dem Ziel der Interaktion mit dem Kunden zum Aufbau und Erhalt einer langfristigen Beziehung gemeint [Vgl. Wir05, S. 10].
Im nächsten Abschnitt wird zunächst die Abgrenzung zum indirekten (Massen-) Marketing erläutert. Danach folgt in Abschnitt 3.1.2 ein kurzer Überblick über die Aufgaben und Instrumente des Direktmarketings. Schließlich wird die Bedeutung des Direktmarketings für den Unternehmenserfolg betrachtet. Diese erklärt die Notwendigkeit der Identifikation von potentiellen Kunden und damit die Zielstellung dieser Arbeit.

3.1.1. Abgrenzung zum Massenmarketing

Unter dem Obergriff „Direktmarketing“ werden alle Methoden und Konzepte des Marketing-Mix zusammengefasst, die mit individualisierten Maßnahmen die direkte Interaktion mit dem Kunden erreichen wollen [Vgl. Win08, S. 5]. Damit kann das Direktmarketing klar vom Massenmarketing (auch als klassisches Marketing bezeichnet) abgegrenzt werden: Letzteres stellt lediglich Instrumente für den indirekten und nicht-individualisierten Kundenkontakt bereit. Darüberhinaus wird im Massenmarketing deutlich weniger Einzelwissen – Wissen über die individuellen Bedürfnisse der Kunden – benötigt [Vgl. BFU08, S. 409]. Die Kosten für die Vorbereitung und Durchführung sind folglich geringer als beim Direktmarketing. Ein weiterer wichtiger Unterschied liegt in der Zielstellung der beiden Ansätze: Das Direktmarketing strebt die langfristige Bindung des Kunden an ein Unternehmen und dessen Produkte an. Massenmarketing hingegen dient vorwiegend der Erzeugung einer einmaligen und oft einseitigen[1] Transaktion [Vgl. Wir05, S. 11ff]. Abbildung 3.1 visualisiert diese Abgrenzung von Massen- und Direktmarketing.

[1] Beispielsweise kauft ein Kunde ein Produkt in einem Supermarkt.

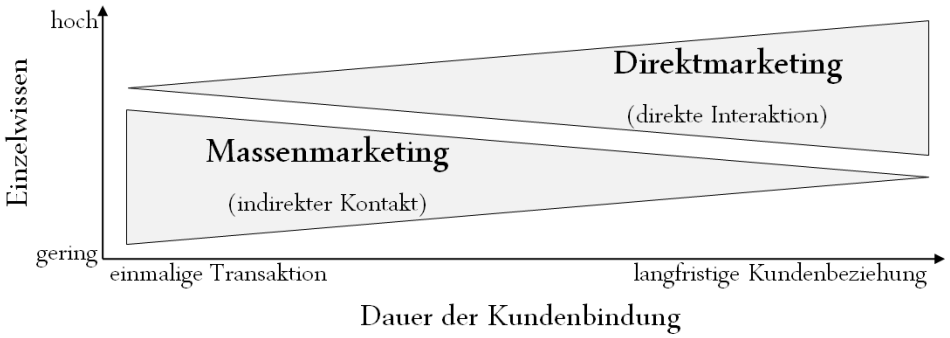

Abbildung 3.1. Abgrenzung von Direkt- und Massenmarketing nach vorhandenem Einzelwissen und Dauer der Kundenbindung [In Anlehnung an Wir05, S. 17].

Beide Varianten des Marketings erfüllen unterschiedliche Zielstellungen und sind verschieden aufwendig. Das Direktmarketing ist zu bevorzugen, wenn

- die angebotenen Produkte bzw. Dienstleistungen besonders erklärungsbedürftig sind,
- der Entscheidungsprozess komplex ist,
- Folgekäufe für die langfristige Sicherung des Unternehmenserfolges zwingend erforderlich sind [Vgl. Hol04, S. 9].

Um die Methoden des Direktmarktings einsetzen zu können, sind folgende Bedingungen zu erfüllen:

Die Zielgruppe ist individuell identifizierbar. Einerseits muss die Zielgruppe möglichst klein und damit überschaubar sowie feinkörnig analysierbar sein. Andererseits sind Kontaktdaten jedes (potentiellen) Kunden zu ermitteln.

Das Produkt bzw. die Dienstleistung ist höherwertig. Geringwertige Produkte oder Dienstleistungen sollten mit Methoden des Massenmarketings vermarktet werden. Da die Methoden des Direktmarketing deutlich zeit- und kostenintensiver sind, können sie nicht wirtschaftlich für die Vermarktung geringwertiger Güter eingesetzt werden [Vgl. Hol04, S. 7ff].

3.1.2. Instrumente

Die Instrumente des Direktmarketings sind äußerst vielfältig. Sie werden nach ihrer Zugehörigkeit zum Marketing-Teilbereich unterschieden:

Produktpolitik Neben der Weiterentwicklung von Konzepten zur Maximierung der Indivualisierungsmöglichkeiten von Produkten wird die Auswahl und Zusammenstellung des Sortiments im Direktmarketing unterstützend eingesetzt [Vgl. Hol04, S. 8]. Ein Beispiel sind die im Internet verfügbaren Programme zur individuellen Gestaltung von T-Shirts wie „www.shirtinator.de“.

Preispolitik In der Preispolitik werden sowohl Modelle der zeit- oder loyalitätsbedingten Preisbildung als auch Preisgleitklauseln, Preisgarantien und weitere individualisierte Formen der Preisbildung angewandt. Ebenso können individualisierte Liefer- und Zahlungsbedingungen positiven Einfluss auf die Kundenbindung haben [Vgl. Wir05, S. 109]. Exemplarisch sei auf die zahlreichen Punktesysteme wie DeutschlandCard verwiesen. Der Kunde bekommt Punkte für seine Treue, für die er wiederum Rabatte erhält. Der Preis richtet sich also nach dem individuellen Kaufverhalten.

Distributionspolitik In diesem Bereich wird zwischen Face-to-Face (persönlicher Kontakt und Verkauf), mediengestütztem (z. B. Telefonverkauf) und -geführtem (z. B. Onlineshops) Verkauf unterschieden [Vgl. Win08, S. 287f]. Insbesondere Internet-basierte Methoden gewinnen an Bedeutung [Vgl. BFU08, S. 422]. Beispiel für die Individualisierung in diesem Bereich sind die Kaufempfehlungen – basierend auf vorangegangen Käufen und Profilübereinstimmungen mit anderen Kunden – bei „www.amazon.de“

Kommunikationspolitik Neben Direktwerbemedien – wie Postwurfsendungen, telefonischen, fax- oder emailgestützten Werbeansprachen – werden Massenmedien mit Antwortelement eingesetzt. Als Antwortelemente werden diejenigen Bestandteile der Werbung bezeichnet, die den Kunden aktiv zur direkten Kontaktaufnahme auffordern. Massenmedien mit Antwortelement können das Internet, Funk oder Fernsehen sein. Beispielsweise werden besondere Vergünstigungen bei kurzfristiger Antwort beworben.

Die Mehrzahl der Instrumente ist in den beiden letztgenannten Bereichen angesiedelt [Vgl. Hol04, S. 6].
Alle Verfahren des Direktmarketings streben die direkte Interaktion mit potentiellen Kunden (auch als „Leads“ bezeichnet) an. Die Identifikation dieser Leads ist erforderlich. Kontaktdaten und möglichst viele zusätzliche Informationen werden für eine höchstmögliche Individualisierung der Kontaktaufnahme ermittelt. In Abschnitt 3.2.2 werden beispielhaft die bei der DECODON GmbH angewandten Techniken zur Identifikation von potentiellen Kunden beschrieben.

3.1.3. Bedeutung für den Unternehmenserfolg

Noch in den Zwanziger- und Dreißigerjahren des letzten Jahrhunderts lag der Fokus des Marketing auf der **Markenführung** zur Sicherung und Vermehrung des Unternehmenserfolgs. Sie fördert die Differenzierung und Identifikation von Firmen (Dachmarken) und Produkten (Familien- und Einzelmarken) und hilft dem Käufer bei der Orientierung und Entscheidungsfindung [Vgl. Man08, S. 128].
Marken sind „Vorstellungsbilder in den Köpfen der Anspruchsgruppen“ [TM05, S. 63], die

- zu *voreingenommener Wahrnehmung* von Unternehmen und Produkten führen,
- die *Abwertung von No-Name-Produkten*[2] fördern,

[2] Als No-Name-Produkte werden Produkte bezeichnet, die keiner starken Dach- oder Familienmarke angehören bzw. deren Name keine starke Einzelmarke ist.

- die *Toleranz bzw. Nachsichtigkeit* bei Mängeln erhöhen,
- zu *aktiven Beziehungspartnern* werden können

[Vgl. TM05, S. 236]. Bei starken[3] Marken sind die genannten Vorteile besonders ausgeprägt. Daraus folgt, dass erfolgreiches Markenmanagement

- zur Mehrpreis-Akzeptanz - dem sog. Preispremium - führt,
- die Immunität gegen Preiskämpfe fördert und somit die Notwendigkeit, aggressive und kurzfristige Verkaufsförderungsaktionen durchzuführen, mindert,
- die Belastbarkeit bei Veränderungen der Wettbewerbssituation erhöht,
- die Verhandlungsposition z. B. gegenüber Distributoren verbessert[4] [Vgl. MH06, S. 37f].

Chancen auf Erstkäufe und -aufträge werden ebenso positiv beeinflusst [Vgl. Hel06, S. 554] wie die Bindung zum Kunden [Vgl. IM06, S. 156].
Die Macht der Marke ist weiterhin unumstritten. Das Markenmanagement allein ist allerdings kein Garant für den Unternehmenserfolg. Seit den Fünzigerjahren werden Kundenzufriedenheit und -verhalten erforscht. Dadurch wurde die kundenbindende und absatzfördernde Wirkung der Methoden des Direktmarketings entdeckt. Die stetige Akquise von Aufträgen mittels direkter Marketingmethoden ist erfolgsentscheidend [Vgl. Hel06, S. 553]. Die Kombination von Markenmanagement und Direktmarketing im integrierten Kundenbindungsmanagement sichert demnach den langfristigen Erhalt und die Kapitalisierung von Unternehmen [Vgl. TM05, S. 233]. Einen Überblick über diese Entwicklung zeigt die Abbildung 3.2.
Der verstärkte Einsatz von direktem Marketing ist unter anderem mit der Entwicklung vom Massenmarkt zum Individualmarkt zu begründen: Kunden möchten individualisierte Angebote erhalten und ihren Bedürfnissen entsprechend angesprochen werden [Vgl. Win08, S. 24f]. Gleichzeitig bereichert jede direkte Interaktion mit einem Unternehmen das Markenerleben. Identifizieren sich Mitarbeiter mit der Marke, werden sie zum „Markenbotschafter“. Jeder Markenbotschafter steigert – unter Nutzung der Instrumente des Direktmarketings – den Markenwert [Vgl. Sch07].
Die Methoden des Direktmarketing sind inzwischen in der Marketingtheorie anerkannt und in der Praxis weit verbreitet [Vgl. Wir05, S. 4]. Die zunehmende Verbreitung hat allerdings auch zu einer zunehmenden Abwehrhaltung und Abkapselung („Cocooning“) der potentiellen Kunden geführt. Aus diesem Grund gewinnt das Empfehlungsmarketing an Bedeutung [Vgl. Fin08, S. 20]. Ziel ist die Förderung des Ausprechens von Empfehlungen durch Kunden. Das Empfehlungsmarketing nutzt die Instrumente des Direktmarketing: Die regelmäßige, direkte Interaktion mit Bestandskunden fördert deren Bindung zum Unternehmen und dessen Produkten. Diese Bindung fördert die Bereitschaft des Kunden, Empfehlungen auszusprechen [Vgl.

[3] Ein guter Überblick über Methoden für die Quantifizierung des Markenwertes findet sich beispielsweise in [Win08] oder [Sch04].

[4] Diese verbesserte Verhandlungsposition führt zu Vorteilen wie geringeren Werbekostenzuschüsse, höhere Listungsgebühren und viele weiteren.

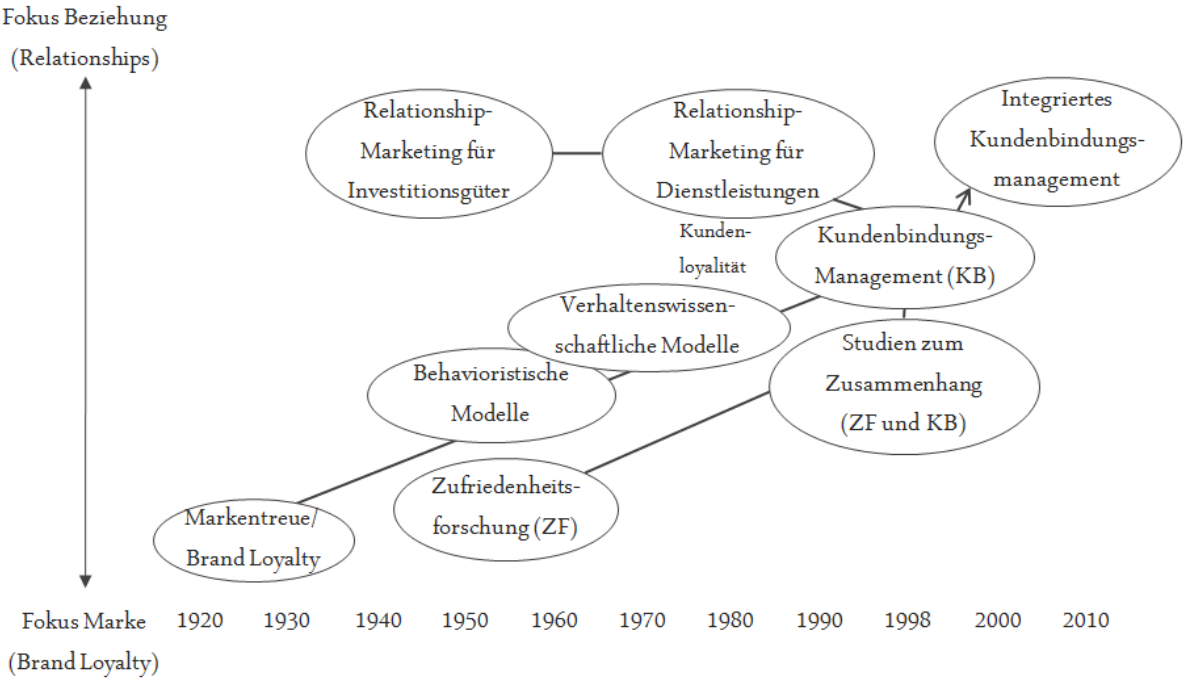

Abbildung 3.2. Entwicklungstendenzen des Kundenbindungsmanagements [In Anlehnung an TM05, S. 235].

Wir05, S. 611]. Die Empfehlungen wiederum unterstützen bzw. ermöglichen überhaupt erst Kontakte zu neuen Interessenten [Vgl. Fin08, S. 20].
Empfehlungsmarketing unter Nutzung des Direktmarketing könnte bespielsweise in folgenden Schritten ablaufen:

1. **Identifikation eines potentiellen Kunden**

   Die hier einzusetzenden Methoden sind abhängig von Branche und Zielgruppe sehr unterschiedlich. Beispielhaft sei auf die in Abschnitt 3.2.2 beschriebenen Varianten verwiesen.

2. **Ermittlung von relevanten Bestandskunden**

   Dies sollten Kunden sein, die den potentiellen Kunden kennen könnten, weil sie z.B. in der gleichen Branche, Region usw. tätig sind.

3. **Kontaktaufnahme mit den relevanten Bestandskunden**

   Ziel dieses Schrittes muss die Ermutigung des Kunden zur Empfehlung der Produkte bzw. Dienstleistungen beim im ersten Schritt identifizierten potentiellen Neukunden sein. Beispielsweise könnten Prämien für die Weiterempfehlung in Aussicht gestellt werden.

Im Idealfall wird der potentielle Kunde nun die Initiative ergreifen, d. h. er wird Kontakt mit dem Unternehmen aufnehmen. Ist sicher, dass eine Empfehlung ausgesprochen wurde, ist nun auch eine direkte Ansprache mit Bezug auf den Bestandskunden erfolgversprechender.

Neben dem Empfehlungsmarketing entstehen weitere Strömungen im Marketing, die die Aufmerksamkeit und das Interesse von Leads trotz der Vielzahl der Direktmarketing-Aktivitäten zahlreicher Unternehmen wecken sollen. In den letzten Jahren gewinnt vor allem das Neuromarketing an Bedeutung. Erkenntnisse aus der Hirnforschung werden für die Optimierung des Marketings genutzt. Sie werden im Direktmarketing zukünftig verstärkt Anwendung finden [Vgl. WM07].
Zusammenfassend kann festgehalten werden, dass die Instrumente und Konzepte des Direktmarketing aus der modernen Marketingtheorie und -praxis nicht mehr wegzudenken sind. Durch kontinuierliche Optimierung vorhandener und Entwicklung neuer Techniken wird das Direktmarketing entscheidend für den Erfolg von Unternehmen bleiben.

3.2. Das Bioinformatik-Unternehmen DECODON

Die DECODON GmbH ist ein mittelständisches[5] Bioinformatik-Unternehmen mit Sitz in Greifswald. Sie wurde am 19. September 2000 als Spin-Off der Ernst-Moritz-Arndt-Universität Greifswald gegründet. Das Unternehmen beschäftigt zehn fest angestellte und einige freie Mitarbeiter (Stand: Dezember 2010). Letztere unterstützen DECODON bei Software-Auftragsentwicklungen.
Im folgenden Abschnitt werden Kerngeschäft und Zielgruppe der DECODON GmbH beschrieben. Darauf aufbauend folgt eine Erläuterung der eingesetzten Methoden zur Identifikation potentieller Kunden. Insbesondere dieser Teil wird das Verständnis für die Nützlichkeit der vorliegenden Arbeit fördern.

3.2.1. Kerngeschäft, Zielgruppe und Konkurrenz

Kerngeschäft von DECODON ist die *Entwicklung und Vermarktung von Software* zur Unterstützung der Forschung im Bereich der Lebenswissenschaften[6]. Darüber hinaus werden

- *Auftragsentwicklungen* (häufig im Rahmen von Forschungsprojekten),
- *software-basierte Auftragsanalysen* sowie
- *(Produkt-)Schulungen*

angeboten und durchgeführt.
Besonderer Fokus liegt auf der Entwicklung und Vermarktung der Bildanalysesoftware **Delta2D**. Analysiert werden Bilder von Gelen, auf denen Proteine mit Hilfe der zweidimensionalen Gelelektrophorese[7] nach Ladung und Masse aufgetrennt wurden. Abbildung 3.3 zeigt ein solches „zwei-dimensionales Gelbild“. Jeder Fleck – Spot genannt – auf dem Bild repräsentiert – bei gelungener Trennung – genau ein Protein. Mehrere tausend können auf einem einzigen Gel aufgetrennt werden [Vgl. JEC06, S. 22].

[5] Laut dem Institut für Mittelstandsforschung (IfM) zählen zum Mittelstand alle kleinen und mittleren Unternehmen (KMU). DECODON ist gemäß KMU-Definition ein Unternehmen mittlerer Größe [Vgl. MI10].
[6] Diesem Bereich werden unter anderem die Disziplinen Medizin, Zahnmedizin, Veterinärmedizin, Biologie, Genetik, Biochemie, Zellbiologie, Biotechnologie und Biomedizin zugeordnet [Vgl. NCB10a, S. 1].
[7] Die Methode wird beispielsweise bei [O'F75] und [Klo75] erläutert.

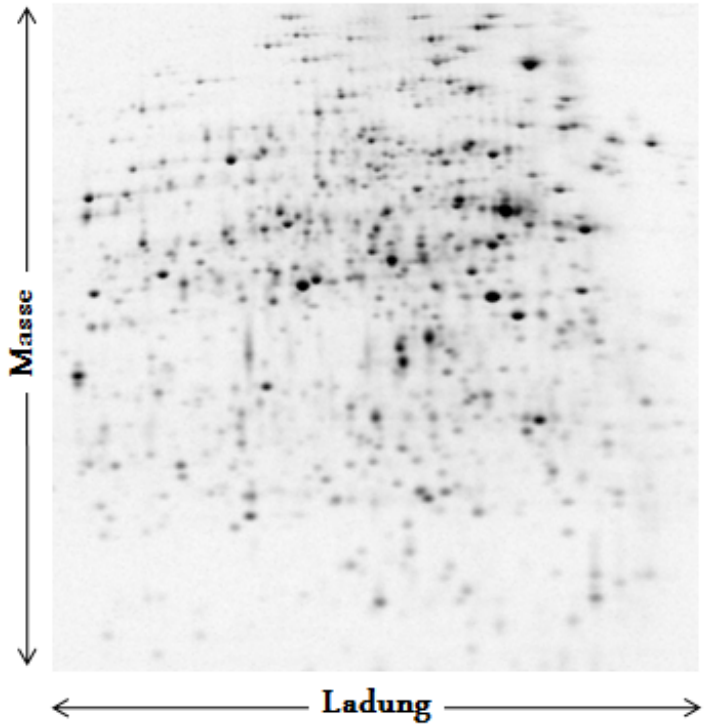

Abbildung 3.3. Zwei-dimensionales Gelbild (Quelle: DECODON GmbH).

Ziel der Anwendung der Methode ist die Entdeckung von Proteinen, die beispielsweise im Blut von kranken Patienten ausschließlich, häufiger oder vermindert auftreten. Diese werden als Biomarker-Kandidaten bezeichnet [Vgl. HJI06, S. 248]. Dazu werden

1. Proben aus dem Blut von gesunden und kranken Menschen entnommen,
2. auf mehreren Gelen aufgetrennt[8] und mittels diverser Färbetechniken [Vgl. JEC06, S. 27ff] sichtbar gemacht,
3. die Gele eingescannt und
4. die auf diese Weise entstandenen Bilder software-basiert verglichen.

Ergebnis des Vergleichs der Bilder ist in der Regel eine Liste von Spots, die Biomarker sein könnten. Diese Biomarker-Kandidaten müssen anschließend weiter erforscht und in einem aufwändigen Prozess verifiziert und validiert werden. Gelingt dies, können die Biomarker als Grundlage für die Entwicklung neuer Medikamente und Diagnostika verwendet werden.
Der Weg vom zwei-dimensionalen Gel bis zum vermarktungsfähigen Produkt ist folglich ein kosten- und zeitintensiver, jahrelanger Prozess der Grundlagenforschung. Die Methode wird

[8] In der Regel werden zusätzlich technische – die gleiche Probe auf mehreren Gelen – und biologische – mehrere Proben von Individuen, die identisch behandelt wurden – Replikate angefertigt, um möglichst verlässliche Ergebnisse zu erhalten.

daher vorwiegend in Universitäten und Forschungsinstituten sowie von im Auftrag von Pharmaunternehmen forschenden Unternehmen angewandt. Die Forschung von Pharmaunternehmen setzt in der Regel zu einem späteren Zeitpunkt im Prozess ein.
Zielgruppe für Delta2D sind demzufolge Arbeitsgruppen in Universitäten und Forschungsinstituten, die im Bereich der Lebenswissenschaften proteinanalytische Forschung betreiben.
Das Marketing erfolgt weltweit, konzentriert sich jedoch auf Europa und Nordamerika. Delta2D ist ein Nischenprodukt. Die Liste der **Konkurrenzprodukte** ist überschaubar. Neben Großunternehmen wie General Electrics (GE) Healthcare und BioRad treten andere Bioinformatik-Unternehmen als Konkurrenten auf. Tabelle 3.1 nennt die Konkurrenten (Stand: Oktober 2010). Insbesondere die genannten Großunternehmen haben Wettbewerbsvorteile

Unternehmen	Produkt(e)
Applied Maths	Bionumerics2D
Bio-Rad	PDQuest, ProteomWeaver
DECODON	Delta2D
GE Healthcare	Decyder 2D, ImageMaster Platinum
Genebio	Melanie
Nonlinear Dynamics	SameSpots
Syngene	Dymension

Tabelle 3.1. Produkte für die Auswertung zwei-dimensionaler Gelbilder und deren Anbieter [In Anlehnung an Ber+07, S. 1224]

durch den höheren Markenwert. GE zählt bespielsweise zu den bekanntesten und wertvollsten Marken weltweit [Vgl. Mil10, S. 16], [Vgl. Int10, S. 2]. Zudem haben diese Unternehmen ein großes Produktportfolio für die Forschung in den Lebenswissenschaften. Sie sind dadurch potentiellen Kunden von Analysesoftware für zwei-dimensionale Gelbilder bereits aus anderen Forschungsprojekten bekannt. Oft bestand oder besteht direkter Kontakt zu Vertretern dieser Unternehmen.
DECODON muss folglich die Stärkung der Unternehmensmarke „DECODON“ und der Produktmarke „Delta2D“ fördern. Die Notwendigkeit der kontinuierlichen Weiterentwicklung von Delta2D macht überdies – in Verbindung mit der relativ kleinen Zielgruppe – die langfristige Kundenbindung zur Generierung von Folgeaufträgen notwendig. Deswegen und aufgrund der Tatsache, dass Delta2D ein höherwertiges, äußerst erklärungsbedürftiges Produkt ist, ist der kombinierte Einsatz von Methoden des Direktmarketings unerlässlich.

3.2.2. Identifikation von potentiellen Kunden

Um langfristig auf dem Markt zu bestehen, müssen Methoden des Markenmanagements und des Direktmarketings kombiniert werden [Vgl. Abschnitt 3.1.3]. Letzteres ist nur möglich, wenn potentielle Kunden identifiziert wurden.

Eine erste wichtige Methode zur Identifikation von potentiellen Kunden ist die **Befragung von Bestandskunden**. Kunden kennen potentielle Käufer von Fachkonferenzen, durch gemeinsame Forschungsprojekte, weil sie in der Vergangenheit in der gleichen Arbeitsgruppe tätig waren oder weil sie ihnen durch Publikation von Forschungsergebnissen aufgefallen sind.
Weiterhin werden potentielle Kunden auf **Fachkonferenzen** identifiziert: Die Proteinanalytik ist in den letzten Jahren zu einer bedeutenden Teildisziplin der Lebenswissenschaften gewachsen, die fachübergreifend eingesetzt wird. Zahlreiche Fachkonferenzen werden zum wissenschaftlichen Austausch veranstaltet. Bei diesen Konferenzen können Unternehmen an Ständen, wie sie auch bei Messe üblich sind, ihre Produkte und Dienstleistungen vorstellen. Die zweidimensionale Gelelektrophorese – als eines der wichtigsten Verfahren in diesem Bereich [Vgl. Rab+10], [Vgl. VCP09] – wird von vielen Konferenzteilnehmern eingesetzt. Sie sind potentielle Kunden für DECODON und Kontakte zu ihnen werden durch Außendienstmitarbeiter geknüpft.
Zusätzlich steht den Vertriebsmitarbeitern eine interne **Adressdatenbank** (Vgl. Abschnitt 4.1) als Quelle zur Verfügung. Die Außendienstmitarbeiter von DECODON durchsuchen die Adressdatenbank regelmäßig nach Personen, die

- noch keinen bzw. lange keinen Kontakt mit Unternehmensvertretern hatten,
- im gleichen Forschungsgebiet wie ein (neuer) Kunde tätig sind oder
- in einem Gebiet wohnen, in dem Kundenbesuche geplant wurden.

Auch das **Internet** wird durchsucht. Im universitären Bereich veröffentlichen viele Forscher – auf personen- oder arbeitsgruppenbezogenen Websites – Informationen über ihre aktuellen Forschungsprojekte. Häufig werden Kontaktdaten angegeben. Zudem werden Informationen über Fachkonferenzen – dort gehaltene Vorträge oder veröffentlichte Poster über Forschungsfortschritt und -resultate – publiziert. In vielen Ländern und Regionen sind nicht-kommerzielle Organisationen gegründet worden, die die Interessen der Forscher der Proteinanalytik wahren. Diese Organisationen sind meist im Internet mit einer Website vertreten. Auf der Website werden Informationen über gemeinsame Aktivitäten, Mitglieder und vieles mehr veröffentlicht. Zudem gibt es an einigen Universitäten nicht-kommerziell tätige Service-Abteilungen, die für die Abteilungen Schulungen und Auftragsforschung anbieten. Exemplarisch ist in Abblidung 3.4 die Website der Service-Abteilung für Proteinanalytik der Universität Göteborg zu sehen. Die Vertriebsmitarbeiter von DECODON googeln nach den genannten Seiten. Suchbegriffe sind z. B. „zwei-dimensionale Gelelektrophorese“ in diversen Sprachen oder häufig in Verbindung mit der Methode verwendete Terme. Auch Namen von Konkurrenzprodukten werden erfolgreich verwendet.
Schließlich ist die **PubMed-Datenbank** eine wichtige Quelle für die Identifikation neuer Kunden. Sie beinhaltet - neben vielen weiteren Forschungsgebieten - nahezu allen relevanten wissenschaftlichen Publikationen zur Proteinanalytik. Mit Suchbegriffen kann die Suche eingeschränkt werden. Die Datenbank wird in Abschnitt 4.2 beschrieben.
Alle genannten Methoden sind sie sehr zeitintensiv. Die Suche in PubMed, die im Rahmen dieser Arbeit optimiert werden soll, ist manuell besonders aufwendig: Die Trefferliste ist nach

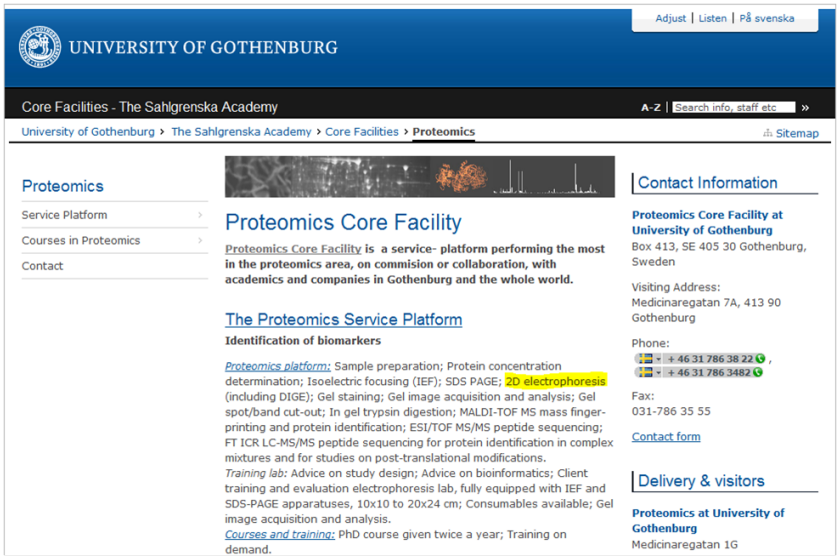

Abbildung 3.4. Website der Service-Abteilung für Proteinanalytik der Universität Göteborg (Screenshot).

relevanten Publikationen zu durchsuchen. Veröffentlichungen von Autoren, die bereits Kunden sind oder aus anderen Gründen[9] nicht als potentielle Kunden in Frage kommen, müssen manuell herausgefiltert werden. Die Verringerung des benötigten Zeitaufwandes zur Identifikation potentieller Kunden ist wünschenswert.

[9]Beispielsweise könnte die zwei-dimensionale Gelelektrophorese im Rahmen einer Kooperation von einer anderen Gruppe gemacht worden sein.

4. Beschreibung der Datenquellen

Im ersten Abschnitt dieses Kapitels wird die Adressdatenbank der DECODON GmbH vorgestellt. Sie dient nicht als Datenquelle für die in Kapitel 6 erläuterte Modellbildung, unterstützt aber die Maximierung des betriebswirtschaftlichen Nutzens der Webanwendung (Vgl. Kapitel 8).
Das Verständnis der PubMed-Datenbank – als Datenbasis für die Erfüllung der Data-Mining-Ziele dieser Arbeit – ist im Sinne des CRISP-DM-Referenzmodells erforderlich. Sie wird im zweiten Teil dieses Kapitels detailliert betrachtet.

4.1. Adressdatenbank der DECODON GmbH

Die Adressdatenbank der DECODON GmbH wird seit dem Jahr 2002 im Tagesgeschäft eingesetzt. Sie dient der internen Verwaltung und Speicherung von Informationen zu Kontaktpersonen des Unternehmens. Im November 2010 beinhaltete die Datenbank knapp 13.000 Datensätze.
Diese Eigenentwicklung der DECODON GmbH basiert auf dem Open-Source-Webanwendungsserver ZOPE (Z Object Publishing Environment) und der dazugehörigen Datenbank ZODB (ZOPE Object Database). Anfragen können mit HTTPS GET Requests durchgeführt werden. Zugang zur Datenbank haben alle Mitarbeiter des Unternehmens.[1]
Eine Person wird als Kontakt in der Adressdatenbank gespeichert, wenn sie

- eine Demoversion von Delta2D von der Website heruntergeladen hat (Registrierung erforderlich),
- sich für den Newsletter auf der Website anmeldet,
- Informationen über Produkte und Preise angefordert hat oder
- durch Vertriebsmitarbeiter als potentieller Kunde identifiziert wurde (Vgl. Abschnitt 3.2.2)

Neben Kontaktdaten wie Email, Telefonnummer und Adresse wird jeder Person eine Organisation und ein *Ansprechpartner* bei DECODON zugeordnet. Ebenso können beliebig viele *Tags* (beschreibende Stichwörter) oder *Gruppen* zugeordnet werden. Zusätzlich besteht die Möglichkeit, Kontaktinstanzen mit *Issues* zu verbinden. Issues werden in Lizenz- und Kontaktissues untergliedert. Erstere stellen die Beschreibung einer (Software-)Lizenz dar, die der

[1] Stand der Informationen: Dezember 2010.

Kontaktperson zur Verfügung gestellt wurde. Letztere dienen der Dokumentation der Kommunikation mit der Person. Kommunikationskanal und -inhalt werden mit Datum und Uhrzeit abgelegt. Beiden Arten von Issues kann ein Wiedervorlage-Datum zugeordnet werden. An dem entsprechenden Tag werden die Issues in einer Wiedervorlage-Liste angezeigt. Dies erleichtert den Mitarbeitern von DECODON das Einhalten von Terminen und unterstützt das Aufrechterhalten des Kontakts mit Kunden. Die Verbindung zwischen Kontakten und Issues ist in der nachfolgenden Abbildung 4.1 dargestellt.

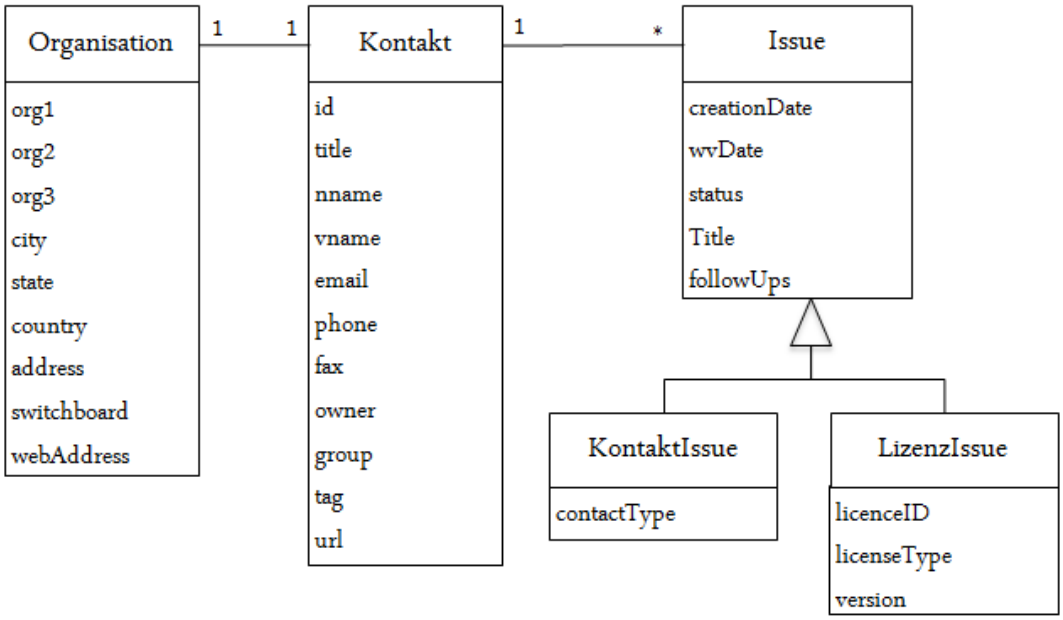

Abbildung 4.1. Vereinfachte Darstellung relevanter Objekte in der Adressdatenbank der DECODON GmbH (Eigene Darstellung).

Für die Anmeldung zum Newsletter ist lediglich die Angabe einer Email-Adresse erforderlich. Diese lässt oft keine Rückschlüsse auf Identität oder Organisationszugehörigkeit des Interessenten zu. Beim Download sind zumindest die Nennung einer validen Email-Adresse und der Organisationszugehörigkeit obligatorisch. Erstere verrät meist wenig über die Person. Bei letzterer genügt die Angabe eines – z. B. durch Webrecherche nachvollziehbar – existenten Organisationsnamens[2]. Geben die Interessenten keine organisationsspezifische Email-Adresse (z. B. max.mustermann@uni-musterhausen.de) an, kann die Korrektheit der Organisationszuhörigkeit nicht mit Sicherheit bestätigt werden. Mittels Webrecherche versuchen die Mitarbeiter von DECODON, jeden neu entstandenen Datensatz zu verifizieren, zu vervollständigen und Fehler zu korrigieren. Die Fehlerkorrektur umfasst typografische Fehler sowie die richtige Zuordnung der Informationen zu den Attributen des Kontaktes. Letzteres wird beispielsweise erforderlich, wenn ein Downloader seinen Nachnamen in das Feld für den Vornamen eingetragen hat.

[2]Jede Downloadanfrage mit ausgefüllten Pflichtfeldern berechtigt zum Download einer Installationsdatei für Delta2D. Zudem erhält der Downloader eine Lizenz für vier Tage. Mitarbeiter von DECODON senden – nach Prüfung der Organisationsangabe – eine weitere, für dreißig Tage gültige Lizenz via Email zu.

Trotz stetiger Bemühungen weist die Qualität der Daten in der Adressdatenbank Mängel auf.

Unvollständige Datensätze. In einigen Fällen liefert die Webrecherche keine weiteren Informationen zur Vervollständigung eines Datensatzes. Auf die Eintragung von vermeintlich unwichtigen Informationen wie Bundesland oder Telefonnummer der Organisationszentrale wird zugunsten anderer Aufgaben der Mitarbeiter von DECODON verzichtet. Wechseln Kontaktpersonen die Organisation, sind die bisherigen Kontaktdaten nicht mehr valide. Die Ermittlung der neuen Daten ist manchmal nicht möglich.

Gleiche Organisationen mit unterschiedlichen Namen. Organisationen aus dem deutschsprachigen Raum werden mit ihren deutschen Namen, ausländische Organisationen mit der englischen Übersetzung eingetragen. Diese Vorgehensweise ist üblich, da offizielle Dokumente wie Angebote und Rechnungen an deutschsprachige Kunden in deutsch, an alle anderen Kunden in englisch verfasst werden. Bei einigen Datensätzen wurde diese „inoffizielle Konvention“ nicht eingehalten. Folglich existieren Einträge in anderen Sprachen als deutsch oder englisch. Einige deutsche Organisationen sind mit der englischen Bezeichnung eingetragen und andersherum. Weiterhin unterscheidet sich die Genauigkeit, mit der Organisationsbezeichnungen eingetragen werden. Die „Ernst-Moritz-Arndt-Universität Greifswald“ ist beispielsweise neben der vollständigen Bezeichnung auf deutsch als „University of Greifswald“, „Greifswald University“, „EMAU Greifswald“ oder „EMAU“ eingetragen.

Voneinander verschiedene Kontaktpersonen mit gleichen Namen. Dies ist nicht nur ein Problem in der Adressdatenbank von DECODON. Weltweit haben viele Menschen vollkommen identische Namen. Einige Datensätze sind nicht eindeutig einem Individuum zuordenbar.

Vertauschte Attributwerte. Ursache kann z. B. die unzureichende Korrektur der Angaben von Downloadern sein. Ebenso unterlaufen bei der Eintragung von identifizierten potentiellen Kunden Fehler: In der Adressdatenbank können bis zu drei Hierarchieebenen angegeben werden. Zweite und dritte Hierarchieebene (z. B. Institut einer Universität und Arbeitsgruppe) können versehentlich vertauscht werden.

4.2. PubMed

PubMed ist eine Meta-Datenbank, die vom National Center for Biotechnology Information (NCBI) entwickelt wurde. NCBI ist eine Unterorganisation der National Library of Medicine (NLM), welche wiederum eine Unterorganisation der National Institutes of Health (NIH) der Vereinigten Staaten von Amerika ist. Herausgeber von mehr als 5000 Zeitschriften und Online-Büchern aus allen Teilbereichen der Lebenswissenschaften veröffentlichen wissenschaftliche Arbeiten – Publikationen genannt – in PubMed [Vgl. NCB10a, S. 1]. Im Dezember 2010 beinhaltete die Datenbank mehr als 20 Millionen Publikationen, jährlich kommen etwa 500.000 hinzu.

Die meisten Publikationen stammen aus MEDLINE. Dies ist die wichtigste bibliographische Datenbank der National Library of Medicine. MEDLINE enthält Publikationen aus den Lebenswissenschaften mit besonderem Fokus auf Biomedizin [Vgl. NLM10b].
Aufgrund der Fülle der verfügbaren Daten ist die manuelle Untersuchung von PubMed ohne Werkzeuge mindestens sehr zeitintensiv, häufig nicht realisierbar. PubMed ist daher eine häufig verwendete Datenquelle für die Forschung in den Bereichen Information Retrieval, Information Extraction und Text Mining [Vgl. FS07, S. 2].
Im folgenden Abschnitt werden zunächst die verfügbaren Ausgabeformate erläutert. Anschließend werden die Attribute von Publikationen betrachtet. Die Qualität der PubMed-Einträge ist Bestandteil des letzten Teils dieser Sektion.

4.2.1. Formate und Zugriffsmöglichkeiten

Der Zugriff auf PubMed ist über eine öffentlich-zugängliche Webschnittstelle mittels **HTTP GET Request** möglich. Nutzer können die Datenbank nach für sie relevanten Publikationen durchsuchen. Jedem Attribut einer Publikation ist ein einzigartiger Deskriptor („Search Field Tag“) zugewiesen. Der Benutzer kann diesen verwenden, um die Suche auf die jeweiligen Attribute zu beschränken. Andernfalls wird in allen Attributfeldern gesucht.
Standard-Ausgabeformat für die Trefferliste der Suche ist HTML. Maximal 20 Suchergebnisse werden pro Seite – sortiert nach Eingangsdatum bei PubMed – angezeigt. Zu jedem Treffer werden Titel, veröffentlichende Zeitschrift, Publikationsdatum und -status sowie – wenn vorhanden – die Namen der Autoren angegeben (Vgl. 4.2.2). Abbildung 4.2 zeigt beispielhaft eine Ergebnisseite einer PubMed-Suche.

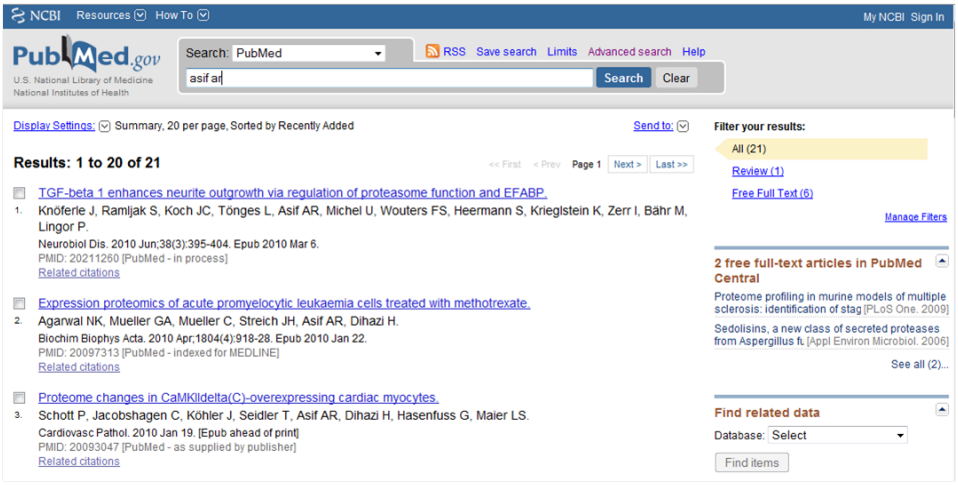

Abbildung 4.2. Beispiel für eine Trefferliste einer PubMed-Suche (Screenshot).

Die Ausgabe ist auf 200 Treffer je Ergebnisseite limitiert. Zu jeder Publikation kann – wenn verfügbar – eine Kurzzusammenfassung („Abstract“) aufgerufen werden. Die Ausgabe erfolgt standardmäßig im HTML-Format. **Volltexte werden in PubMed nicht veröffentlicht.** Sie sind – häufig kostenpflichtig – über die Websites der Zeitschriften erhältlich. Die entsprechenden Links werden – wenn vorhanden – angezeigt [Vgl. NCB10a, S. 5].

Für automatisierte Suchanfragen nach Dokumenten bietet NCBI **„Entrez Programming Utilities“** an. Diese Sammlung von Hilfswerkzeugen ermöglicht den Zugriff auf alle öffentlichen, von NCBI betreuten Datenbanken. Mit **„ESearch“** können beispielsweise Identifikationsnummern von Suchergebnissen abgefragt werden. Die maximale Anzahl kann vom Nutzer frei gewählt werden. Bis zu drei Suchanfragen dürfen pro Sekunde gesendet werden. Die Liste der gefundenen Identifikationsnummern kann anschließend verwendet werden, um z. B. mit „EFetch“ weitere Informationen zu dem Dokument zu erhalten.
Einträge werden nicht nur in HTML ausgegeben. Zusätzlich stehen reiner Text, MEDLINE und XML als Ausgabeformate zur Auswahl. Für den Menschen sind **HTML** und **reiner Text** einfach lesbar. Insbesondere die Lesbarkeit von HTML ist durch die unterschiedliche Formatierung der Attribute gegeben. Beide Formate sind allerdings für automatisierte Auslesen weniger geeignet: Attribute sind im reinen Text lediglich durch Absätze getrennt, im HTML durch formatierende Tags. Schlüssel für Attribute sind nicht vorhanden.
Das **MEDLINE**-Format ist Text, in dem jede Publikation einen Absatz einnimmt. Absätze sind durch Leerzeilen voneinander getrennt. Innerhalb einer Publikation beginnt jede Zeile einem eindeutigen Attributsschlüssel oder fünf Leerzeichen. Der Schlüssel besteht aus vier Zeichen – Großbuchstaben oder Leerzeichen – gefolgt von einem Bindestrich und einem Leerzeichen. Der Schlüssel

- beginnt mit mindestens zwei Großbuchstaben und
- nach einem Leerzeichen darf kein Großbuchstabe mehr folgen.

Nach dem Schlüssel folgt der Inhalt des jeweiligen Attributs. Beginnt die folgende Zeile mit fünf Leerzeichen, d. h. nicht mit einem Attributsschlüssel, ist der Inhalt dem Attribut zuzuordnen, für das zuletzt ein Schlüssel angegeben wurde. In Abbildung 4.3 ist exemplarisch der Anfang einer Publikation – dargestellt im MEDLINE-Format – zu sehen.

```
PMID- 19919853
OWN - NLM
STAT- MEDLINE
DA  - 20100215
DCOM- 20100325
IS  - 1096-0937 (Electronic)
IS  - 1087-1845 (Linking)
VI  - 47
IP  - 3
DP  - 2010 Mar
TI  - Proteomic analysis of early phase of conidia germination in Aspergillus nidulans.
PG  - 246-53
AB  - In order to investigate proteins involved in early phase of conidia germination,
      proteomic analysis was performed using two-dimensional gel electrophoresis
      (2D-GE) in conjunction with MALDI-TOF mass spectrometry (MS). The expression
      ...
CI  - Copyright (c) 2009 Elsevier Inc. All rights reserved.
AD  - Department of Microbiology and Reserch Institute of Life Science, Gyeongsang
      National University, Jinju, Republic of Korea.
```

Abbildung 4.3. Ausschnitt aus einer Publikation im MEDLINE-Format (Quelle: PubMed).

Neben MEDLINE können Publikationen im **XML**-Format abgerufen werden. Abbildung 4.4 zeigt einen Teil der Darstellung in diesem Format. XML bietet gegenüber MEDLINE den Vorteil, dass Elemente hierarchisch angeordnet werden können. Das Element „AuthorList“ hat beispielsweise Kindelemente „Author“, welche jeweils ein Kindelement „LastName“, „ForeName“ und „Initials“ haben. PubMed-spezifische Daten wie der Status (Vgl. Attribut „Status“ von „MedlineCitation“ in 4.2.2) oder das Eingangsdatum (Vgl. „DateCreated“ in 4.2.2)

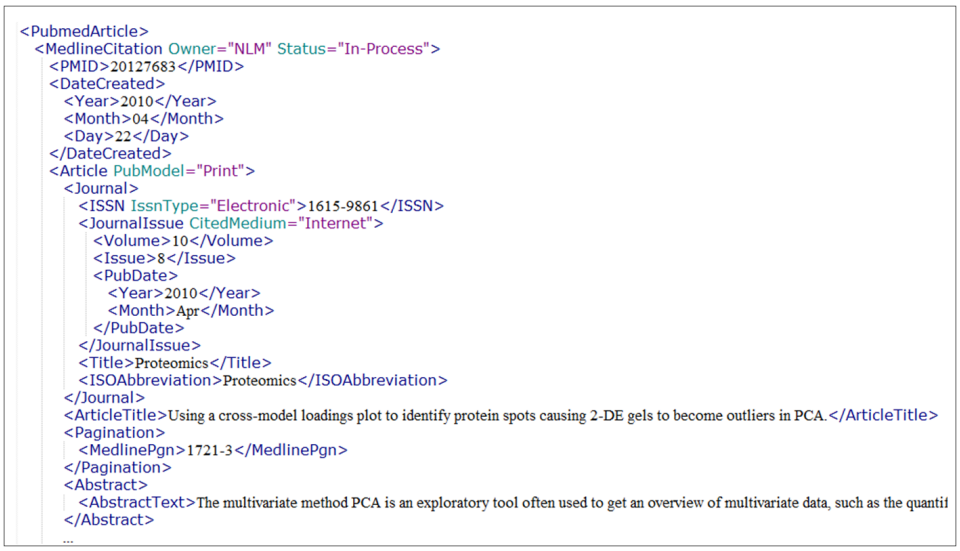

Abbildung 4.4. Beginn einer PubMed-Publikation im XML-Format (Quelle: PubMed).

sind klar von den publikationsspezifischen Daten getrennt (Vgl. „Article“ in 4.2.2). XML wird nicht nur in für PubMed-Publikationen eingesetzt. Das Format wird in verschiedensten Bereichen und für die Ablage diverser Daten – wie beispielsweise für die Reuters-Sammlung[3] [Vgl. Reu10a] – verwendet. Dadurch sind für das Auslesen von XML-Dateien zahlreiche freie Pakete verfügbar. Im weiteren Verlauf werden aus PubMed nur Informationen im XML-Format verwendet.
Unabhängig vom Ausgabeformat sind alle Daten in UTF-8 (Unicode Transformation Format 8bit) kodiert.[4] [Vgl. NLM10c]

4.2.2. Attribute von PubMed-Einträgen

Einige der Elemente der XML-Darstellung von Publikationen tragen Namen, die mit „Medline“ beginnen. Grund hierfür ist die Vermeidung von Irritationen durch neue, von den MEDLINE-Attributsnamen abweichende Bezeichnungen. Alle nachfolgenden Angaben entsprechen den Bezeichnungen, wie sie im XML-Format verwendet werden.
Werden Publikationen in PubMed zitiert, sind laut National Library of Medicine mindestens die nachfolgenden Angaben zu machen [Vgl. NLM10f]:

MedlineCitation Dieses Element hat die Attribute „Owner“ und „Status“. Beide müssen angegeben werden. Der *Owner* ist die für die Eintragung verantwortliche Organisation. Neben der NLM können dies die National Aeronautics and Space Administration (NASA), das Kennedy Institute of Ethics (KIE) und einige weitere berechtigte Organisationen sein [Vgl. NLM10d, S. 2].

[3] Die Nachrichtenagentur Reuters stellt Sammlungen von Nachrichten-Dokumenten für die Forschung frei zugänglich zur Verfügung [Vgl. Reu10b].

[4] UTF-8 wird zur Kodierung von Unicode-Zeichen verwendet. Das Format ist weit verbreitet – nicht zuletzt weil es mit diversen auf ASCII-basierenden Dateisystemen und Parsern kompatibel ist [Yer03].

Jeder Eintrag in PubMed durchläuft einen Qualitätssicherungsprozess. Dieser Prozess ist in mehrere Phasen untergliedert. Der *Status* gibt an, in welcher Phase des Eintragungsprozesses eine Publikation ist. Ist der Prozess abgeschlossen, sind drei Status möglich: MEDLINE (Original-MEDLINE), OLDMEDLINE (MEDLINE-Publikationen, deren MeSH-Liste noch aktualisiert werden muss), PubMed-not-Medline (PubMed-Publikationen, die nicht in MEDLINE zitiert werden).

PMID Der PubMed unique Identifier (PMID) ist eine Identifikationsnummer, die mindestens aus sechs und höchstens aus acht Ziffern besteht.

DateCreated Das „DateCreated"-Element beinhaltet das Datum, zu dem eine Publikation zu PubMed hinzugefügt wurde.

DateCompleted Dieses Element gibt das Datum der Beendigung der Qualitätssicherung an.

Article „Article" muss das Attribut „PubModel" enthalten, welches die Form angibt, in der die publizierende Zeitschrift veröffentlicht wird [Vgl. NLM10f]. Erlaubte Werte sind „Print" (gedruckt), „Electronic" (elektronisch), „Print-Electronic" (gedruckt und elektronisch) und „Electronic-Print" (erst elektronisch, dann gedruckt) [Vgl. NLM10d, S. 6f]. Zusätzlich sind folgende Unterelemente anzugeben:

- **ISSN** Die International Standard Serial Number (ISSN) besteht aus acht Zeichen, wobei die ersten vier von den letzten vier durch einen Bindestrich getrennt sind. Das Element enthält diese Nummer und das Attribut „IssnType".[5] *IssnType* gibt an, in welcher Form die Publikation veröffentlicht wurde - gedruckt (Print) oder elektronisch (Electronic) [Vgl. NLM10d, S. 7].
- **PubDate** Dieses Element gibt das Datum – mindestens die Jahreszahl – der Ersterscheinung der veröffentlichenden Zeitschrift an [Vgl. NLM10f].
- **Title** Das Title-Element beinhaltet den vollständigen Namen der Zeitschrift, in der die jeweilige Publikation erschienen ist [Vgl. NLM10d, S. 9].
- **ArticleTitle** ArticleTitle – der Titel einer Publikation – ist stets in englischer Sprache. Bei Veröffentlichungen in einer anderen Sprache wird der Titel ins Englische übersetzt und die Übersetzung in eckige Klammern gesetzt. Befindet sich eine Publikation noch im Qualitätssicherungsprozess und ist noch keine englische Variante des Titels vorhanden, wird unter ArticleTitle „[In Process Citation]" eingetragen [Vgl. NLM10d, S. 10].
- **MedlinePgn** Die Angabe für „MedlinePgn" kann mit dem Element „Pagination", welches die Seitenangaben für die Publikation beinhaltet, gemacht werden. Seit 2008 ist die Nutzung von „ELocationID" möglich. Dieses Element enthält einen Document Object Identifier (DOI) bzw. einen Publisher Item Identifier (PII) für die eindeutige Zuordnung [Vgl. NLM10d, S. 10f].
- **Language** In diesem Element wird die Sprache, in der die Publikation verfasst wurde, als Code aus drei Buchstaben angegeben [Vgl. NLM10d, S. 17]. Der

[5] In einigen älteren Einträgen fehlt IssnType.

Anteil von Zitaten von Publikationen in englischer Sprache steigt kontinuierlich. Mehr als 90%[6] der zwischen 2005 und 2009 veröffentlichten und in PubMed gelisteten Publikationen sind auf englisch [Vgl. NLM10e].

PublicationTypeList Innerhalb des Elements „PublicationTypeList“ sind – in alphabetischer Reihenfolge – beliebig viele Elemente „PublicationType“ vorhanden. Sie geben den Typ der Publikation – beispielsweise „Review“ oder „Journal Article“ – an [Vgl. NLM10d, S. 22].

MedlineJournalInfo Mindestens sind die nachfolgenden Elemente in „MedlineJournalInfo“ enthalten:

Country Dieses Element beinhaltet die Angabe des Landes, in dem die veröffentlichende Zeitschrift erschienen ist.

MedlineTA „MedlineTA“ ist die offizielle Abkürzung des unter „Title“ angegebenen Zeitschriftentitels.

NlmUniqueID „NlmUniqueID“ ist ein NLM-spezifischer, eindeutiger Identifier.

CitationSubset Dieses Element gibt an, welcher Teilmenge von Zitaten das vorliegende zugeordnet wird. „OM“ wird beispielsweise für „OLDMEDLINE“ verwendet [Vgl. NLM10d, S. 3].

Überdies können Publikationen durch viele weitere Elemente ergänzt werden. Alle verfügbaren Element und deren Attribute sind beispielsweise bei [NLM10a] aufgelistet. Insbesondere sind folgende Kindelemente von Article für den weiteren Verlauf dieser Arbeit von Interesse.

Abstract Dieses Attribut beinhaltet eine kurze Zusammenfassung des Inhaltes der Publikationen. Dieser Text wird mit dem Titel und den MeSH für die Klassifikation verwendet.

Authors Alle Autoren einer Publikation können gelistet werden. Nachname und Initialen werden in der Regel angegeben. Der Autor, der den größten Anteil an der Fertigstellung hatte, wird meist zuerst genannt. Der zuletzt genannte Autor ist meist Leiter der Abteilung, des Instituts oder der Organisation, in der die Mittel zur Durchführung der Forschungsarbeit bereitgestellt wurden. Jeder Autor könnte ein neuer potentieller Kunde sein. Das Auslesen und Abspeichern der Namen der Autoren ist erforderlich. Das aus der Adressdatenbank bekannte Problem der Namensdopplungen besteht auch in PubMed. Eine eindeutige Identifikationsnummer o. Ä. existiert nicht.

Affiliation Affiliation beinhaltet die Informationen über die Organisation, für die die Autoren tätig sind. Üblich sind die Angabe der Adresse des erst- oder letztgenannten Autors. Bei Autoren aus verschiedenen Organisationen werden die entsprechenden Informationen häufig für jeden Autor einzeln aufgeführt. Die Ablage der Affiliation ist für die Verwendung im Direktmarketing hilfreich.

Schließlich sind einige Publikationen um ein Kindelement von MedlineCitation erweitert: „MeshHeadingList“ beinhaltet eine Liste von „MeshHeading“-Elementen. **Medical Subject**

[6] Anteil basiert auf einer Erhebung vom 2. April 2010, d. h. das Jahr 2009 ist nicht vollständig erfasst.

Headings (MeSH) ist ein kontrollierter Thesaurus, der eine Hierarchie von Begriffen aus den Lebenswissenschaften beinhaltet [Vgl. NIH10]. Der Inhalt von Publikationen wird durch die Angabe von MeSH stichwortartig beschrieben. Jeder Publikation können beliebig viele MeSH zugeordnet werden. Ein MeSH-Begriff, der nur relevanten Publikationen zugeordnet ist, ist „Electrophoresis, Gel, Two-Dimensional“. MeSH hat direkten Inhaltsbezug und wird folglich – ebenso wie Abstract und Title – für die Klassifikation verwendet.

4.2.3. Datenqualität

Alle in PubMed veröffentlichten Publikationen durchlaufen die Qualitätskontrolle[7] der National Library of Medicine [Vgl. NCB10a, S. 45]. Während dieser Kontrolle werden nicht nur typografische Fehler behoben. Auch die Vollständigkeit der Pflichtattribute wird überprüft. Veröffentlichungen, die nicht in englisch verfasst sind, erhalten einen in eckige Klammern gesetzten englisch-sprachigen Titel [Vgl. NLM10d, S. 10]. Dank dieser Datenkontrolle ist die Qualität der Pflichtattribute von PubMed-Einträgen sehr hoch. Fehler oder gar fehlende Werte konnten während der in Abschnitt 5.1 erläuterten Datenselektion nicht entdeckt werden.
Größere Defizite sind bei den fakultativen Attributen zu finden. Tabelle 4.1 gibt exemplarisch an, wie häufig die Attribute MeshTerms, Abstract, Affiliation und Authors in den 3.180 Publikationen der Instanzmenge (Vgl. 5.1) angegeben wurden.

MeSH	Abstract	Affiliation	Authors
659	3.162	2.976	3.142

Tabelle 4.1. Vorkommen von Publikationsattributen in der für die Evaluierung der Modellbildung verwendeten Instanzmenge.

Über 90 Prozent der Autoren wurden mit vollständigem Vor- und Nachnamen sowie – wenn vorhanden – der Initiale des zweiten Vornamen angegeben. Jeder Abstract enthielt im Mittel 128 einzigartige Wörtern (Vgl. Kapitel 5) mit einer Standardabweichung von rd. 36. Damit ist für das Text Mining eine gute Grundlage gegeben: Die Abstracts sind für nahezu alle Publikationen verfügbar und die durchschnittliche Länge lässt eine relativ gute Zusammenfassung des Themas vermuten.
Rund 21 Prozent der Publikationen wurden MeSH-Begriffe zugeordnet. Im Mittel sind dies zwölf Begriffe (Standardabweichung rd. 4). Autoren und veröffentlichende Zeitschriften scheinen ein stärkeres Interesse daran zu haben, die Autorennamen und deren Organisationszugehörigkeit mit der Publikation verbunden zu wissen. Die Bereitschaft, zusätzlichen Aufwand in die Vergabe von MeSH-Begriffen zu investieren, scheint jedoch deutlich geringer. Eine kurze Zusammenfassung (Abstract) ist für viele Publikationen – da von den Zeitschriften gefordert – ohnehin vorhanden und ist daher oft ohne Mehraufwand angebbar.

[7] Trotz mehrerer Versuche zur Kontaktaufnahme mit der NLM konnte die Autorin den exakten Ablauf des Prozesses nicht in Erfahrung bringen.

5. Datenvorverarbeitung

Beim Data Mining im engeren Sinne sind die Behandlung von Ausreißern, fehlenden und inkonsistenten Werten Hauptaufgaben der Datenvorverarbeitung [Vgl. TSK06, S. 40ff]. Beim Web und Text Mining ist überdies die Extraktion der zu analysierenden Texte wichtig für den Erfolg des Mining [Vgl. FS07, S. 59]. Die Anforderungen unterscheiden sich folglich bezüglich des Anwendungsbereiches. Gemeinsam ist allen Kategorien des Data Mining unter anderem, dass – aufgrund der riesigen Datenmengen – die Auswahl einer Teilmenge aus den verfügbaren Daten für Evaluierungszwecke erforderlich ist.
Im ersten Abschnitt dieses Kapitels wird erläutert, welche der über 20 Millionen PubMed-Publikationen für die Modellbildung verwendet wurden. Extraktion und weitere Vorbereitung sind Thema des zweiten Teils. Der Abschnitt 5.3 stellt Verfahren zur Dimensionsreduktion für Texte vor. In welchem Umfang diese Verfahren in der vorliegenden Arbeit zum Einsatz gekommen sind, wird ebenfalls erläutert. Der letzte Teil dieses Kapitels behandelt die Konvertierung der PubMed-Publikationen in das ARFF-Format, welches in der Webanwendung verwendet wurde.

5.1. Datenselektion

5.1.1. Auswahl der Instanzmenge

Für die Bewertung der Ergebnisse der „Modellbildung“ (Vgl. Kapitel 6) ist eine Instanzmenge erforderlich. Die Klassenzugehörigkeiten für deren Elemente müssen bekannt sein. Soweit bekannt existiert keine Quelle, in der Publikationen in den Klassen „beinhaltet Ausführungen über Arbeiten mit zweidimensionaler Gelelektrophorese“ – im Folgenden als Klasse **„2DGE“** bezeichnet – und restliche PubMed-Einträge – Klasse **„No2DGE“** – vorliegen. Der Aufbau einer Instanzmenge mit manueller Klassifikation der Elemente ist also erforderlich.
Aufgrund des immensen zeitlichen Aufwandes für die manuelle Klassifikation wurden zunächst der Autorin bekannte Publikationen beider Klassen gesammelt. Bei einer Anzahl von 327 Publikationen wurde der Aufbau der Instanzmenge angehalten, um erste Erkenntnisse über das Verhalten der Klassifikation und für die Erweiterung der Instanzmenge zu gewinnen. Die beiden Klassen waren in dieser ersten Instanzmenge etwa gleich häufig vertreten (185 „2DGE“, 142 „No2DGE“). Eine Eigenimplementation des k-Nearest-Neighbors-Klassifikationsverfahren (Vgl. 6.1.1) wurde zur Evaluierung in einer dreifachen Kreuzvalidierung[1]. Mit einer maximalen Erfolgsrate von rd. 0,6 waren die Ergebnisse der dreifachen

[1] Entgegen der Erläuterungen in Abschnitt 2.5.2, wurde aufgrund der geringen Anzahl von Dokumenten $n = 3$ für die Kreuzvalidierung gewählt.

Kreuzvalidierung wenig überzeugend. Das Ergebnis ist nur unwesentlich besser als eine zufällige Klassenzuordnung. Durch Anwendung einer vierfachen Kreuzvalidierung wurde die Anzahl der Elemente in der Trainingsmenge – bei unveränderter Instanzmenge – erhöht. Die Erfolgsrate konnte auf rd. 0,7 verbessert werden. Abbildung 5.1 zeigt die Ergebnisse beider Kreuzvalidierungen bei Variation des k-Parameters.

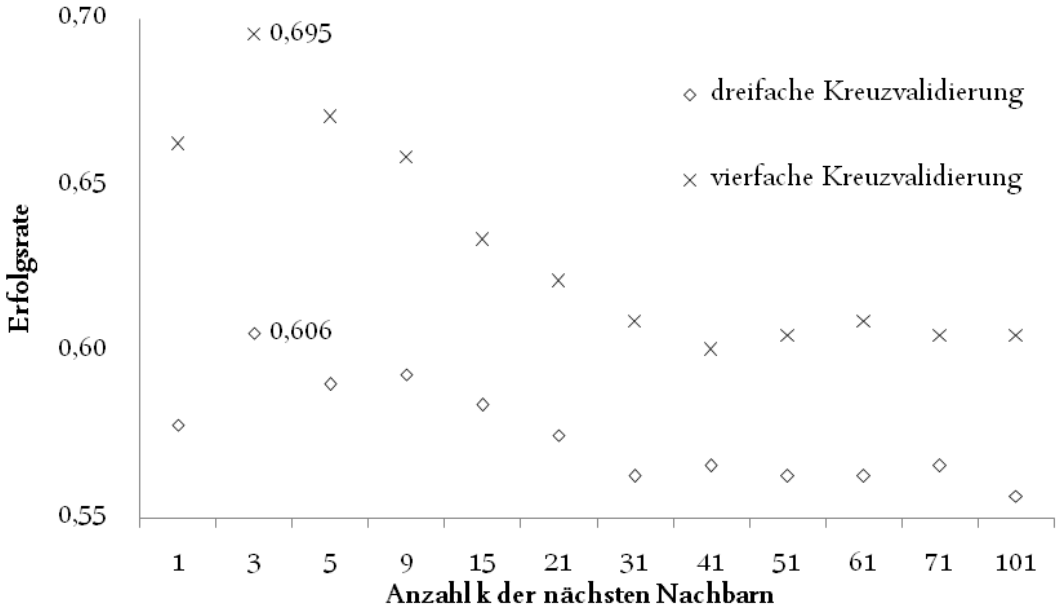

Abbildung 5.1. Ergebnisse der Klassifikation mit k-Nearest-Neighbors für 327 Publikationen, Trainingsstrategie: vierfache Kreuzvalidierung (Eigene Darstellung).

Zwei Erkenntnisse konnten aus diesen ersten Tests gewonnen werden:

1. **Die Erweiterung der Trainingsmenge kann die Resultate der Klassifikation verbessern.**

   Die deutliche Verbesserung der Erfolgsrate bei Wechsel von drei- auf vierfache Kreuzvalidierung lässt einen positiven Trend vermuten. Diese positive Auswirkung durch Vergrößerung der Trainingsmenge wurde bereits in anderen Arbeiten beobachtet [Vgl. For07, S. 21].

2. **Die Klassenverteilung in der Instanzmenge sollte der realen Verteilung angepasst werden.**

   Beim Aufbau der ersten Instanzmenge wurde die Klassenverteilung vernachlässigt. Lediglich das Vorhandensein von ausreichend Exemplaren aus jeder Klasse wurde sichergestellt. Die Aussagekraft bezüglich der Performanz der Klassifikation auf realen Daten ist gering.

Die tatsächliche Klassenverteilung wurde mittels Untersuchung der jeweils ersten 20 Publikationen – sortiert nach CreationDate – der Monate Januar 2009 bis einschließlich Juni 2010

ermittelt. Neben einer Suche ohne weitere Einschränkung wurden Suchen mit den Suchbegriffen „electrophoresis“, „protein“, „DIGE“ und „2-DE“ durchgeführt. Tabelle 5.1 zeigt den Anteil der Publikationen mit 2DGE in Prozent.

Suchbegriff	*keiner*	electrophoresis	protein	DIGE	2-DE
Summe	0,00%	13,89%	1,67%	96,81%	97,35%

Tabelle 5.1. Auswirkung von Suchbegriffen auf den Anteil positiver Dokumente.

Wurde die Suche nicht mit Suchbegriffen (Spalte „*keiner*“) eingeschränkt, konnten keine Publikationen der Klasse 2DGE ermittelt werden. Der Anteil relevanter Publikationen unter allen PubMed-Einträgen scheint zu gering zu sein. Die Einschränkung der Suche ist erforderlich, um die ressourcenaufwändige Speicherung, Verarbeitung und Klassifikation von unnötig vielen irrelevanten Publikationen in der Webapplikation (vgl. Kapitel 8) zu vermeiden. „2-DE“ und „DIGE“ – übliche Abkürzungen für die zweidimensionale Gelelektrophorese respektive eine Variante dieser Methode – führten hingegen fast ausschließlich zu Treffern der Klasse 2DGE. Tabelle 5.2 zeigt die unter Nutzung der Prozentzahlen aus Tabelle 5.1 hochgerechneten Anzahlen von relevanten Publikationen im Zeitraum Januar 2009 bis Juni 2010.

Suchbegriff	Gesamt	Anzahl positiver Dokumente
electrophoresis	13.595	1.888
protein	312.676	5.211
2-DE	605	586
DIGE	304	296

Tabelle 5.2. Vermutete Anzahl positiver Dokumente Januar 2009 bis Juni 2010 bei Einschränkung durch Suchbegriffe.

Bei Verwendung des Suchbegriffes „protein“ können absolut die meisten positiven Publikationen gefunden werden. Relativ sind die positiven Instanzen unterrepräsentiert. Weitere Suchen mit Kombination oder Ausschluss der genannten Suchbegriffe wurden durchgeführt. Abbildung 5.2 stellt die ermittelte Verteilung schematisch dar. Graue Bereiche symbolisieren relevante Publikationen, weiße irrelevante. Für Bereiche außerhalb der Ellipsen können keine Aussagen getroffen werden. Die Größen der Ellipsen spiegeln nur annähernd die tatsächlichen Relationen der Trefferanzahlen wider.
Die Suchen haben gezeigt, dass der Begriff „electrophoresis“ als Suchbegriff geeignet ist: Im Gegensatz zu den Treffermengen aller anderen Suchbegriffe werden absolut und relativ viele relevante Publikationen gefunden. Nur einige wenige relevante Publikationen konnten ermittelt werden, die bei der Suche mit diesem Suchbegriff nicht in der Trefferliste sind. Für die Erweiterung der Instanzmenge wird „electrophoresis“ zur Einschränkung der Suche verwendet. Ebenso wird dieser Begriff für die Erweiterung der Datenbasis in der produktiven Anwendung (Vgl. Kapitel 8) empfohlen.
Schließlich wurden für den Aufbau der Instanzmenge folgende Überlegungen berücksichtigt: Die zunehmende Verwendung alternativer Methoden in der Proteinanalytik könnte zu

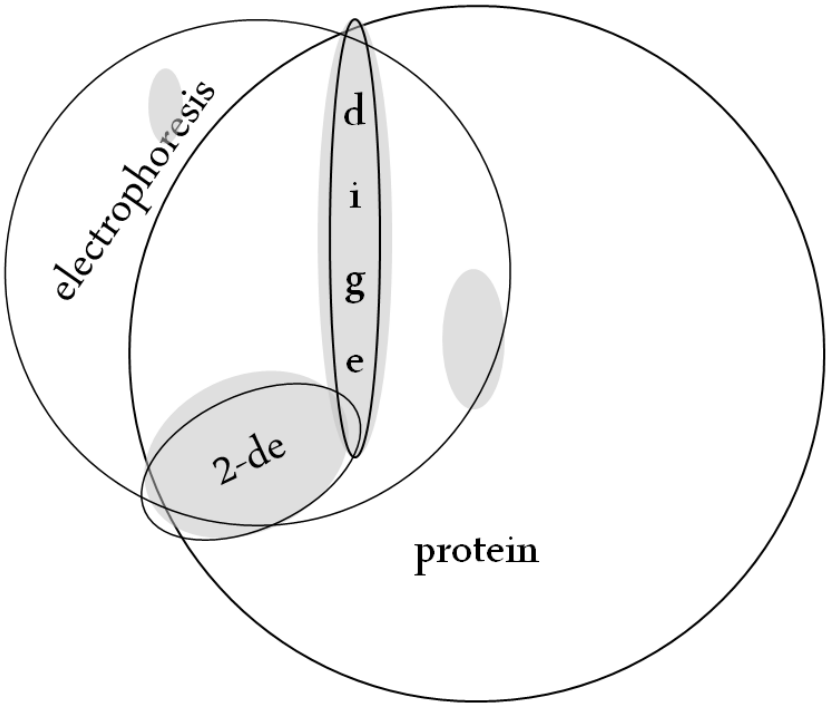

Abbildung 5.2. Verteilung der relevanten Publikationen bei Einschränkung der PubMed-Suche mit unterschiedlichen Suchbegriffen (Eigene Darstellung).

einer Verringerung des Anteils relevanter Publikationen führen. Die Methode zweidimensionale Gelelektrophorese könnte in einer Arbeit über alternative Techniken – wie beispielsweise die von Venugopal et al. [Vgl. VCP09, S. 3] – nur genannt, nicht aber verwendet werden. Derartige Publikationen sind irrelevant und führen zu einer Verringerung des relativen Anteils von relevanten Einträgen in der Trefferliste für „electrophoresis“. Zudem ist in der Webanwendung (Vgl. Kapitel 8) die Möglichkeit der Erweiterung des Datenbestandes durch Suche weiterer Publikationen eines Autors vorgesehen. Diese Vorgehensweise könnte den Anteil zusätzlich irrelevanter Publikationen erhöhen (Vgl. Abschnitt 5.1.2). Für die Instanzmenge wurde daher der Anteil etwas niedriger gewählt als durch die PubMed-Suche ermittelt.
Die resultierende Instanzmenge besteht aus 3.180 manuell klassifizierten Publikationen:

$$\underbrace{\text{Instanzmenge}}_{3.180\ (100\%)} = \underbrace{\text{Dokumente Klasse „2DGE“}}_{390\ (rd.\ 12\%)} + \underbrace{\text{Dokumente Klasse „No2DGE“}}_{2.790\ (rd.\ 88\%)}$$

5.1.2. Auswahl der verwendeten Attribute

Die Attribute von Publikationen können in diejenigen

- ohne direkten Inhaltsbezug – wie „DateCreated“ oder „PubMed“ – und
- mit direktem Inhaltsbezug – wie „Title“ oder „Abstract“ –

unterschieden werden. Um für die Klassifikation verwendbare Attribute von Publikationen zu ermitteln, klassifizierten Denecke et al. technische und naturwissenschaftliche Publikationen aus der Technischen Informationsbibliothek (TIB) Hannover. Sie konnten beobachten, dass Textklassifikation basierend auf den inhaltsbezogenen Attributen Titel (PubMed-Attribut „ArticleTitle“) und Zusammenfassung („Abstract“) gute Ergebnisse liefert, wenn der gesamte Text nicht zur Verfügung steht [Vgl. DRB10, S. 3]. Ähnlich wie Titel und Zusammenfassung weisen MeSH-Begriffe direkten Inhaltsbezug auf. Diese drei Attribute wurden für die Modellbildung verwendet.
Sollten zusätzlich die Autorennamen zur Klassifikation verwendet werden? In einer Reihe von Arbeiten wurde gezeigt, dass Autoren anhand der von ihnen verfassten Texte identifiziert werden können [Vgl. Seb01, S. 13]. Sogar die Muttersprache des Autors kann mit einer Erfolgsrate von über 0,8 ermittelt werden [Vgl. KSZ05]. Könnten dann nicht andersherum Autoren Hinweis auf den Inhalt der Texte geben? Aus folgenden Gründen wird dies für hinreichend unwahrscheinlich gehalten und auf die Verwendung der Autorennamen verzichtet:

- **Textanalysen zur Identifikation der Autoren konzentrieren sich auf den Stil der Arbeit.** Können Worte oder Wortgruppen identifiziert werden, die besonders häufig auftreten? Welche Satzkonstruktionen verwendet der Autor bevorzugt? Wie wurde der Text durch Absätze, Einbindung von Abbildungen und weitere Elemente strukturiert? Der Inhalt ist bei diesem Ansatz zumeist irrelevant.

- **Wissenschaftler versuchen, ein Forschungsthema möglichst umfassend zu bearbeiten.** Dies erfordert den Einsatz unterschiedlichster Methoden. Die Erkenntnisse aus der zweidimensionalen Gelelektrophorese können beispielsweise durch die Anwendung komplementärer Techniken erweitert werden [Vgl. VCP09, S. 8f]. Namen von auf diese Weise forschenden Wissenschaftlern werden weniger Indiz für eine Methode als vielmehr für einen Forschungsbereich sein.

- **Namensdopplungen mindern den Informationsgewinn.** In Abschnitt 4.1 wurde bereits auf die Problematik gleicher Namen unterschiedlicher Personen hingewiesen. Der Greifswalder Professor Michael Hecker ist Autor zahlreicher Publikationen der Klasse „2DGE“. Sein Namensvetter aus Jena verwendet diese Methode nicht, veröffentlicht allerdings in den letzten Jahren mit zunehmender Häufigkeit. Die Anzahl der neuerscheinenden Publikationen vom erstgenannten Autor werden – aufgrund des baldigen Eintritts in den Ruhestand – zurückgehen. Mittelfristig wird der Name „Michael Hecker“ – bei Betrachtung aktueller Publikationen – dann eher Indiz für die Klasse „No2DGE“ sein.

- Die verwendete Datenquelle enthält sehr viele Publikationen. Nur ein geringer Anteil wird klassifiziert. **Publikationen desselben Autors treten selten auf.** Denecke et al. konnten zeigen, dass bei einem derartigen Ansatz die Nutzung der Autorennamen keinen positiven Einfluss auf die Klassifikation hat [Vgl. DRB10, S. 3].

In Abschnitt 4.2.1 wurde darauf hingewiesen, dass für einige Publikationen ein Link zu deren **Volltext** verfügbar ist. Warum werden die Volltexte nicht verwendet? Sie könnten für die Klassifikation nützlich sein. Allerdings weist die Extraktion einige Schwierigkeiten auf. Einerseits

unterscheidet sich die „Entfernung“ des PubMed-Links vom Volltext: Oft müssen einige weitere Links verfolgt werden, um den Volltext zu erreichen. Je nach publizierender Zeitschrift muss ein anderer Pfad verfolgt werden. Andererseits veröffentlichen Zeitschriften Publikationen in diversen Formaten, aus denen der Volltext für die weitere Verwendung extrahiert werden muss. Darüberhinaus ist der Volltext für die meisten Publikationen nicht frei verfügbar: Von den 10.231 Publikationen, im Jahr 2008 veröffentlichten Treffer für den Suchbegriff „electrophoresis“ haben nur rd. 26% einen kostenfrei abrufbaren Volltext. In 2009 trifft dies (Stand: November 2010) nur noch auf rd. 24% (von 10.026 veröffentlichten Publikationen) zu. Im Zeitraum Januar bis einschließlich Oktober 2010 sind es nur noch rd. 14%. Insbesondere für die neueren – für die Zielstellung interessanteren – Publikationen sind demnach wenige Volltexte verfügbar. Im Rahmen dieser Arbeit wird daher auf die Volltexte verzichtet.

5.2. Vorbereitung und Reinigung der Daten

Eine zentrale Aufgabe der Datenvorbereitung beim Data Mining im engeren Sinne ist die Behandlung von fehlenden Werten [Vgl. Cle10, S. 55]. In Abschnitt 4.2.3 wurde bereits auf einige relevante Attribute von Publikationen hingewiesen, die teilweise nicht mit Werten gefüllt sind. Eine Möglichkeit ist die manuelle Ergänzung. Dieser Prozess wäre mit extrem hohem zeitlichen Aufwand verbunden: Jede Publikation müsste gelesen und z. B. dem Inhalt entsprechende MeSH-Begriffe ergänzt werden. Die „Affiliation“ ist schwer bis gar nicht ermittelbar – insbesondere wenn Autorennamen fehlen. Alternativ ist die automatisierte Zuordnung von MeSH-Begriffen denkbar. Ähnlichen Publikationen würden gleiche MeSH-Begriffe zugeordnet werden. Beide Varianten entsprechen der – manuellen bzw. automatischen – Lösung der in dieser Arbeit gestellten Klassifikationsaufgabe. Daher wird eine häufig angewandte Methode zur Behandlung der fehlenden Werte verwendet: Sie werden ignoriert.
Neben der Behandlung fehlender Werte umfasst die Datenvorverarbeitung viele weitere Aufgaben. Für das Text Mining kann in vorbereitende und linguistische Techniken unterschieden werden. Abbildung 5.3 zeigt diese beiden Aufgabenbereiche.

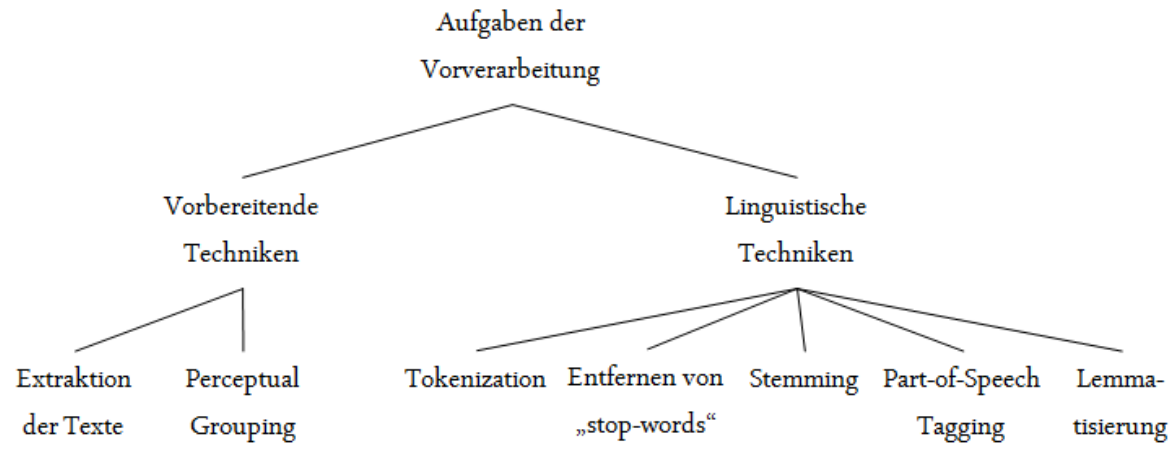

Abbildung 5.3. Bereiche der Vorverarbeitung (In Anlehnung an [FS07, S. 59]).

Zu den vorbereitenden Techniken gehört unter anderem das Perceptual Grouping. Diese Methode hat die Einteilung der Dokumente in Titel und Kapitel zum Ziel. Das XML-Format von PubMed beinhaltet bereits eine für die Klassifikation in dieser Arbeit ausreichende Unterteilung. Daher wird im folgenden Abschnitt lediglich die angewandte vorbereitende Technik der Extraktion erläutert. Die in Abbildung 5.3 genannten linguistischen Techniken sind Gegenstand von Abschnitt 5.2.2.

5.2.1. Extraktion der relevanten Daten

Ein umfangreicher Teil der Datenvorbereitung im Text Mining ist die Extraktion der Texte aus unterschiedlichen Formaten. Bei eingescannten Dokumenten kann beispielsweise die Optical Character Recognition (OCR) verwendet werden, um Texte auszulesen [Vgl. FS07, S. 59].
In Abschnitt 4.2.1 wurde bereits auf die Vorteile der verwendeten XML-Dokumente hingewiesen. Die Extraktion der Texte kann mit vergleichsweise geringem Aufwand realisiert werden:

1. **Dekodierung**

   Mindestens ist die Dekodierung der Tag[2]-umschließenden spitzen Klammern von „<“ zu „<“ und von „>“ zu „>“ erforderlich. Zusätzlich sollten Symbole wie „&“ (im XML-Dokument „&“) dekodiert werden [Vgl. HS09, S. 20].

2. **Extraktion der Inhalte relevanter Elemente**

   Die Webanwendung (Vgl. Kapitel 8) wird in Python programmiert. Für die Extraktion der Elementinhalte wird das freie Python-Paket „BeautifulSoup“ verwendet: Das XML-Dokument wird in ein Objekt der Klasse „BeautifulStoneSoup“ eingelesen. Anschließend können beispielsweise Autoren-Elemente bei diesem Objekt (im Beispiel „soup“ genannt) abgefragt und deren Kindelemente „forename“ ausgelesen werden:

   ```
   for author in soup.findAll('author'):
       forename = author.forename.renderContents()
   ```

5.2.2. Linguistische Techniken

Die Mehrzahl der im Text Mining verwendeten linguistischen Techniken entstammt dem Natural Language Processing (NLP). Forschungsgegenstand des NLP sind die natürlichen Sprachen. Ziel ist die Entwicklung von Techniken zur automatisierten Verarbeitung von Texten. Eine grundlegende Aufgabe des Natural Language Processing ist die Zerlegung von Texten in durch Leerzeichen voneinander getrennte Worte, Zahlen und weitere Zeichenketten. Dieser Prozess wird als **Tokenization**, die Zeichenketten als Token bezeichnet. Tokenization beinhaltet die Entfernung von Satzzeichen wie Punkt, Komma, Semikolon, Frage- oder Ausrufezeichen [Vgl. HS09, S. 20]. Abbildung 5.4 zeigt ein einfaches Beispiel für einen Satz („Eingabe“) und die resultierenden „Token“ („Ausgabe“).

[2] Jedes Element in einem XML-Dokument wird durch einen Starttag im Format „<TAGNAME>“ und einen schließenden Tag „</TAGNAME>“ begrenzt. Ausnahme stellen lediglich leere Elemente dar. Sie werden nur mit einem Tag „<TAGNAME />“ gekennzeichnet.

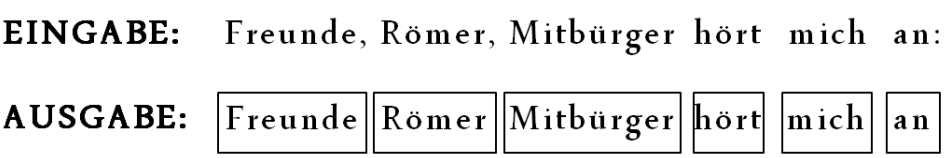

Abbildung 5.4. Beispiel für Tokenization (Quelle (aus dem Englischen übersetzt): [HS09, S. 22]).

Jeder entstehende Token wird als „Feature“ bezeichnet. Die Menge aller Features, die aus einem Dokument extrahiert wurden, als „bag of words“ [Vgl. FS07, S. 68].
Eine übliche Weiterverarbeitung der Features ist die **Entfernung der Großschreibung** [Vgl. For07, S. 6], [Vgl. HS09, S. 30]. In Texten in englischer Sprache werden Wörter fast ausschließlich dann groß geschrieben, wenn sie am Satzanfang stehen. Um zu verhindern, dass gleiche Worte nur aufgrund ihrer Stellung im Satz zu unterschiedlichen Features werden, wurde die Großschreibung in dieser Arbeit entfernt.
Jedes Feature kann mittels **Stemming** weiter verarbeitet werden. Bei diesem Verfahren werden Suffixe entfernt, um Wörter gleicher oder ähnlicher Bedeutung in einer Stammform zusammenzuführen. Die Stammform muss kein existierendes Wort sein. Ein weit verbreitetes Verfahren für die englische Sprache ist der Porter-Stemming-Algorithmus [Vgl. HNP05, S. 29]. Das Verfahren läuft in fünf Schritten ab, in denen systematisch Suffixe entfernt werden [Vgl. Por80]. Toman et al. kamen in ihrer Arbeit über Lemmatisierungs- und Stemming-Algorithmen zu dem Ergebnis, dass der Stemming-Algorithmus von Porter die Performanz nur unwesentlich beeinflusst [Vgl. MTJ06, S. 5]. Gleichzeitig führte die Anwendung auf insgesamt 8.000 Dokumente aus der Reuters-Sammlung zu einer signifikanten Verringerung der Features. Da die Reduktion der Features positiven Einfluss auf die Performanz hat, wurde der Porter-Stemming-Algorithmus während der Evaluierung der Modellbildung getestet (Vgl. Abschnitt 6.2).
Eine weitere Methode zur Bearbeitung von Tokens ist die **Lemmatisierung**. Voll- und Grundformenlexika werden verwendet, um Wörter in ihre Grundform (Lemma) zu überführen. Erstere ermöglichen das Nachschlagen jedes Wortes zur Ermittlung der Lemmata. Für letztere wird zunächst jedem Token – mithilfe des **Part-of-Speech Tagging (PoS Tagging)** – dessen Wortart zugeordnet. Anschließend werden z.B. Substantive in den Singular und Verben in den Infinitiv überführt. Die Existenz des ermittelten Lemmas wird schließlich – unter Nutzung des Grundformenlexikons – verifiziert [Vgl. OKO07, S. 7]. Die Lemmatisierung ist im Vergleich zum Stemming deutlich ressourcenaufwändiger. Die Ergebnisse sind – zumindest gemäß aktuellem Stand der Forschung – stark fehlerbehaftet, so dass sich der Mehraufwand (noch) nicht lohnt [HNP05]. Lediglich für morphologisch komplexere Sprachen ist die Anwendung empfehlenswert [Vgl. MTJ06, S. 5]. Lemmatisierung wurde daher im Rahmen dieser Arbeit nicht angewandt.
Ein weiteres Verfahren ist das **Entfernen von „stop words“** aus jedem bag of words. Stop words sind – unabhängig von der Klasse der Dokumente – häufig vorkommende Wörter. Sie liefern keinen zusätzlichen Informationsgewinn und können herausgefiltert werden [Vgl. HNP05, S. 25f]. Das Verfahren ist im weitesten Sinne eine linguistische Technik, gehört aber nicht zum Natural Language Processing. In einigen Quellen wird das Entfernen von stop

words den in Abschnitt 5.3 erläuterten Feature-Auswahlverfahren zugeordnet. Grund hierfür ist, dass dieses Verfahren zur Dimensionsreduktion führt [Vgl. FS07, S. 68]. Allerdings trifft dies beispielsweise auch auf das Entfernen der Großschreibung zu: Aus den zwei Features „Protein“ und „protein“ wird ein Feature „protein“. Der entscheidende Unterschied ist, dass die stop words unabhängig von der vorliegenden Instanzmenge entfernt werden. Feature-Auswahlverfahren hingegen selektieren Features basierend auf Maßen, die aus der Instanzmenge berechnet wurden (Vgl. 5.3). Die Entfernung der stop words erfolgt unabhängig von den vorliegenden Dokumenten. Die verwendeten Listen von stop words unterscheiden sich lediglich für verschiedene Datenquellen. Für die PubMed-Datenbank stellt die National Library of Medicine eine Liste von stop words zur Verfügung (Vgl. Anhang A.1). Diese Liste wurde im Rahmen dieser Arbeit verwendet.
Neben den genannten Techniken werden beispielsweise bei [TS99] und [FS07] weitere beschrieben. Sie werden selten angewendet, weil sie weniger ausgereift als die vorgestellten Verfahren sind [Vgl. FS07, S. 60f]. Aus letztgenanntem Grund werden diese Methoden in dieser Arbeit weder erläutert noch angewandt.
Für die Vorverarbeitung wurde folgendes Vorgehen gewählt:

1. Titel und Zusammenfassung jeder Publikation wurden mittels Tokenization in einen „bag of words“ transformiert, die Großschreibung wurde entfernt.

2. Der Einfluss der Entfernung von stop words und des Stemmings sollte evaluiert werden. Daher wurden in diesem zweiten Schritt vier alternative Vorgehensweisen gewählt:
   - Weder Entfernung von stop words noch Stemming.
   - Nur Entfernung von stop words.
   - Nur Stemming.
   - Entfernung von stop words und Stemming.

3. MeSH-Begriffe wurden unverändert übernommen, um deren Bedeutung beizubehalten.

Die Auswirkung der Anwendung von stop-words-Entfernung respektive Stemming auf 800, 600, 400 und 200 zufällig gewählte Dokumente aus der Instanzmenge zeigt Abbildung 5.5.
Auffällig ist der geringe Einfluss der stop-words-Entfernung sowie der starke Einfluss des Stemmings auf die Anzahl der Features: In 800 zufällig ausgewählten Publikationen sind ohne weitere Bearbeitung 21.543 Features vorhanden. Wird Stemming angewandt, sinkt die Anzahl auf 17.845. Dies entspricht einer Reduktion von mehr als 17 Prozent. Die Abbildung zeigt, dass die relative Reduktion bei allen Dokumentanzahlen nahezu identisch ist.
Nach der Datenvorbereitung werden alle einzigartigen Features ermittelt. Jede Publikation kann dann als Vektor der Feature-Anzahlen dargestellt werden. Die meisten Features kommen nicht in allen Publikationen vor. Daher wird das Ergebnis für jede Veröffentlichung ein **hochdimensionaler, spärlich-besetzter Vektor** sein. Werden alle Publikationen zusammengefasst, entsteht eine **hochdimensionale, dünn-besetzte Datenmatrix** [Vgl. TSK06, S. 31].

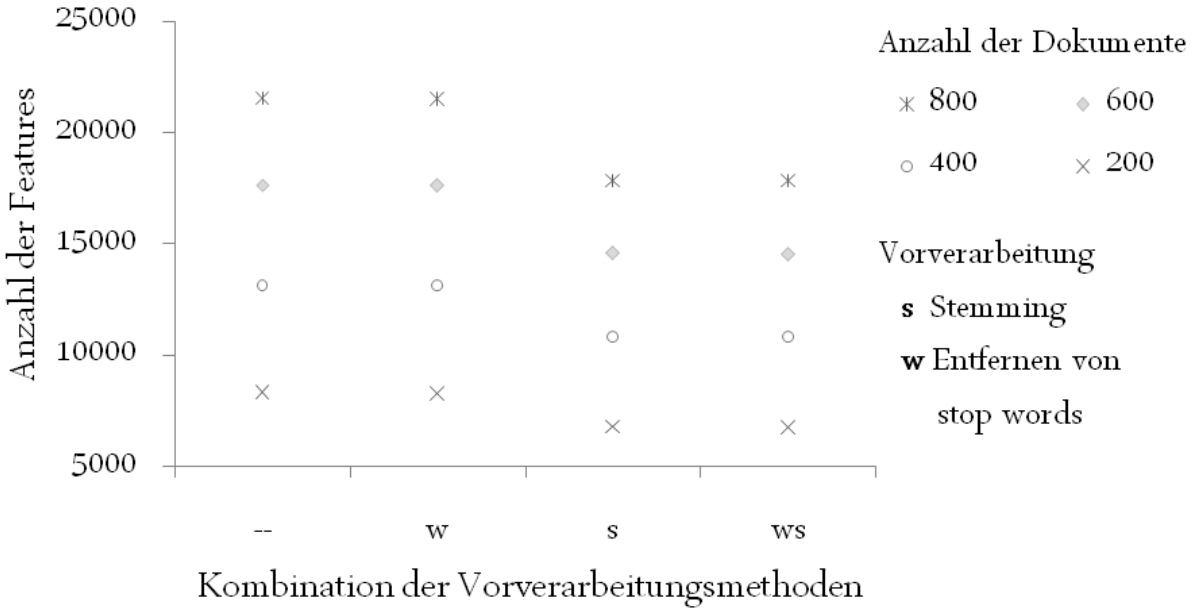

Abbildung 5.5. Einfluss der Anwendung von Vorverarbeitungsmethoden auf die Anzahl der Features (Eigene Darstellung).

5.3. Dimensionsreduktion mittels Feature-Selektion

Dimensionsreduktion im Bereich des Text Mining hat die Verringerung der Anzahl der Features zum Ziel. Dies entspricht der Reduktion der Dimensionen der Publikationsvektoren. Einerseits wird dadurch der Ressourcenbedarf für die Klassifizierung möglichst gering gehalten [Vgl. KHP05, S. 50]. Andererseits wird eine Verminderung der Gefahr von Overfitting erzielt [Seb01, S. 16]: Einige Verfahren optimieren ihre Ergebnisse für die Trainingsmenge, d. h. sie streben die fehlerfreie Performanz auf diesen Daten an. Dazu werden alle Features verwendet (Vgl. Abschnitt 6.1.2). Je mehr Features vorhanden sind, desto spezialisierter kann das aufgebaute Modell werden. Werden diesem Modell neue Daten präsentiert, sind die Ergebnisse meist signifikant schlechter als das Verhalten bzgl. der Instanzmenge suggeriert. Dieses Phänomen wird als Overfitting bezeichnet.
Zahlreiche Methoden stehen für die Reduktion der Anzahl der Features zur Auswahl. Einige werden beispielsweise bei [Seb01] genannt. Für jedes Feature wird eine Punktzahl verfahrensspezifisch berechnet. Ist diese Punktzahl größer als ein vordefinierter Schwellwert, wird das Feature weiter verwendet, andernfalls als irrelevant verworfen.
Die Vielfalt der Möglichkeiten erfordert die Vorauswahl einiger Verfahren für die Evaluierung. Ein Grundsatz im Data Mining ist „simplicity first“ [WF05, S. 83]. Daher wurde die auch als „Ad-Hoc-Ansatz“ bezeichnete Methode **Document Frequency (DF)** als erstes zu evaluierendes Verfahren gewählt. Sie wird in Abschnitt 5.3.2 erläutert.
Als Gegenbeispiel wurde ein deutlich komplexeres, im Text Mining häufig zitiertes und erfolgreich eingesetztes Maß (Vgl. [HS09], [FS07], [Vit04], [For07] und viele weitere) ausgewählt: das **Produkt aus Term Frequency und Inverse Document Frequency (TF*IDF)**. TF*IDF wird in Abschnitt 5.3.3 detailliert beschrieben.

Fraglich war, ob der Einsatz dieser Methoden zu besseren oder zumindest gleich guten Klassifikationsergebnissen führt. Aizawa konnte ohne Feature-Selektion zufriedenstellende Resultate erzielen [Vgl. PA01, S. 314]. Daher wurden alle modellbildenden Verfahren auch **ohne Dimensionsreduktion** mittels Feature-Selektion getestet.
Für den in Abschnitt 5.1.1 erläuterten Vorab-Test wurden dreizehn Features **manuell selektiert**. Die Auswahl wird in Abschnitt 5.3.1 erläutert.
Mit Ausnahme der manuellen Selektion gilt für alle angewendeten Methoden, dass MeSH-Begriffe von der Entfernung ausgenommen sind. Sie beinhalten Schlüsselworte bzw. -wortgruppen für die Einordnung der Publikationen und werden daher stets ohne Einschränkung verwendet.

5.3.1. Manuelle Selektion

Die manuell selektierte Liste umfasst Features, die – gemäß dem Kenntnisstand der Autorin – typisch für Publikationen der Klasse „2DGE“ sind:

2d-gel	2-de	2d-electrophoresis	2dge	spots
2d-dige	dige	electrophoresis	gel-based	gel
gel-electrophoresis	proteomic	two-dimensional		

Tabelle 5.3. Liste der manuell ausgewählten Features.

Bei Klassifikation mit dem k-Nearest-Neighbors-Verfahren (Vgl. 6.1.1) konnten mit dieser **manuellen Selektion** deutlich bessere Resultate erzielt werden. Abbildung 5.6 zeigt einen Überblick der Ergebnisse. Unabhängig von dem Wert des Parameters *k* konnten in einer drei-

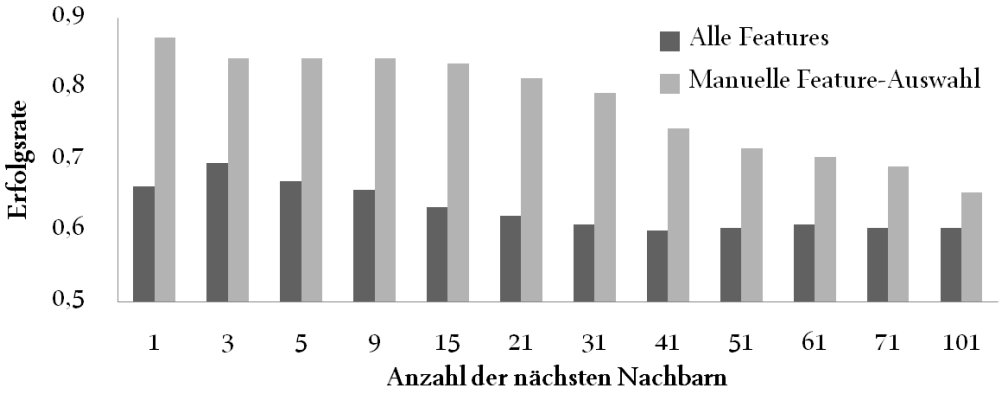

Abbildung 5.6. Erfolgsrate bei Anwendung des k-Nearest-Neighbors-Verfahrens auf 372 Publikationen – 185 der Klasse „2DGE“ und 142 der Klasse „No2DGE“ (Eigene Darstellung).

fachen Kreuzvalidierung und manueller Selektion stets bessere Erfolgsraten als ohne Dimensionsreduktion ermittelt werden. Daher wurde die manuelle Feature-Auswahl als vierte Methode evaluiert.

5.3.2. Document Frequency

Die Document Frequency gibt für jedes Feature an, in wie vielen Dokumenten der Trainingsmenge es vorkommt. Mit einem Schwellwert wird angegeben, wie oft ein Feature mindestens vorkommen muss, um berücksichtigt zu werden [Vgl. YP97, S. 413]. Die Methode führt folglich zur Entfernung von seltenen Features. Drineas et al. haben den Einfluss von Feature-Auswahlverfahren unter anderem für die Klassifikation von Reuters-Texten untersucht. Sie konnten – wie Yang und Pederson in einer vergleichbaren Studie über Feature-Selektion – zeigen, dass mit diesem einfachen Verfahren gute Ergebnisse erzielt werden können [Vgl. Das+07, S. 237].
Als Schwellwert kann jeder Wert zwischen 1 (kein Feature wird entfernt) und der Anzahl aller Dokumente in der Trainingsmenge N gewählt werden. Abbildung 5.7 zeigt die Anzahl der verbleibenden Features bei Verwendung unterschiedlicher Schwellwerte.

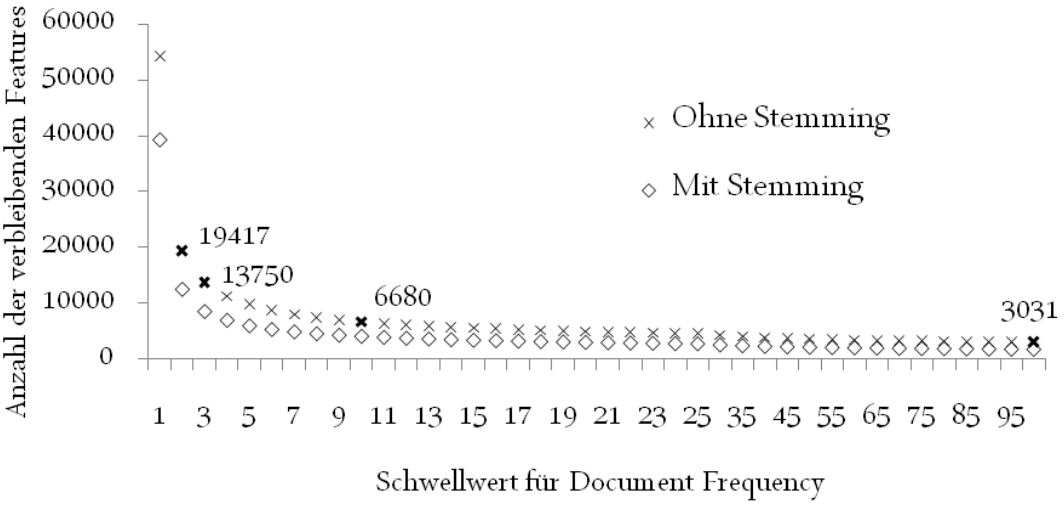

Abbildung 5.7. Finale Schwellwertauswahl für die Document Frequency (Eigene Darstellung).

Alle möglichen Schwellwerte konnten nicht durchgetestet werden. Die Auswahl einiger weniger war erforderlich. Unabhängig von der Reduktion der Features durch zuvor durchgeführtes Stemming konnte eine Verringerung von über 60 Prozent bei Erhöhung des Schwellwertes von 1 auf 2 festgestellt werden. Das heißt, dass mehr als 60 Prozent aller Features in nur einem Dokument der Trainingsmenge vorkommen. Wird der Schwellwert von 2 auf 3 erhöht, führt dies zum Ausschluss von etwa einem Drittel der verbliebenen Features. Die folgenden Schwellwerterhöhungen reduzieren die Anzahl der Features deutlich weniger drastisch. Daher wurde als nächster zu testender Schwellwert 10 ausgewählt. Die Anzahl wird um etwa 50 Prozent im Vergleich zu der bei Schwellwert 3 ermittelten verringert. Schließlich wurde der Schwellwert 100 exemplarisch für „aggressive“ Dimensionsreduktion gewählt. Tabelle 5.4 zeigt die Ergebnisse mit und ohne vorheriges Stemming für die selektierten Schwellwerte.

Schwellwert	2	3	10	100
Anzahl (ohne Stemming)	19.417	13.750	6.680	3.031
Anzahl (mit Stemming)	12.577	8.625	4.198	1.871

Tabelle 5.4. Veränderung der Anzahl der Features bei Schwellwert 2, 3, 10 und 100 für Document Frequency.

5.3.3. Produkt aus Term Frequency und Inverse Document Frequency

Das Produkt aus Term Frequency und Inverse Document Frequency (TF*IDF) wird im Text Mining häufig angewendet (Vgl. [FS07, S. 68], [For07, S. 10]). Für jedes Feature t in jedem Dokument d wird zunächst gezählt, wie oft t in d vorkommt. Diese Anzahl $\mathrm{TF}_{t,d}$ wird mit der inversen Document Frequency IDF_t multipliziert. Sie ist der Quotient aus der Anzahl der Dokumente N und der Document Frequency von Term t – df_t.

$$\mathrm{TF{*}IDF}_{t,d} = \mathrm{TF}_{t,d} * \mathrm{IDF}_t = \mathrm{TF}_{t,d} * \frac{N}{\mathrm{df}_t} \tag{5.1}$$

Der Wert von TF*IDF wird größer, wenn ein Feature häufig in einem Dokument vorkommt. Er sinkt, wenn ein Feature in sehr vielen Publikationen vorkommt. Insbesondere der letzte Teil ist konträr zur Document Frequency, bei der häufige Features bevorzugt werden.
Zahlreiche Varianten von TF*IDF zur Gewichtung der beiden Maße TF und IDF sind verfügbar [Vgl. HS09, S. 188f]. Wenn ein Begriff in einer Publikation 10 statt 15 mal vorkommt, sollte beispielsweise das Ergebnis nicht deutlich anders sein. Dies kann mit Einsetzen von

$$\mathrm{TF}_{t,d} = 1 + \log_2 \mathrm{tf}_{t,d} \tag{5.2}$$

in Gleichung 5.1 erreicht werden [Vgl. HS09, S. 126]. Darüberhinaus wird die Inverse Document Frequency meist als

$$\mathrm{IDF}_t = \log_2 \frac{\mathrm{N}}{\mathrm{df}_t} \tag{5.3}$$

definiert [Vgl. HNP05, S. 28]. Dadurch wird ihr Effekt auf TF*IDF gemindert. Durch Einsetzen von 5.2 und 5.3 in Gleichung 5.1 entsteht

$$\mathrm{TF{*}IDF}_{t,d} = \mathrm{TF}_{t,d} * \mathrm{IDF}_t = \underbrace{(1 + \log_2 \mathrm{tf}_{t,d})}_{\mathrm{TF}_{t,d}} * \underbrace{\log_2 \frac{\mathrm{N}}{\mathrm{df}_t}}_{\mathrm{IDF}_t}. \tag{5.4}$$

Diese relativ komplexe Variante konnte beispielsweise Vitt bei der Klassifikation von MEDLINE-Veröffentlichungen erfolgreich einsetzen [Vit04, Vgl.]. Ihr Einfluss auf die Performanz der Klassifikatoren wird evaluiert. Um von der Dokumentenlänge zu abstrahieren, ist die Normalisierung der Feature-Vektoren erforderlich [Vgl. For07, S. 20], [Joa98, S. 2]. Jeder Wert $\mathrm{tf}_{t,d}$ wird durch die Anzahl der einzigartigen Features in d geteilt. Abbildung 5.8 stellt die Anzahl der Features in Abhängigkeit von den für TF*IDF gewählten Schwellwerten dar.

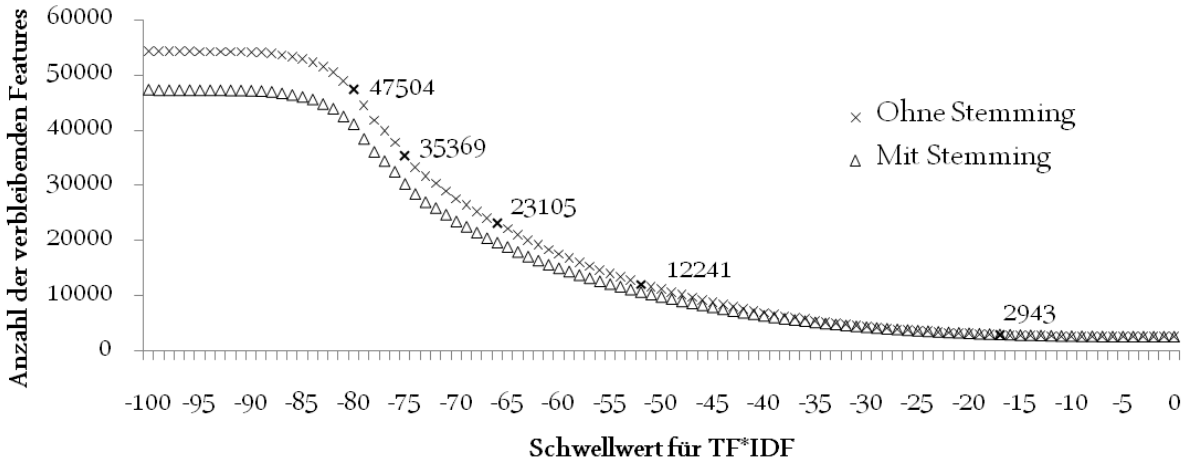

Abbildung 5.8. Finale Schwellwertauswahl für die TF*IDF Methode (Eigene Darstellung).

Für TF*IDF ist ein Schwellwert anzugeben, den ein Feature mindestens in einem Dokument überschreiten muss, um für die weitere Verwendung ausgewählt zu werden. Eine drastische Reduktion der Dimensionen von einem zum nächsten Schwellwert – wie bei der im vorherigen Abschnitt erläuterten Document Frequency – konnte nicht identifiziert werden. Aus diesem Grund wurden vier Schwellwerte ausgewählt, durch die die Anzahl der Features um absolut etwa gleich große Werte reduziert wird. Anschließend wurde – wie für die Document Frequency – ein Wert gewählt, der zu einer extremen Verringerung führt. Alle gewählten Schwellwerte mit Angabe der verbleibenden Feature-Anzahlen mit und ohne vorherigem Stemming sind in Tabelle 5.5 aufgeführt.

Schwellwert	-80	-75	-66	-52	-18
Anzahl (ohne Stemming)	47.504	35.369	23.105	12.241	2.943
Anzahl (mit Stemming)	41.173	30.625	19.623	10.624	3.009

Tabelle 5.5. Veränderung der Anzahl der Features bei Schwellwert -80, -75, -66, -52 und -18 für TF*IDF.

Bei Betrachtung von Tabelle 5.5 fällt auf, dass sich Stemming und TF*IDF mit zunehmender Größe des Schwellwertes scheinbar behindern (die Anzahl der Features bei Schwellwert -18 ist mit Stemming größer als ohne). Dieser Effekt konnte bei der DF nicht beobachtet werden. Eine mögliche Erklärung könnte sein, dass seltene Features vielfältigere Stammformen vorweisen als häufige. Da TF*IDF – im Gegensatz zur Document Frequency – seltene Wörter bevorzugt, bleiben folglich sukzessive mehr Wörter übrig.

5.4. Datentransformation

Nach erfolgreicher Datenvorbereitung und Auswahl der für die Klassifikation genutzten Features ist jede Publikation ein Vektor aus den Feature-Anzahlen. Die Wahrscheinlichkeit des vermehrten Vorkommens von Features steigt mit der Länge der Publikation. Um zu vermeiden, dass die Länge das Ergebnis der Klassifikation beeinflusst, wird normalisiert: Jede Featureanzahl im Vektor wird durch die Anzahl aller Features in der entsprechenden Publikation geteilt. Dieser Vektor der normalisierten Feature-Anzahlen muss nun in ein Format transformiert werden, welches von den Klassifikatoren verwendet werden kann.
In der Webanwendung werden Klassifikatoren genutzt, die das Attribute-Relation File Format (ARFF) verwenden können (Vgl. 8). ARFF-Dokumente sind ASCII-Textdateien zur strukturierten Ablage von großen Datenmengen. Die Instanzen in diesen Datenmengen müssen voneinander unabhängig sein, d. h. sie dürfen keine Beziehungen miteinander haben [Vgl. Bou+10, S. 161]. Im vorliegenden Anwendungsfall kann ARFF zur Speicherung der Publikationsvektoren verwendet werden.
Eine ARFF-Datei beginnt meist mit einem **Kommentarteil**. Allgemeine Informationen wie der Titel der Analyse, der Name des Autors, das Erstelldatum usw. können dokumentiert werden. Jede Kommentarzeile beginnt mit einem Prozentzeichen % [Vgl. WF05, S. 53]:

```
% 1. Title: PublicationTrainingSet
%
% 2. Sources:
% (a) Creator: Christin Lebing
% (b) Date: August, 2010
%
```

Im Anschluss werden der Name der **Relation** – mit „@RELATION “ gekennzeichnet – und die Namen und Typen der **Attribute** – mit „@ATTRIBUTE “ eingeleitet – angegebenen. Dieser obligatorische Teil der ARFF-Datei wird als „Header“ bezeichnet [Vgl. Bou+10, S. 161].

```
@RELATION pubmed

@ATTRIBUTE "two-dimensional" NUMERIC
@ATTRIBUTE "electrophoresis" NUMERIC
...
@ATTRIBUTE documentClass {2DGE,No2DGE}
```

Dargestellt ist ein Teil einer ARFF-Datei, die für die Evaluierung der manuellen Selektion erzeugt wurde. Von den dreizehn tatsächlich vorhandenen Attributen (= verwendeten Features) sind zwei – „two-dimensional“ und „electrophoresis“ – im Ausschnitt enthalten. die restlichen sind durch die drei Punkte angedeutet. Das letzte Attribut „documentclass“[3] gibt die Klassenzugehörigkeit – „2DGE“ oder „No2DGE“ – der Dokumente an. Die Nennung dieses Attributs an letzter Stelle ist üblich aber nicht zwingend erforderlich.

[3] Die Bezeichnung kann frei gewählt werden. Empfehlenswert bei Textdaten ist die Verwendung eines Begriffs, der nicht Feature sein kann. Dadurch würden zwei gleichnamige Attribute erzeugt werden, was nach ARFF-Spezifikation zu einem ungültigen Dokument führt.

Die Instanz-spezifischen **Daten** werden im letzten Abschnitt eines ARFF-Dokuments angegeben. Entsprechend der Attributreihenfolge im Header werden die Attributwerte – durch Kommata voneinander getrennt – in einer Zeile je Instanz eingetragen. Die Attribute der Publikationen sind die ausgewählten Features. Folglich sind die Attributwerte die Feature-Häufigkeiten:

```
@Data
1,1,0,1,0,0,0,0,0,0,0,1,0,"2DGE"
0,0,1,1,1,0,0,0,0,0,0,0,0,"2DGE"
```

Das Beispiel zeigt zwei dünn-besetzte Textvektoren. Die Mehrzahl der Attributwerte ist Null. Für derart spärlich besetzte Daten kann das ARFF-Format abgewandelt werden:

```
@Data
{0 1, 1 1, 3 1, 11 1, 13 "2DGE"}
{2 1, 3 1, 4 1, 13 "2DGE"}
```

Jeder Datensatz wird weiterhin in einer Zeile – nun in geschweiften Klammern – dargestellt. Nur Attributwerte größer Null werden – mit der Positionsnummer des jeweiligen Attributs in der Liste der Attribute zu Beginn der ARFF-Datei – notiert. Vorangestellt wird jedem Attributwert eine Ziffer, die der Position des Features in der Attributliste entspricht. Bei mehreren tausend Features, von denen die meisten nur in wenigen Dokumenten auftreten, ist diese Schreibweise deutlich kürzer. Sie wird in der Webanwendung verwendet. Jede Publikation aus PubMed ist schließlich eine Zeile in einer ARFF-Datei.
Eine ARFF-Datei mit den dreizehn manuell selektierten Features und zwei Publikationen sieht beispielsweise wie folgt aus:

```
% 1. Title: PublicationTrainingSet
%
@RELATION pubmed

@ATTRIBUTE "two-dimensional" NUMERIC
@ATTRIBUTE "electrophoresis" NUMERIC
@ATTRIBUTE "2-de" NUMERIC
@ATTRIBUTE "proteomic" NUMERIC
@ATTRIBUTE "spots" NUMERIC
@ATTRIBUTE "2d-dige" NUMERIC
@ATTRIBUTE "dige" NUMERIC
@ATTRIBUTE "gel-based" NUMERIC
@ATTRIBUTE "gel-electrophoresis" NUMERIC
@ATTRIBUTE "2dge" NUMERIC
@ATTRIBUTE "2d-gel" NUMERIC
@ATTRIBUTE "gel" NUMERIC
@ATTRIBUTE "2d-electrophoresis" NUMERIC
@ATTRIBUTE documentClass {2DGE,No2DGE}

@Data
```

```
{0 1, 1 1, 3 1, 11 1, 13 "2DGE"}
{2 1, 3 1, 4 1, 13 "2DGE"}
```

6. Modellbildung

Kein Klassifikationsverfahren liefert für alle Anwendungen die besten Ergebnisse (Vgl. [Sal97, S. 326], [KHP05, S. 50]). Vielmehr erfordern unterschiedliche Anwendungsgebiete den Einsatz unterschiedlicher Verfahren [Vgl. Seb01, S. 46]. Um bestmögliche Performanz zu erzielen, ist die Evaluierung mehrerer Methoden erforderlich. Der Vergleich aller verfügbaren Klassifikationsverfahren ist aus folgenden Gründen nicht möglich:

- Eine **Vielzahl von Verfahren** steht zur Verfügung. Einige werden beispielsweise in den Data-Mining-Büchern von Tan et al. [Vgl. TSK06, S. 145ff] oder Witten et al. [Vgl. WF05, S. 187ff] erläutert.
- Nahezu jedes Verfahren existiert in **zahlreichen Variationen**. Exemplarisch sei auf die in Abschnitt 6.1.1 genannten Varianten der k-Nearest-Neighbors-Methode verwiesen.
- Die Qualität der Ergebnisse wird durch **diverse Parameter** beeinflusst. Diese müssen zur Optimierung der Performanz mit veränderten Werten getestet werden [Vgl. Sal97, S. 327].

In der Textklassifikation wurden Support Vector Machines und k-Nearest-Neighbors bereits in zahlreichen Fällen erfolgreich angewendet (Vgl. [FS07, S. 80], [Seb01, S. 29], [Yan98, S. 88], [YL99, S. 48]). Diese beiden Verfahren werden daher auch im Rahmen dieser Arbeit evaluiert. Ein weiteres häufig eingesetztes Verfahren ist Naïve Bayes. Alspector und Kolcz konnten bei der Klassifikation von Emails in Spam und Nicht-Spam – ebenso wie Joachims bei der Klassifikation von Reuters-Dokumenten – gute Ergebnisse erzielen (Vgl. [Joa98, S. 141], [KA01, S. 2]). In anderen Anwendungsfällen lieferte das Verfahren deutlich schlechtere Resultate als z. B. k-Nearest-Neighbors und Support Vector Machines (Vgl. [Seb01, S. 45], [YL99, S. 48]). Diese konträren Beobachtungen und der häufige Einsatz im Text Mining führten zur Auswahl von Naïve Bayes als drittes zu testendes Verfahren.
Im folgenden Abschnitt werden die drei ausgewählten Verfahren und die jeweils verwendeten Parameter-Kombinationen erläutert. Anschließend wird der Evaluierungsprozess beschrieben und dessen Ergebnisse präsentiert. Die Parameteroptimierung für die Methode mit der besten Performanz wird im letzten Abschnitt dieses Kapitels betrachtet.

6.1. Verfahren und Parameterauswahl

6.1.1. k-Nearest-Neighbors

„If it walks like a duck, quacks like a duck, and looks like a duck, then it's probably a duck.“[TSK06, S. 224]

Grundidee bei der Klassifikation mit k-Nearest-Neighbors (kNN) ist, dass Elemente mit ähnlichen Eigenschaften gleichen Klassen zugeordnet werden können. Die Klasse eines Testelements wird folglich durch die Klassen der ihm ähnlichsten Instanzen bestimmt. Das Verfahren läuft in den folgenden vier Schritten ab:

1. Berechnung der Ähnlichkeit des Testelements mit allen Elementen der Trainingsmenge.
2. Identifikation der k ähnlichsten Trainingselemente.
3. Ermittlung der Klasse, der die meisten der k ähnlichsten Trainingselemente angehören.
4. Vorhersage der in 3. ermittelten Klasse für das Testelement.

Die Ähnlichkeit wird mithilfe eines Distanzmaßes berechnet [Vgl. Cle10, S. 13f]. Instanzen werden dazu als Vektoren ihrer Attributsausprägungen dargestellt. Die Distanz *dist* zwischen dem Testelementvektor x und dem Vektor eines Trainingselements y wird meist mit der euklidischen Distanz

$$dist_e(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_k-y_k)^2} = \sqrt{\sum_{i=1}^{k}(x_i-y_i)^2} \tag{6.1}$$

berechnet [Vgl. WF05, S. 128]. Diese Metrik wird ebenfalls in der Webanwendung (Vgl. Kapitel 8) verwendet. kNN wird als „lazy learner“ bezeichnet, da während der Trainingsphase kein Modell gebildet sondern lediglich die Trainingsmenge gespeichert wird (Vgl. [TSK06, S. 223], [WF05, S. 413]). Abbildung 6.1 zeigt eine vereinfachte Darstellung des Verfahrens. Trainingselemente der Klasse 1 sind als Dreiecke, die der Klasse 2 als kleine Kreise darge-

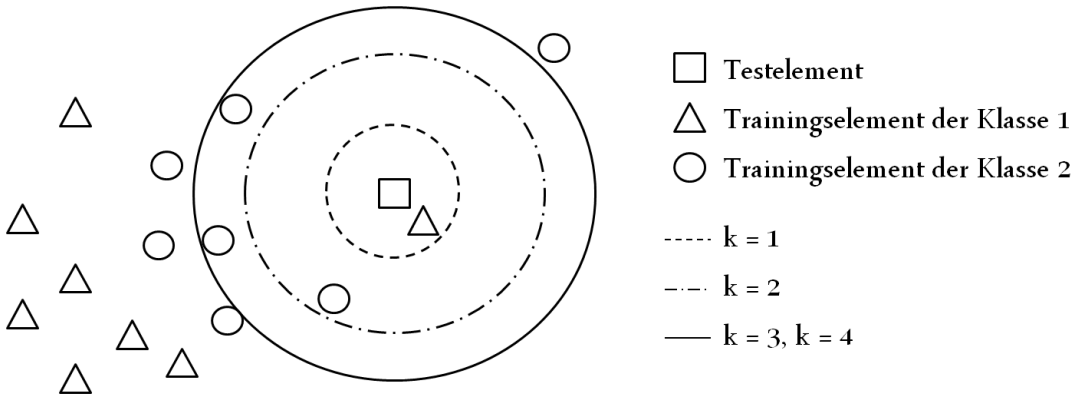

Abbildung 6.1. Vereinfachte Darstellung des kNN-Verfahrens (Eigene Darstellung).

stellt. Das Quadrat stellt ein Testelement dar, für das die Klassenzugehörigkeit vorhergesagt werden soll. Für k wurden drei Werte ausgewählt:

1. $k = 1$ (gestrichelte Kreislinie)

   Lediglich das dem Testelement nächste Trainingselement wird berücksichtigt. Im Beispiel gehört diese Instanz der ersten Klasse an, welche folglich für das Testelement

vorhergesagt werden würde. Bei Betrachtung der Abbildung fällt auf, dass das ermittelte Element ein Ausreißer sein könnte: Alle anderen Dreiecke liegen deutlich näher beieinander und befinden sich links von der Wolke der Kreise. In verrauschten Daten ist die Wahl von $k = 1$ demzufolge eher ungünstig.

2. $k = 2$ (Punkt-Strich-Kreislinie)

   Die beiden, dem Testelement am nächsten gelegenen Elemente gehören jeweils einer der beiden Klassen an. Eine eindeutige Vorhersage für das Testelement ist nicht möglich. Dieser Fall kann immer dann eintreten, wenn gerade Werte für k gewählt werden. Sie sollten daher vermieden werden.

3. $k = 3$, $k = 4$ (durchgezogene Kreislinie)

   Die Identifikation des nächsten und zweitnächsten Elements war eindeutig möglich. Als drittnächstes Element kommen zwei in Frage, da deren Abstand zum Testelement exakt gleich ist. $k = 3$ liefert folglich vier nächste Nachbarn und ist damit identisch mit $k = 4$. Beim Zählen der Klassenzugehörigkeiten ergibt sich ein Verhältnis von 3:1 zugunsten der zweiten Klasse. Diese Klasse würde für das Testelement vorhergesagt werden.

4. $k > 4$ (nicht durch Kreis dargestellt)

   Bei einer weiteren Erhöhung von k bleibt die Vorhersage zunächst gleich. Erst bei Berücksichtigung aller Trainingselemente ändert sich die Verteilung zugunsten der ersten Klasse. Derart hohe Werte von k sind nicht sinnvoll: Für jedes Testelement würde immer die Klasse ermittelt werden, die mit den meisten Instanzen in der Trainingsmenge vertreten ist.

Das Beispiel verdeutlicht den immensen Einfluss der Wahl von k auf das Ergebnis der Klassifikation. Zu kleine (nahe 1) und zu große (nur unwesentlich kleiner oder genauso groß wie die Anzahl aller Trainingselemente) Werte für k sind ungünstig. Ebenso sollten gerade Werte für k vermieden werden.
Der optimale Wert für k kann für einen Anwendungsfall z. B. durch Evaluierung mittels zehnfacher Kreuzvalidierung (Vgl. 2.5.2) ermittelt werden: Unterschiedliche k werden verwendet und die Ergebnisse mithilfe eines Bewertungsmaßes (Vgl. 2.6) verglichen.
Neben einer Anpassung für reellwertige Funktionen [Vgl. Cle10, S. 31] stehen zahlreiche Varianten und Erweiterungen des Verfahrens zur Auswahl. Sie konzentrieren sich vor allem auf die Gewichtung der Nachbarn bzw. der Attribute (hier: Features), um die Performanz des Verfahrens zu verbessern. Zu diesen Erweiterungen zählen „Weight-adjusted k-Nearest-Neighbors“ [Vgl. HKK99] oder „Neighbor-Weighted k-Nearest-Neighbor“ [Vgl. Tan05].
k-Nearest-Neighbors wird in dieser Arbeit in der ursprünglichen Form – ohne weitere Modifikation – verwendet. Für k wurden die Werte 21 und 101 eingesetzt. Ersterer ist in einer im Text Mining üblicherweise gewählten Größenordnung [Vgl. FS07, S. 76]. Letzterer wurde als Vergleichswert getestet. Versehentlich wurde ein Durchlauf von k-Nearest Neighbors mit dem Standardwert $k = 1$ der in der Webanwendung verwendeten Implementation von kNN durchgeführt. Youngs Gesetz „All great discoveries are made by mistake.“ [Cle10, S. 15] folgend und ungeachtet der oben erläuterten Probleme bei zu kleinen k, werden daher auch die Ergebnisse für $k = 1$ in Abschnitt 6.2.2 präsentiert.

6.1.2. Naïve Bayes

Das Naïve Bayes Klassifikationsverfahren basiert auf der Bayesschen Formel

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}. \tag{6.2}$$

$P(A|B)$ ist die Wahrscheinlichkeit des Eintretens eines Ereignisses A unter der Bedingung, dass Ereignis B eingetreten ist. $P(A)$ und $P(B)$ bezeichnen die Wahrscheinlichkeiten des Eintretens eines der beiden Ereignisse - unabhängig davon, ob das andere eingetreten ist [Vgl. TSK06, S. 228]. Setzt sich Ereignis B aus voneinander unabhängigen Teilereignissen b_i zusammen, dann ist

$$P(B|A) = \prod_{i=1}^{k} P(b_i|A) \tag{6.3}$$

[Vgl. TSK06, S. 232].
Bei der Klassifikation von Texten wird jede Instanz als Vektor V der Featurehäufigkeiten v_1, v_2, ..., v_k dargestellt. Jeder Wert v_f gibt an, wie häufig Feature f im Dokument vorkommt. Angenommen wird, dass die Featurehäufigkeiten voneinander unabhängig[1] sind. Unter dieser Annahme kann die Wahrscheinlichkeit $P(V|C_j)$ des Auftretens von V unter der Bedingung, dass Klasse C_j vorliegt, gemäß Gleichung 6.3 berechnet werden. Durch Einsetzen in die bayessche Formel 6.2 kann die Wahrscheinlichkeit $P(C_j|V)$ mit der eine Klasse C_j vorliegt, wenn Vektor V gegeben ist, mit folgender Gleichung ermittelt werden:

$$P(C_j|V) = \frac{\prod_{f=1}^{k} P(v_f|C_j) * P(C_j)}{P(V)} \tag{6.4}$$

[Vgl. TSK06, S. 232]. Der Nenner auf der rechten Seite in Gleichung 6.4 ist die Wahrscheinlichkeit des Auftretens des Vektors V unabhängig von der Klasse C_1, C_2, ..., C_m. Diese Wahrscheinlichkeit berechnet sich wie folgt

$$P(V) = \sum_{i=1}^{m} (P(V|C_i) * P(C_i)) = \sum_{i=1}^{m} (\prod_{f=1}^{k} P(v_f|C_i) * P(C_i)) \tag{6.5}$$

[Vgl. Alp10, S. 50].
Naïve Bayes kann in den folgenden drei Schritten durchgeführt werden:

1. **Berechnung der relativen Häufigkeiten** h_{v_i,C_j}

   Die tatsächliche Wahrscheinlichkeit $P(v_f|C_j)$ ist nicht bekannt. Aus der Trainingsmenge kann stattdessen die relative Häufigkeit jedes Wertes v_f in einer Klasse C_j ermittelt werden. Angenommen wird, dass diese relative Häufigkeit annähernd der realen Wahr-

[1] Diese Annahme begründet die Bezeichung „Naïve Bayes“: Die Unabhängigkeit wird in der Praxis nicht verifiziert – wohl wissend, dass diese nicht immer vorliegt (Vgl. [Alp10, S. 397], [Seb01, S. 23]). Tritt beispielsweise in einem Text „Wetter“ häufig auf, ist das Auftreten des Wortes „Gewitter“ wahrscheinlicher als wenn „Wetter“ nicht und stattdessen „Rotwein“ häufig vorkommen würde.

scheinlichkeit entspricht.

$$h_{v_f,C_j} = \frac{\text{Anzahl der Dokumente der Klasse } C_j \text{ mit gleichem } v_f}{\text{Anzahl aller Dokumente mit gleichem } v_f}. \tag{6.6}$$

Die Berechnung wird für alle Dokumentvektoren V_i aller Klassen C_j während der Modellbildung durchgeführt.

2. **Berechnung der Likelihoods** $L_{C_j}(V)$

   Die Likelihood $L_{C_j}(V)$ ist das Produkt aus der relativen Häufigkeit h_{C_j} einer Klasse C_j

$$h_{C_j} = \frac{\text{Anzahl der Dokumente der Klasse } C_j}{\text{Anzahl aller Dokumente}} \tag{6.7}$$

   und dem Produkt aller relativen Häufigkeiten h_{v_f,C_j} der Featurehäufigkeiten v_f eines Testvektors V. Unter der Annahme, dass die Trainingsdaten exakt die tatsächlichen Wahrscheinlichkeiten widerspiegeln, gilt

$$L_{C_j}(V) = \prod_{f=1}^{k} h_{v_f,C_j} * h_{C_j} = \prod_{f=1}^{k} P(v_f|C_j) * P(C_j). \tag{6.8}$$

   Der letzte Term in der Gleichung entspricht bereits exakt dem Zähler in Gleichung 6.4.

3. **Berechnung der bedingten Wahrscheinlichkeiten** $P(C_j|V)$

   Unter Beibehaltung der Annahmen aus erstem und zweitem Schritt kann die Wahrscheinlichkeit $P(C_j|V)$ aus Gleichung 6.4 wie folgt berechnet werden

$$P(C_j|V) = \frac{L_{C_j}(V)}{\sum_{i=1}^{m} L_{C_i}(V)} \tag{6.9}$$

   [Vgl. Cle10, S. 51].

Einige Abwandlungen von Naïve Bayes – insbesondere mit dem Ziel der Abschwächung der Unabhängigkeitsvermutung – wurden entwickelt. Bisher wurde – soweit bekannt – über deren Einsatz keine signifikante Verbesserung berichtet [Vgl. FS07, S. 71]. Möglichkeiten zur Optimierung durch Parameteranpassungen bietet das Verfahren nicht. Naïve Bayes wurde daher in der beschriebenen Form eingesetzt.

6.1.3. Support Vector Machines

Support Vector Machines (SVM) wurden zuerst von Joachims in [Joa98] für die Klassifikation von Texten der Reuters und der OHSUMED[2] Sammlung erfolgreich eingesetzt. Joachims nennt für den Erfolg von SVM im Text Mining folgende Gründe:

[2] Die OHSUMED Sammlung ist eine Zusammenstellung von bereits klassifizierten MEDLINE-Texten der Oregon Health & Science University (OHSU) [Vgl. OHS10].

- **Die Klassikation von Texten erfordert den Umgang mit hochdimensionalen Daten.**

  Texte bestehen aus hunderten oder gar tausenden voneinander verschiedenen Features. Wird eine große Trainingsmenge aufgebaut (Vgl. Abschnitt 5.1), sind schnell weit über 10.000 Features und damit Dimensionen vorhanden (Vgl. Abschnitt 5.3). Joachims konnte zeigen, dass nur wenige dieser Features bei der Klassifikation von Texten ignoriert werden können. Er weist darüber hinaus auf die Gefahr von Informationsverlust bei zu „aggressiver“ Dimensionsreduktion hin. Im Text Mining muss daher immer mit hochdimensionalen Daten gearbeitet werden. Support Vector Machines kommen mit diesen Daten gut zurecht, ohne Overfitting zu erzeugen [Vgl. Joa98, S. 139].

- **Dokumentvektoren sind dünn besetzt.**

  Kivinen et al. konnten für den Perzeptron-Algorithmus zeigen, dass dieses Verfahren besser als andere Klassifikatoren mit derartigen Daten umgehen kann [Vgl. KWA97, S. 341]. Der Perzeptron-Algorithmus ist ein naher Verwandter von Support Vector Machines. Joachims argumentiert daher, dass auch Support Vector Machines besser mit dünn besetzten Vektoren umgehen können [Vgl. Joa98, S. 140].

- **Support Vector Machines sind vielseitig einsetzbar.**

  Support Vector Machines sind für Anwendungsfälle, in denen die Klasseninstanzen linear trennbar sind, geeignet. Ebenso ermöglicht der Einsatz von Kernel-Funktionen die nicht-lineare Trennung [Vgl. Joa98, S. 138].

Die Grundidee von Support Vector Machines ist die Trennung der Instanzen unterschiedlicher Klassen durch eine Hyperebene. In der vorliegenden Arbeit wird dabei ausschließlich die Trennung zweier Klassen betrachtet. Abbildung 6.2 zeigt Instanzen einer ersten Klasse als Dreiecke und die der zweiten als Kreise. Ähnlich wie bei der Regression (Vgl. 2.4.4) kann die

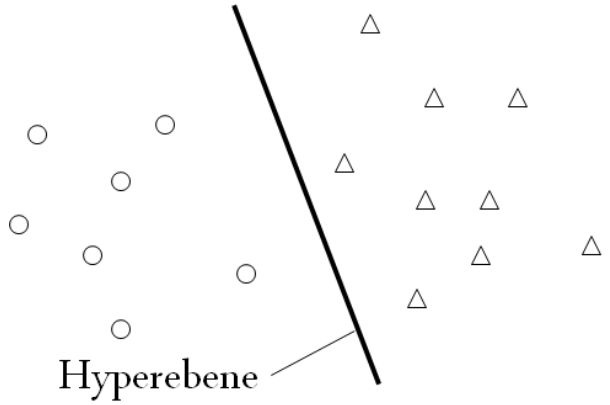

Abbildung 6.2. Lineare Trennung mit einer Hyperebene. Elemente einer ersten Klasse sind als Dreiecke, die der zweiten als Kreise dargestellt (Eigene Darstellung).

trennende Hyperebene mithilfe von Gewichten w_0, w_1, ..., w_k berechnet werden. Jeder Punkt

$\mathbf{x}_h$ auf der Hyperebene erfüllt folgende Gleichung

$$w_0 + w_1 x_{h,1} + w_2 x_{h,2} + \ldots + w_k x_{h,k} = w_0 + \sum_{j=1}^{k} w_j x_{h,j} = w_0 + \mathbf{w}^T \cdot \mathbf{x}_h = 0 \tag{6.10}$$

Alle Instanzen „oberhalb“ der Hyperebene zählen zur ersten, alle „unterhalb“ zur zweiten Klasse. Jedes Trainingselement ist also genau einer Klasse c zugeordnet. Festgelegt wird, dass für alle Instanzen der ersten Klasse $c = +1$ gilt, für die der zweiten $c = -1$. Instanzen, die genau auf der Hyperebene liegen, kann keine Klasse eindeutig zugeordnet werden. Folglich gilt für die Bestimmung der Klassen c_t für alle Instanzen $\mathbf{x}_t$

$$c_t = \begin{cases} +1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t > 0 \\ -1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t < 0 \end{cases} \tag{6.11}$$

[Vgl. TSK06, S. 259]. Falls die Klassen – wie im vorliegenden Beispiel – linear trennbar sind, existieren unendlich viele Lösungen für die trennende Hyperebene. Für die Klassifikation am besten geeignet ist diejenige, die den größten Abstand zwischen den Klassen erzeugt. Diese Hyperebene wird als „Maximum Margin Hyperplane“ bezeichnet. Abbildung 6.3 stellt die Trennung mit der Maximum Margin Hyperplane dar. Der Margin der Hyperebene ist durch

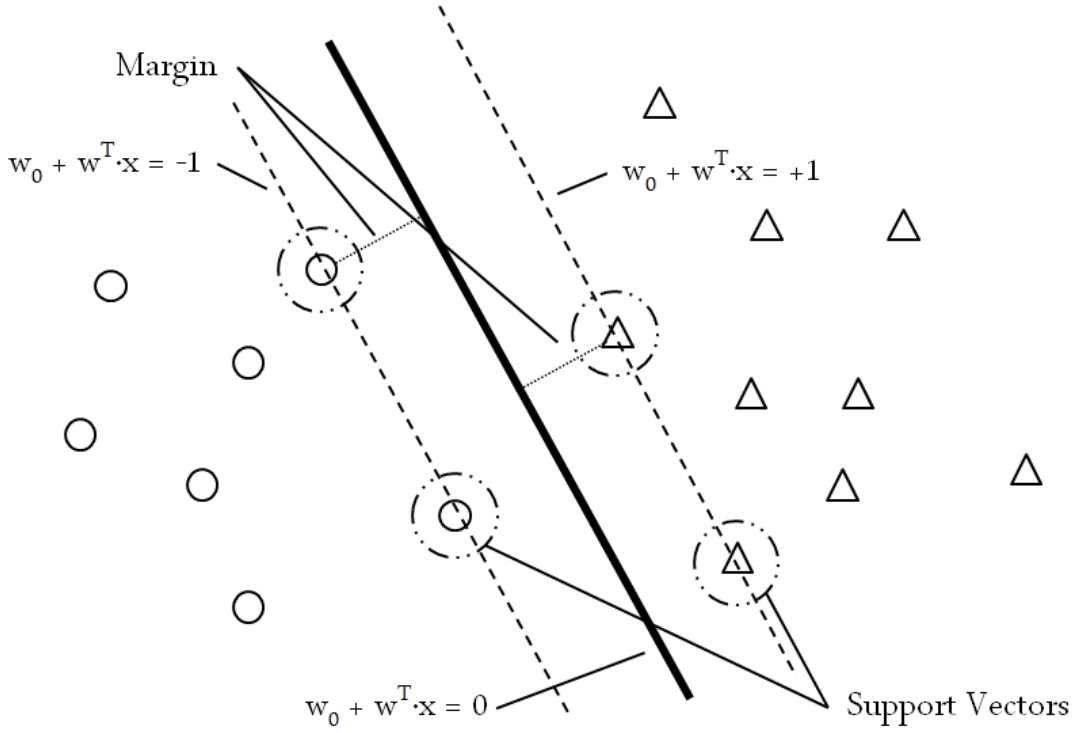

Abbildung 6.3. Lineare Trennung mit Maximum Margin Hyperplane und Support Vectors (Eigene Darstellung).

zwei Randebenen begrenzt. Sie werden im weiteren Verlauf als Rand bezeichnet, die Distanz zwischen diesen Rändern und der Hyperebene als Margin. Gefordert wird, dass alle Instanzen

$\mathbf{x}_o$ auf dem „oberen“ Rand Gleichung 6.12 erfüllen.

$$w_0 + \mathbf{w}^T \cdot \mathbf{x}_o = +1 \tag{6.12}$$

Instanzen $\mathbf{x}_u$ auf dem „unteren“ Rand erfüllen folglich Gleichung 6.13.

$$w_0 + \mathbf{w}^T \cdot \mathbf{x}_u = -1 \tag{6.13}$$

Alle Instanzen $\mathbf{x}_o$ und $\mathbf{x}_u$ werden als Support Vectors bezeichnet. Mindestens eine Instanz aus jeder Klasse ist ein Support Vector.
Festgelegt wird, dass jede Trainingsinstanz $\mathbf{x}_t$ mindestens auf dem Rand der Hyperebene liegen muss. Folglich gilt die Klasse c_t aller $\mathbf{x}_t$:

$$c_t = \begin{cases} +1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t \geq +1 \\ -1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t \leq -1 \end{cases} \tag{6.14}$$

[Vgl. Alp10, S. 311]. Wird c_t in die Formeln eingesetzt, kann die Fallunterscheidung entfallen und Ungleichung 6.15 verwendet werden:

$$c_t(w_0 + \mathbf{w}^T \cdot \mathbf{x}_t) \geq 1 \tag{6.15}$$

Wie kann nun der Margin maximiert werden? Um diese Frage zu beantworten, wird zunächst ermittelt, wie groß euklidische Distanz d zwischen einem beliebigen Vektor $\mathbf{x}$ und der Hyperebene ist. Dazu wird ein Lot auf die Hyperebene gefällt. Der Vektor $\mathbf{w}$ ist orthogonal zur Hyperebene und damit auch der entsprechende Einheitsvektor $\mathbf{w}/|\mathbf{w}|$. Sei $\mathbf{x}'$ der Punkt, an dem das Lot die Hyperebene trifft. Er kann demnach wie folgt berechnet werden:

$$\mathbf{x}' = \mathbf{x} - cd(\frac{\mathbf{w}}{|\mathbf{w}|}) \tag{6.16}$$

[Vgl. HS09, S. 322]. Durch das Einsetzen von der Klasse c von x ist Gleichung 6.16 unabhängig von der Lage von $\mathbf{x}$ bezüglich Hyperebene. Da $\mathbf{x}'$ auf der Hyperebene liegt, muss er Gleichung 6.10 erfüllen. Durch Einsetzen von Gleichung 6.16 in 6.10 ergibt sich:

$$0 = w_0 + \mathbf{w}^T \cdot (\mathbf{x} - cd\frac{\mathbf{w}}{|\mathbf{w}|}) \tag{6.17}$$

Die Umformung nach d führt schließlich zu

$$d = c\frac{\mathbf{w}^T \cdot \mathbf{x} + w_0}{|\mathbf{w}|} \tag{6.18}$$

[Vgl. HS09, S. 323]. Für alle Punkte auf dem Rand – die Support Vectors – wird Ungleichung 6.15 zu einer Gleichung. Daraus folgt, dass der Margin $1/|\mathbf{w}|$ ist. Der Abstand zwischen beiden Rändern ist folglich $2/|\mathbf{w}|$. Die Maximierung dieses Abstandes entspricht der

Minimierung der Funktion $f(w)$

$$f(w) = \frac{|\mathbf{w}|^2}{2} = \frac{\mathbf{w}^T \cdot \mathbf{w}}{2} \tag{6.19}$$

unter der Bedingung

$$c_t(w_0 + \mathbf{w}^T \cdot \mathbf{x}_t) \geq 1, \forall t \tag{6.20}$$

[Vgl. TSK06, S. 261f]. Dies ist ein quadratisches Optimierungsproblem mit linearen Nebenbedingungen, die als Ungleichungen 6.11 gegeben sind. Zur Lösung eines derartigen Problems können Langrange-Multiplikatoren verwendet werden. Die k Ungleichungsnebenbedingungen werden für alle Instanzen $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_k$ und deren Klassen c_1, c_2, ..., c_k als Linearkombination mit Lagrange-Multiplikatoren λ_1, λ_2, ..., λ_k eingebunden:

$$L_P(\mathbf{w}, w_0, \lambda) = \frac{\mathbf{w}^T \cdot \mathbf{w}}{2} - \sum_{t=1}^{k} \lambda_t (c_t(w_0 + \mathbf{w}^T \cdot \mathbf{x}_t) - 1) \tag{6.21}$$

$$= \frac{\mathbf{w}^T \cdot \mathbf{w}}{2} - \sum_{t=1}^{k} \lambda_t c_t (w_0 + \mathbf{w}^T \cdot \mathbf{x}_t) + \sum_{t=1}^{k} \lambda_t \tag{6.22}$$

$$= \frac{\mathbf{w}^T \cdot \mathbf{w}}{2} - w_0 \sum_{t=1}^{k} \lambda_t c_t + \sum_{t=1}^{k} \lambda_t c_t (\mathbf{w}^T \cdot \mathbf{x}_t) + \sum_{t=1}^{k} \lambda_t \tag{6.23}$$

mit den Bedingungen

$$\lambda_t \geq 0, \forall t \tag{6.24}$$

$L_P(\mathbf{w}, w_0, \lambda)$ ist die primale Langrange'sche Funktion. Um $\mathbf{w}$ und w_0 eleminieren zu können, werden die kritischen Punkte gesucht. Dazu werden die Ableitungen von L_P nach w^T bzw. w_0 gleich 0 gesetzt wird. Daraus ergeben sich

$$\frac{\partial Lp}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{t=1}^{k} \lambda_t c_t \mathbf{x}_t \tag{6.25}$$

und

$$\frac{\partial Lp}{\partial w_0} = 0 \Rightarrow \sum_{t=1}^{k} \lambda_t c_t = 0. \tag{6.26}$$

Durch Einsetzen der Gleichungen 6.25 und 6.26 in Funktion 6.23 entsteht die sogenannte duale Lagrang'sche Funktion:

$$L_D = \frac{1}{2} \sum_{t=1}^{k} \sum_{s=1}^{k} \lambda_t \lambda_s c_t c_s \mathbf{x}_t^T \cdot \mathbf{x}_s - \sum_{t=1}^{k} \sum_{s=1}^{k} \lambda_t \lambda_s c_t c_s \mathbf{x}_t^T \cdot \mathbf{x}_s + \sum_{t=1}^{k} \lambda_t \tag{6.27}$$

$$= -\frac{1}{2} \sum_{t=1}^{k} \sum_{s=1}^{k} \lambda_t \lambda_s c_t c_s \mathbf{x}_t^T \cdot \mathbf{x}_s + \sum_{t=1}^{k} \lambda_t. \tag{6.28}$$

mit den Bedingungen

$$\sum_{t=1}^{k} \lambda_t c_t = 0 \quad \text{und} \quad \lambda_t \geq 0, \forall t \tag{6.29}$$

[Vgl. Alp10, S. 313]. Funktion L_D ist zu maximieren. Ein wesentlicher Vorteil im Vergleich zu L_P ist die Unabhängigkeit von der Anzahl der Dimensionen der Trainingsdaten. Lediglich die Anzahl k der Trainingsinstanzen ist für die Komplexität maßgebend.
Zahlreiche Anwendungen für die Lösung eines derartigen Optimierungsproblems sind verfügbar [Vgl. WF05, S. 217]. Beispielsweise ist die numerische Technik „quadratische Programmierung“ anwendbar [Vgl. TSK06, S. 264]. Auf eine Erläuterung wird verzichtet. Für den weiteren Verlauf ist lediglich das Wissen um die Existenz der Methode wichtig.
Die Ungleichung 6.20 wird bei Einsetzen aller $\mathbf{x}_t$, die Support Vectors sind, zu einer Gleichung. Durch Einsetzen dieser Vektoren in die Nebenbedingung von Gleichung 6.19 kann w_0 berechnet werden:

$$w_0 = c_t - \mathbf{w}^T \cdot \mathbf{x}_t \tag{6.30}$$

In Abhängigkeit von den gewählten Support Vectors ändert sich der Wert von w_0. In der Praxis wird der Mittelwert aller berechneten w_0 für die Lösung verwendet [Vgl. TSK06, S. 264]. $\mathbf{w}$ kann mithilfe von Gleichung 6.25 ermittelt werden. Für die Lösung werden nur diejenigen $\mathbf{x}_t$ verwendet, die Support Vectors sind.
Alle bis hierhin aufgeführten Erläuterungen beziehen sich auf linear trennbare Klassen. Sind die Instanzen nicht linear trennbar, bieten Support Vector Machines folgende Möglichkeiten:

Lineare Support Vector Machines mit „Soft Margin“ Der Soft Margin erlaubt, dass Instanzen zwischen Rand und Hyperebene liegen. Trainingsfehler – Instanzen liegen auf der „falschen“ Seite der Hyperebene – werden zugelassen. Diese Lockerung der Nebenbedingung für Gleichung 6.19 wird durch Einfügen von Schlupfvariablen ε_t in Ungleichung 6.11 realisiert. Für alle Trainingselemente $\mathbf{x}_t$ gilt folglich:

$$c_t = \begin{cases} +1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t \geq +1 - \varepsilon_t \\ -1, & \text{wenn } w_0 + \mathbf{w}^T \cdot \mathbf{x}_t \leq -1 + \varepsilon_t. \end{cases} \tag{6.31}$$

Die Schlupfvariablen ε_t müssen für alle t größer Null sein [Vgl. TSK06, S. 267]. Gleichung 6.31 wird wie 6.11 weitergeführt. Zu optimieren ist in diesem Fall die Funktion

$$f(w) = \frac{|\mathbf{w}|^2}{2} + C \sum_{i=1}^{l} \varepsilon_i \tag{6.32}$$

[Vgl. HM04, S. 2]. Die Komplexitätskonstante C wird zur Gewichtung des Fehlers verwendet. Sie kann zur Performanzoptimierung des Verfahrens modifiziert werden (Vgl. Abschnitt 6.2.3). Abbildung 6.4 visualisiert die Trennung mit Linearen Support Vector Machines unter Nutzung von Schlupfvariablen. Alle Instanzen auf und innerhalb des Margins sind hier Support Vectors. Diejenigen Instanzen, die auf der „falschen“ Seite

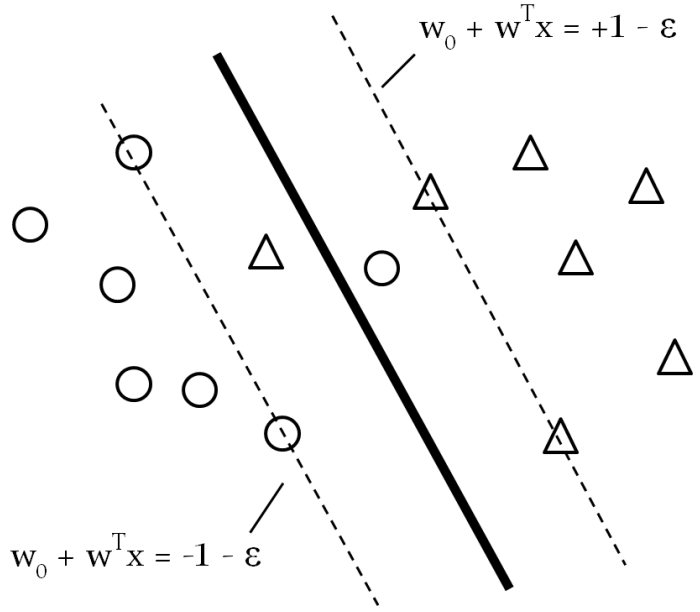

Abbildung 6.4. Nicht-lineare Trennung mit Schlupfvariablen in linearen Support Vector Machins (Eigene Darstellung)

der Hyperebene liegen, werden falsch klassifiziert. Ziel ist die Maximierung des Margins bei gleichzeitiger Minimierung der Fehleranzahl.

Nicht-lineare Support Vector Machines Nicht-lineare Support Vector Machines verwenden eine zu spezifizierende Funktion $\varphi(\mathbf{x})$ mit der die Trainingsdaten in eine höhere Dimension überführt werden. In dieser höheren Dimension sind die Klassen wahrscheinlich linear trennbar. Für den Fall, dass dies nicht möglich sein sollte, werden zusätzlich Schlupfvariablen verwendet. Das oben erläuterte Verfahren wird wie folgt abgewandelt: Die Funktion $\varphi(\mathbf{x})$ wird in Funktion L_D für $\mathbf{x}$ eingesetzt:

$$L_D = -\frac{1}{2}\sum_{t=1}^{k}\sum_{s=1}^{k}\lambda_t\lambda_s c_t c_s \varphi(\mathbf{x}_t)^T \cdot \varphi(\mathbf{x}_s) + \sum_{t=1}^{k}\lambda_t \tag{6.33}$$

mit den Bedingungen

$$\sum_{t=1}^{k}\lambda_t c_t = 0 \quad \text{und} \quad \lambda_t \geq 0, \forall t. \tag{6.34}$$

Weiterhin wird $\mathbf{x}$ durch Funktion $\varphi(\mathbf{x})$ in Gleichung 6.25 und 6.30 ersetzt:

$$\mathbf{w} = \sum_{t=1}^{k}\lambda_t c_t \varphi(\mathbf{x}_t) \tag{6.35}$$

$$w_0 = c_t - \mathbf{w}^T \varphi(\mathbf{x}_t) \tag{6.36}$$

Im Gegensatz zu Gleichung 6.28 für linear-trennbare Klassen wird L_D in 6.33 nun mit zunehmender Anzahl der Dimensionen komplexer. Um dies zu vermeiden, wird der „Kernel-Trick“ verwendet: Das innere Produkt von $\varphi(\mathbf{x}_t)^T$ und $\varphi(\mathbf{x}_s)$ in Gleichung 6.33 wird – bei gleichbleibenden Bedingungen für λ_t – durch eine Kernel-Funktion $K(\mathbf{x}_t, \mathbf{x}_s)$ ersetzt:

$$L_D = -\frac{1}{2} \sum_{t=1}^{k} \sum_{s=1}^{k} \lambda_t \lambda_s c_t c_s K(\mathbf{x}_t, \mathbf{x}_s) + \sum_{t=1}^{k} \lambda_t. \tag{6.37}$$

Im Jahr 1999 veröffentlichte John Platt in [Pla99] den – soweit bekannt – schnellsten Algorithmus zur Lösung dieses Optimierungsproblems: **Sequential Minimal Optimization (SMO)**. Die in der Webanwendung verwendeten Weka-SVM-Klassen nutzen diesen Algorithmus.

Die Kernel-Funktion verwendet die Vektoren in der ursprünglichen Dimensionalität. Dadurch steigt die Komplexität wieder mit der Anzahl der Trainingsinstanzen k und nicht mit der Anzahl der Dimensionen d. Eine der bekanntesten ist die polynomiale Funktion

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \cdot \mathbf{y} + 1)^p. \tag{6.38}$$

Sie wurde im Text Mining bereits mehrfach erfolgreich eingesetzt (Vgl. [KHP05, S. 47], [Joa98, S. 141]). Abbildung 6.5 zeigt, wie die Trennung in der ursprünglichen Dimension aussieht. Neben der polynomialen Kernel-Funktion konnten mit Radial Basis Functions (RBF) wie

$$K(\mathbf{x}, \mathbf{y}) = e^{-(\gamma \|\mathbf{x}-\mathbf{y}\|^2)} \tag{6.39}$$

mehrfach gute Resultate erzielt werden [Vgl. HS09, S. 332].

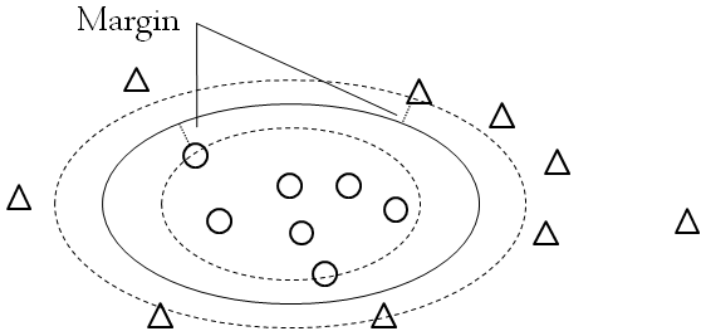

Abbildung 6.5. Nicht-lineare Trennung mit Support Vector Machines und polynomialer Kernel-Funktion (In Anlehnung an [TSK06, S. 275]).

Aufgrund der erfolgreichen Anwendung von Support Vector Machines mit polynomialem und RBF-Kernel wurden diese beiden Varianten evaluiert. In der Webanwendung wurden die Klassen aus dem Open Source Projekt „Weka“ für die Klassifikation verwendet (Vgl. Kapitel 8). Weka bietet – neben RBF- und polynomialem Kernel – eine normalisierte Variante des polynomialen Kernels an:

$$K(\mathbf{x},\mathbf{y}) = \frac{(\mathbf{x}^T \cdot \mathbf{y}+1)^p}{\sqrt{(\mathbf{x}^T \cdot \mathbf{x}+1)^p (\mathbf{x}^T \cdot \mathbf{y}+1)^p}}. \tag{6.40}$$

Welchen Einfluss hat die Normalisierung auf die Performanz? Diese Frage weckte die Neugier der Autorin und führte dazu, dass auch dieser Kernel getestet wurde.
Für alle Kernel wurden die in Weka vordefinierten Standardwerte für die Parameter getestet. Beim RBF-Kernel ist der Parameter γ standardmäßig 0,01. Der normalisierte polynomiale Kernel nutzt 2 als Standardwert für den Exponenten p. Eine Ausnahme wurde beim polynomialem Kernel gemacht: Exponent p ist standardmäßig gleich 1, was einer linearen Variante entspricht [Vgl. WF05, S. 219]. Daher wurde für diesen Kernel zusätzlich die Variante mit $p = 2$ evaluiert.

6.2. Evaluierung

6.2.1. Testaufbau

Evaluiert wurden alle Kombinationen aus den in Kapitel 5 erläuterten Datenvorbereitungs- und Feature-Auswahlmethoden und den Klassifikationsverfahren aus dem ersten Abschnitt dieses Kapitels. Abbildung 6.6 visualisiert diese drei Dimensionen der Evaluierung. Die Datenvorbereitungsmethoden sind in der feinsten Granularität dargestellt. Feature-Auswahl- und Klassifikationsverfahren sind teilweise aggregiert. Nahezu jeder Teilwürfel beinhaltet folglich mehrere Kombination. Diese hohe Komplexität erfordert – zugunsten der Übersichtlichkeit in den Ergebnisdarstellungen – die Abkürzung bzw. Kodierung der Verfahrensnamen:

- **Datenvorbereitung** (vier Varianten, vgl. Abschnitt 5.2.2)
  - **S**: Stemming
  - **W**: Entfernen der stop words
  - **WS**: Kombination aus W und S

  Ist kein Zeichen angegeben, wurde keine weitere Modifikation vorgenommen.
- **Feature-Selektion** (elf Varianten, vgl. Abschnitt 5.3)
  - **All**: Verwendung aller Features
  - **Manuell**: Manuelle Selektion
  - **DF**: Document Frequency, Schwellwerte werden angehängt (z. B. ist „DF2“ Document Frequency mit Schwellwert 2)

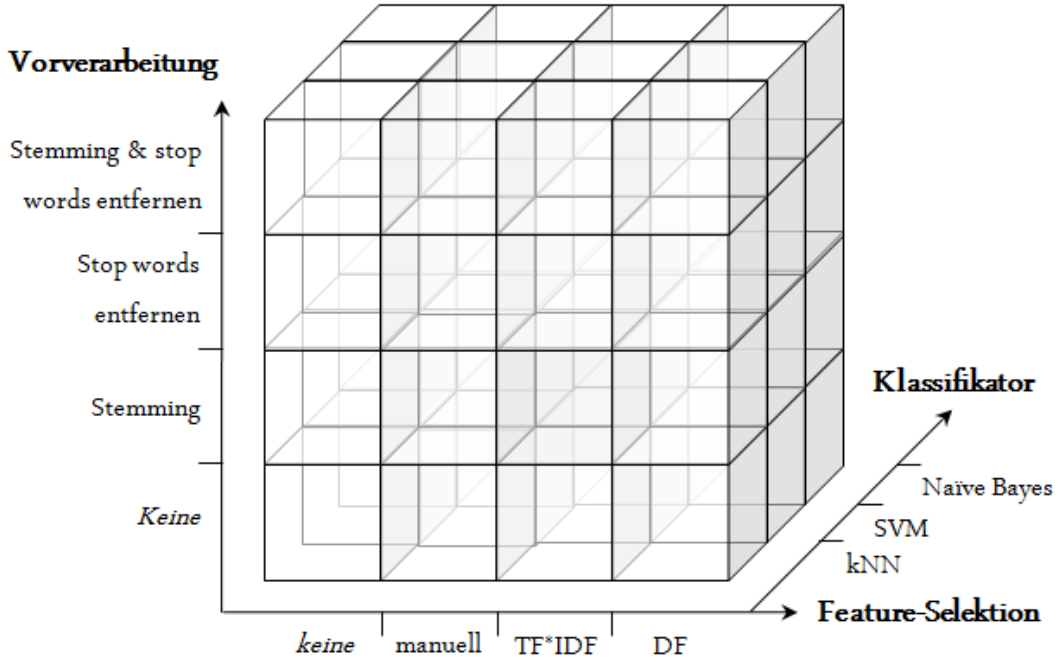

Abbildung 6.6. Kombinationen aus Datenvorbereitungs-, Feature-Auswahl- und Klassifikationsverfahren (Eigene Darstellung).

  - **TF*IDF**: Produkt aus Term Frequency und Inverse Document Frequency (Schellwerte werden – ohne Vorzeichen – wie bei DF direkt angehängt)

- **Klassifikatoren** (acht Varianten, vgl. Abschnitt 6.1)
  - **kNN**: k-Nearest-Neighbors (der Wert für k wird mit angegeben: z. B. „kNN; k=1“ für k gleich 1)
  - **SVM_PK**: Support Vector Machines mit polynomialem Kernel (der Wert für den Exponenten p wird bei $p = 2$ entsprechend angegeben)
  - **SVM_NPK**: Support Vector Machines mit normalisiertem, polynomialem Kernel
  - **SVM_RBF**: Support Vector Machines mit Radial-Basis-Function-Kernel

  Die Bezeichnung **Naïve Bayes** ist kurz und konnte daher vollständig verwendet werden.

Jede Alternative aus jedem der drei Bereiche wird mit den Alternativen der jeweils anderen beiden Bereiche kombiniert. Eine Ausnahme stellt die manuelle Selektion dar: Weder Stemming noch das Entfernen von stop words ist sinnvoll. Ersteres würde die ausgewählten Features in die Stammform überführen. Dadurch würden sie nicht mehr mit den festgelegten Features identisch sein. Letzteres ist unnötig, da keine stop words in der manuell erstellten Feature-Liste enthalten sind (Vgl. Abschnitt 5.1). Insgesamt existieren folglich **328 Kombinationen**. Zur Abschätzung der Performanz wurde jede einer **zehnfachen Kreuzvalidierung** unterzogen. Diese Trainingsstrategie wird für das Text Mining empfohlen (Vgl. [Sal97, S. 325], [WF05, S. 150], [FS07, S. 79]).

Die Evaluierung beschränkt sich auf den Vergleich der Bewertungsmaße wie sie in Abschnitt 2.6 erläutert wurden. Der zeitliche bzw. der Ressourcen-Aufwand wird aus den folgenden Gründen nicht berücksichtigt.

1. **Der zeitliche Aufwand ist gering.**

   Der zeitliche Aufwand steigt – bei gleichbleibender Größe von Trainings- und Testmenge – mit der Anzahl der Features. Die höchste Anzahl verbleibt, wenn keine weitere Datenvorbereitung und keine Feature-Selektion durchgeführt wird. Werden die genannten Klassifikatoren mit 2862 Publikationen trainiert und mit 318 getestet (entspricht einer Iteration in der Kreuzvalidierung), dann werden *in etwa* die in Tabelle 6.1 angegebenen Zeiten benötigt. Monatlich werden in PubMed circa 800 Publikationen eingestellt, die

	Modellbildung	Anwendung des Modells	Gesamt
Naïve Bayes	04:05	00:19	04:24
kNN; k=1	00:00	00:24	00:24
kNN; k=21	00:00	00:27	00:27
kNN; k=101	00:00	00:29	00:29
SVM_PK	00:26	00:02	00:28
SVM_PK; p=2	01:09	00:02	01:11
SVM_NPK	01:39	00:03	01:42
SVM_RBF	01:21	00:03	01:24

Tabelle 6.1. Dauer (mm:ss) der Klassifikation ohne Stemming, Entfernung von stop words oder Anwendung eines Feature-Selektionsverfahrens.

   Treffer bei der in Abschnitt 5.1.1 erläuterten Suche sein werden. Die Bewältigung dieser Menge wird – auch bei sukzessiver Vergrößerung der Trainingsmenge zur weiteren Optimierung der Klassifikation – nur wenige Minuten dauern.

2. **Die Klassifikation in der Webanwendung wird außerhalb der Geschäftszeiten durchgeführt.**

   Die Arbeitszeit der Mitarbeiter von DECODON wird nicht in Anspruch genommen. Kein Mitarbeiter muss warten. Zusätzliche Kosten entstehen nicht.

3. **Der Ressourcenbedarf ist niedrig.**

   Handelsübliche Computer können die notwendigen Berechnungen durchführen. Der eingesetzte Rechner kann auch für andere Aufgaben verwendet werden und muss nicht exklusiv für die Klassifikation zur Verfügung stehen.

6.2.2. Vergleich der Verfahren

Für alle evaluierten Verfahren wurden die Bewertungsmaße Erfolgsrate, Recall und Precision für die relevante Klasse „2DGE“ ermittelt. Aus den beiden letztgenannten wurde zusätzlich das F-Measure berechnet. Die Diagramme in den Abbildungen 6.7 und 6.8 zeigen die durchschnittlichen F-Measures nach 10-facher Kreuzvalidierung aller Kombinationen. Eine tabellarische Übersicht dieser Ergebnisse ist in Anhang A.2 aufgeführt.
Folgende Erkenntnisse konnten aus dem Vergleich der F-Measures gewonnen werden:

Datenvorverarbeitungsmethoden

- Die Methoden der Datenvorverarbeitung hatten den *geringsten Einfluss auf das F-Measure*. Die Ergebnisse wurden am stärksten durch den Klassifikator bestimmt. Anschließend entschied vor allem das gewählte Feature-Selektionsverfahren und der Schwellwert über das Resultat der Klassifikation.
- In Kombination mit Support Vector Machines und Feature-Selektion mit DF bzw. ohne weitere Dimensionsreduktion war die *ausschließliche Anwendung des Porter-Stemming-Algorithmus am erfolgreichsten*.

Feature-Selektionsverfahren

- *Mit manueller Feature-Selektion konnten stets gute Ergebnisse* erzielt werden. Unabhängig vom eingesetzten Klassifikator waren die Resultate sehr gut – bei Naïve Bayes und k-Nearest-Neighbors sogar besser als mit allen anderen Auswahlverfahren. Laut Kruskal-Wallis-Test ($\alpha = 0,05$) waren lediglich Support Vector Machines mit RBF-Kernel bei manueller Selektion signifikant schlechter als bei Feature-Auswahl mit DF (Schwellwerte 2, 3, 10) bzw. mit allen Features.

  Demnach ist die manuelle Selektion sehr gut für Ad-hoc Tests geeignet. Voraussetzung ist die gute Kenntnis der Daten. Nachteil dieser Methode ist der damit verbundene manuelle Aufwand. Jede Übertragung auf eine neue Zielstellung bedingt die erneute Auswahl von Features. Zudem waren einige Kombinationen mit anderen Feature-Auswahlverfahren noch erfolgreicher (z. B. DF in Kombination mit SVM). Aus diesen Gründen wird die Verwendung der manuellen Selektion für die Nutzung in der Anwendung nicht in Erwägung gezogen.
- Der *Einfluss von TF*IDF und DF auf die Performanz war gegensätzlich*. Feature-Auswahl mittels TF*IDF führte mit Support Vector Machines zu schlechteren Ergebnissen, zu gleichbleibenden oder sogar besseren mit k-Nearest-Neighbors und Naïve Bayes. Umgekehrt war der Einfluss der Auswahl mittels Document Frequency. Die Kombinationen mit DF und SVM waren stets besser als TF*IDF mit kNN bzw. Naïve Bayes.
- Die *Auswirkung von hoher bzw. geringer Featureanzahl war ebenso konträr*:
  - Die *Performanz von k-Nearest-Neighbors und Naïve Bayes verbesserte sich mit der Verringerung der Anzahl der Features*. Beide Verfahren kommen mit hochdimensionalen Daten nicht gut zurecht [Vgl. For07, S. 2]. Mit weniger

Features konnte – sowohl bei kNN als auch bei Naïve Bayes – eine Steigerung der Performanz beobachtet werden.

– *Support Vector Machines lieferten meist bessere Resultate mit mehr Features*. Die im Mittel besten Ergebnisse lieferten Support Vector Machines bei Verwendung aller Features bzw. Dimensionreduktion mittels Document Frequency. Bei Selektion von Document Frequency wirkte sich lediglich die aggressive Dimensionsreduktion mit Schwellwert 100 negativ auf die Performanz aus. Joachims konnte bei Anwendung von Support Vector Machines ähnlich überzeugende Resultate ohne Feature-Selektion erzielen [Vgl. Joa98, S. 142].

  Lediglich bei der Klassifikation mit polynomialem Kernel und $p = 2$ sowie mit normalisiertem polynomialem Kernel konnte bei Feature-Auswahl mit TF*IDF bei sinkender Feature-Anzahl eine Verbesserung der Ergebnisse beobachtet werden. Die besten Resultate blieben jedoch hinter denen mit allen Features bzw. geringer Dimensionsreduktion mit DF zurück.

Klassifikatoren

- Der Klassifikator *k-Nearest-Neighbors lieferte mit $k = 21$ die schlechtesten Ergebnisse*. Die Resultate bei $k = 1$ und $k = 101$ waren sehr ähnlich und stets gleich oder etwas besser als bei $k = 21$.
- Alle *kNN-Varianten schnitten deutlich schlechter als die anderen Klassifikatoren ab*. Häufig mussten F-Measures von $0,0$ beobachtet werden. D. h. keine der relevanten Publikationen wurden als solche klassifiziert (TP $= 0$ woraus folgt Precision $= 0,0$ und Recall $= 0,0$).
- *Support Vector Machines waren in nahezu allen Kombinationen deutlich erfolgreicher als Naïve Bayes und k-Nearest-Neighbors*. Lediglich bei manueller Selektion schnitten Support Vector Machines mit RBF-Kernel schlechter ab als kNN und Naïve Bayes bei gleicher Auswahl-Strategie. Dies bestätigt beispielsweise die Arbeiten von Wilbur und Kim, die unter anderem Reuters-Texte und MEDLINE-Veröffentlichungen mit SVM klassifizierten [Vgl. WK09], sowie von Kolcz und Alspector, die SVM als bestes Verfahren für Email-Spam-Filtering ermittelten ([Vgl. KA01]).

  Die relativ schlechte Performanz von k-Nearest-Neighbors und Naïve Bayes konnte lediglich bezüglich der Klasse „2DGE“ beobachtet werden. Erfolgsraten von weit über 0,8 bei nahezu allen Kombinationen mit kNN bestätigen, dass das Verfahren gut für die Textklassifikation geeignet ist (Vgl. Tabelle der Erfolgsraten A.3 in Anhang A.2.2).
- *Bester Kernel für SVM war der polynomiale in der linearen Variante*. Die Erhöhung des Exponenten p für den polynomialen Kernel vom Weka-Standardwert 1 auf 2 führte zu einer Verschlechterung der Resultate. Dies bestätigt die Ergebnisse von Joachims, der zeigen konnte, dass Textklassifikationsprobleme linear trennbar sind [Vgl. Joa98, S. 140]. Begründung liegt in der hohen Anzahl der Features, die

in der Regel deutlich größer als die Anzahl der zu klassifizierenden Texte ist. Die lineare Trennung ist stets möglich [Vgl. BK10, S. 86].

Die beste Kombinationen aus Feature-Selektionsverfahren und Klassifikator sind diejenigen ohne weitere Feature-Auswahl bzw. Auswahl mittels Document Frequency (Schwellwerte 2, 3 und 10) und Support Vector Machines. Alle anderen Kombinationen sind – gemäß Kruskal-Wallis-Test ($\alpha = 0,05$) – signifikant schlechter. Abbildung 6.9 zeigt die Ergebnisse dieser besten Varianten (beim polynomialen Kernel ist lediglich die erfolgreichere Variante mit $p = 1$ abgebildet.). In der Abbildung wird die stets bessere Performanz bei Verwendung des polynomialen Kernels deutlich. Das durchschnittlich höchste F-Measure von 0,852 wurde mit polynomialem Kernel in Kombination mit Stemming und DF-Schwellwert 2 ermittelt. Laut Kruskal-Wallis-Test ($\alpha = 0,05$) ist diese Kombination signifikant besser als DF10 ohne weitere Datenvorbereitung bzw. mit ausschließlicher Entfernung der stop words. Leider konnte beim Vergleich mit den anderen Verfahren (Vgl. Abbildung 6.9) mittels Kruskal-Wallis-Test kein signifikanter Unterschied festgestellt werden.
Der Boxplot der erfolgreichsten Kombination ist in der Abbildung in orange-rot hervorgehoben. Die gestrichelten Linien am oberen und unteren Quartil dienen der Orientierung. Die Verteilung ist – im Gegensatz zu anderen mit größerem Median – nicht rechtsschief. Die Whisker sind relativ kurz, Ausreißer nicht vorhanden. Die Ergebnisse waren folglich nahezu gleichbleibend gut. Zudem konnte bei der genannten Kombination das höchste obere Quartil und das beinahe größte untere Quartil beobachtet werden. Die entsprechende durchschnittliche Konfusionsmatrix ist in Tabelle 6.2 zu sehen.

		Vorhergesagte Klasse	
		„2DGE“	„No2DGE“
Tatsächliche	„2DGE“	32	7
Klasse	„No2DGE“	4	275

Tabelle 6.2. Konfusionsmatrix der durchschnittlichen Ergebnisse für Support Vector Machines mit polynomialem Kernel mit Stemming und Dimensionsreduktion mit dem Feature-Selektionsverfahren Document Frequency (Schwellwert 2).

Einerseits könnte argumentiert werden, dass eine Kombination mit höherem Schwellwert aufgrund des geringeren Aufwands vorzuziehen ist. Andererseits ist letzterer – wie im vorangegangen Abschnitt erläutert – im vorliegenden Anwendungsfall nebensächlich. Zudem steigt der Aufwand bei Support Vector Machines mit der Anzahl der Elemente. Die Anzahl der Dimensionen (= Anzahl der Features) ist nahezu irrelevant (Vgl. Abschnitt 6.1.3).
Support Vector Machines mit polynomialem Kernel, vorangegangenem Stemming und Dimensionsreduktion mittels Document Frequency (Schwellwert 2) wurde als bestes Verfahren für die weitere Verwendung gewählt.

6.2.3. Parameteroptimierung für das erfolgreichste Verfahren

Die Kombination aus Stemming, DF2 und SVM_PK mit $p = 1$ wurde im vorangegangenen Abschnitt als erfolgeichste ermittelt. Ziel dieses Abschnitts ist zu zeigen, wie die Performanz durch Änderung von Standardwerten für einige Parameter (durch Weka festgelegt und bisher unverändert verwendet (Vgl. Abschnitt 6.1.3)) weiter optimiert werden konnte.
Zunächst wurde die standardmäßige Weka-Normalisierung der Attributwerte ausgeschaltet. Dadurch konnte eine Verbesserung des mittleren F-Measures von $0,852$ auf $0,854$ beobachtet werden.
Anschließend wurde – wie bespielsweise bei [Vgl. HCL03, S. 3] empfohlen – die Klassifikation mit modifizierter Komplexitätskonstante C (Vgl. Abschnitt 6.1.3) evaluiert: In Abbildung 6.10 ist die Veränderung des durchschnittlichen F-Measures in Abhängigkeit von dem für C eingesetzten Wert dargestellt. Als optimaler Wert für C wurde im vorliegenden Anwendungsfall $0,021$ ermittelt. Mit dieser Änderung konnte das F-Measure auf $0,862$ verbessert werden. Dementsprechend hat sich die Konfusionsmatrix verändert: Tabelle 6.3 zeigt die aktualisierte Variante. Im Mittel wurde ein False Negative weniger beobachtet.

		Vorhergesagte Klasse	
		„2DGE“	„No2DGE“
Tatsächliche	„2DGE“	33	6
Klasse	„No2DGE“	4	275

Tabelle 6.3. Konfusionsmatrix der durchschnittlichen Ergebnisse nach Parameteroptimierung für Support Vector Machines mit polynomialem Kernel mit Stemming und Dimensionsreduktion mit dem Feature-Selektionsverfahren Document Frequency (Schwellwert 2).

In Abschnitt 5.1.1 wurde auf den positiven Einfluss der Erweiterung der Trainingsmenge hingewiesen. Daher wurden nun zehnfache Kreuzvalidierungen mit vier unterschiedlich großen Instanzmengen zur Beobachtung des Lernverhaltens durchgeführt. Abbildung 6.11 zeigt die positive Lernentwicklung. Neben der Optimierung der Parameter wird demnach die Erweiterung der Trainingsmenge zur Verbesserung der Resultate führen.
In der Webanwendung werden – nach Stemming und Dimensionsreduktion mittels Document Frequency mit Schwellwert 2 – Support Vector Machines mit polynomialem Kernel $p = 1$ und $C = 0,021$ verwendet.

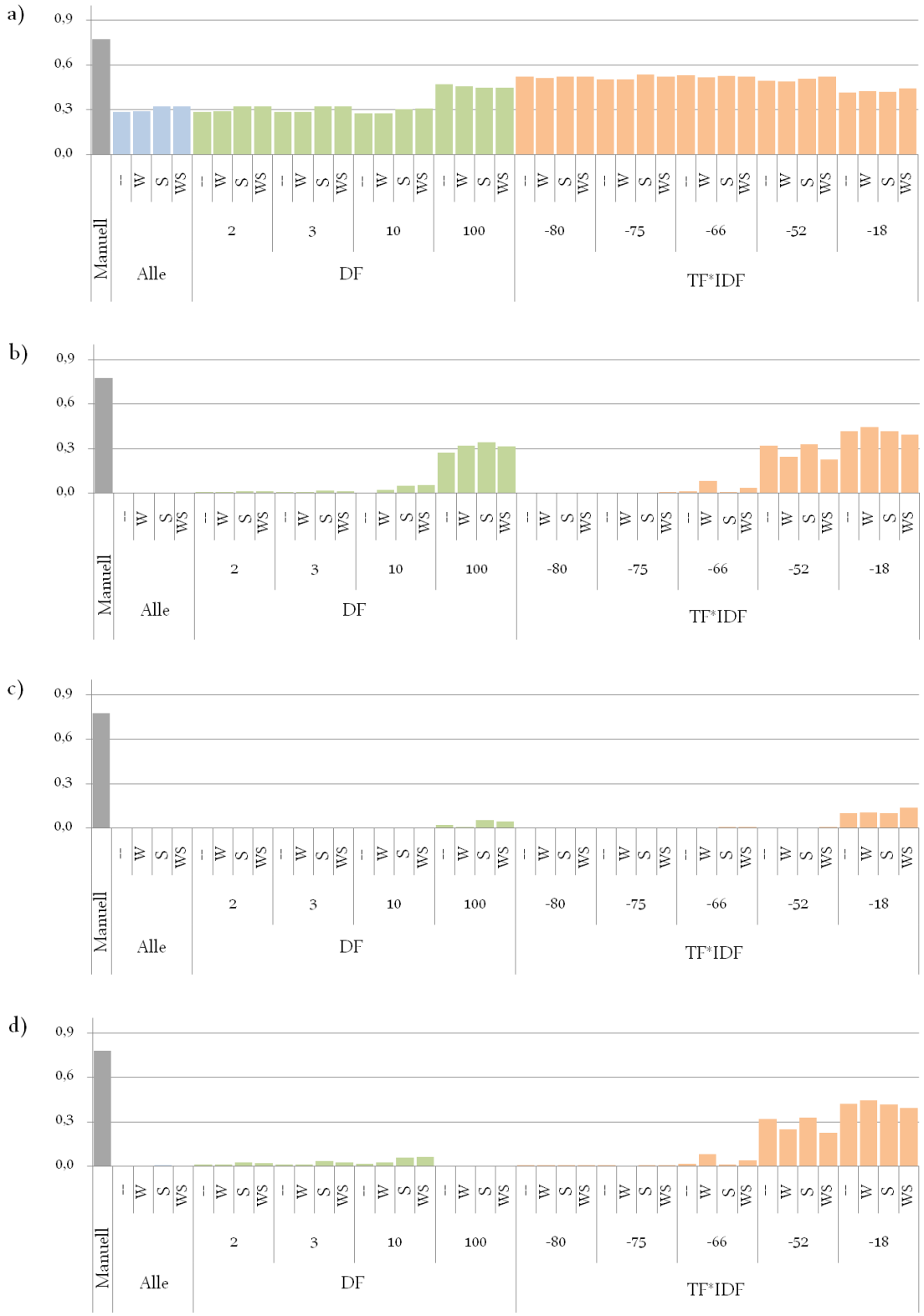

Abbildung 6.7. Arithmetisches Mittel der F-Measures aus den Iterationen der 10-fachen Kreuzvalidierung für a) Naïve Bayes, b) kNN; k=1, c) kNN; k=21 und d) kNN; k=101 (Eigene Darstellung).

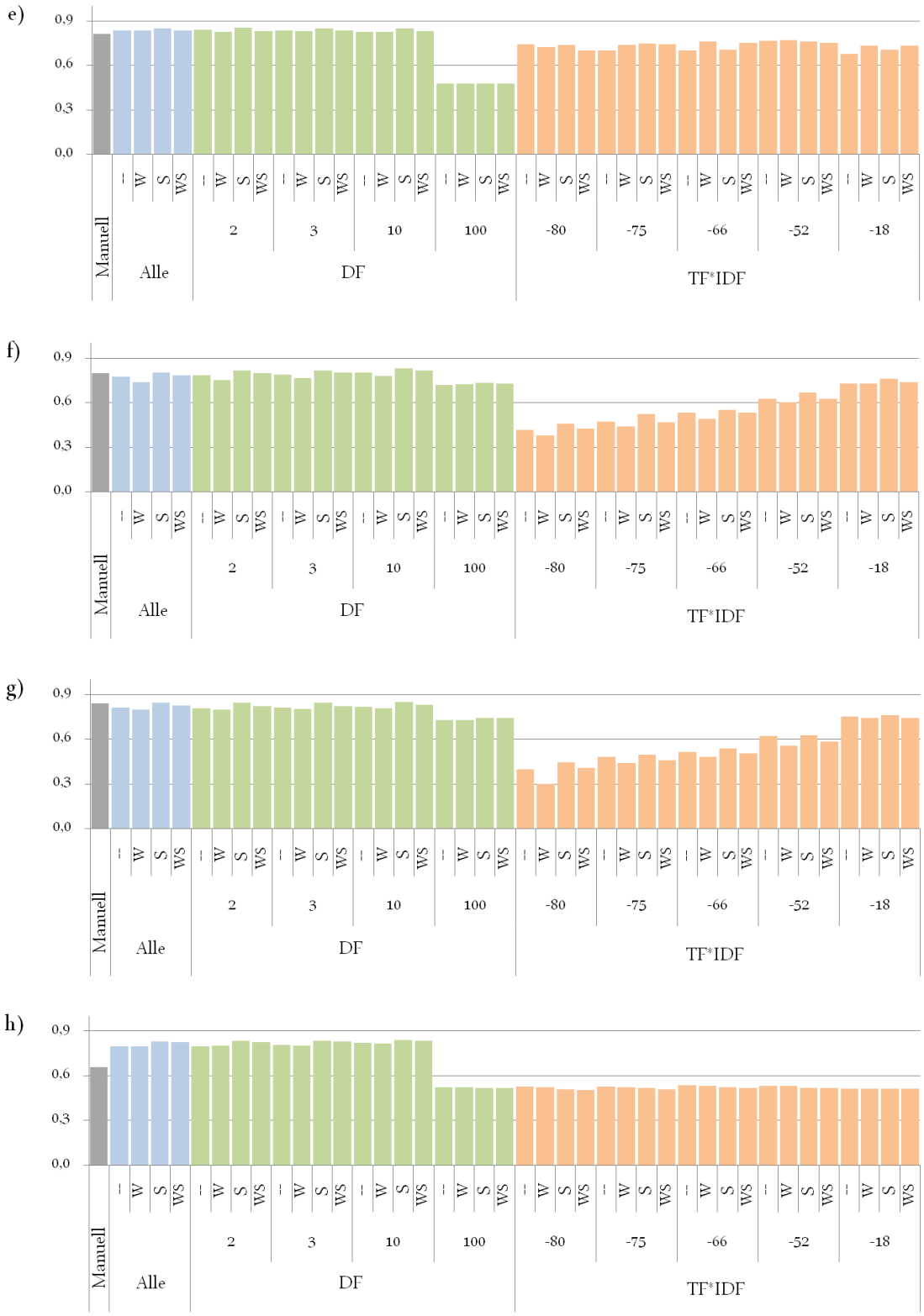

Abbildung 6.8. Arithmetisches Mittel der F-Measures aus den Iterationen der 10-fachen Kreuzvalidierung für e) SVM_PK, f) SVM_PK mit p=2, g) SVM_NPK und h) SVM_RBF (Eigene Darstellung).

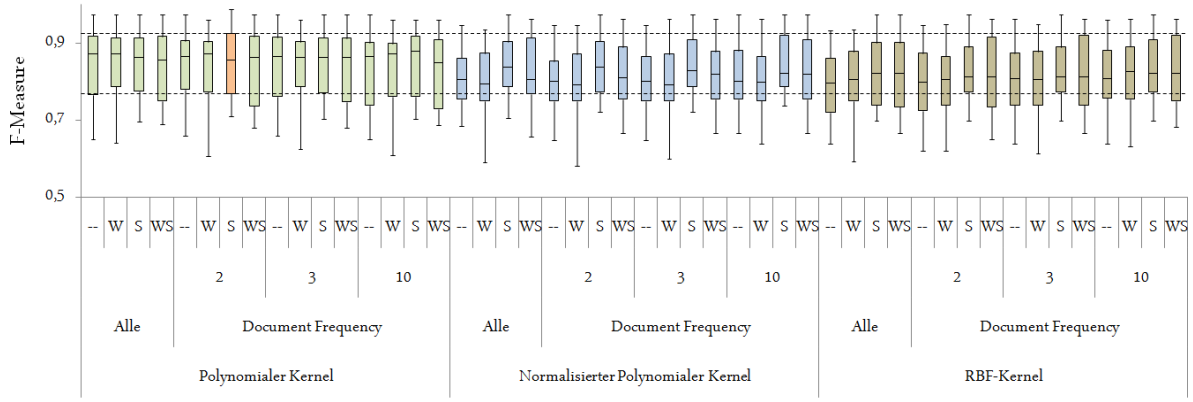

Abbildung 6.9. F-Measures nach 10-facher Kreuzvalidierung aller SVM-Kernel mit allen Features bzw. Feature-Auswahl mittels Document Frequency mit den Schwellwerten 2, 3 und 10 (Eigene Darstellung).

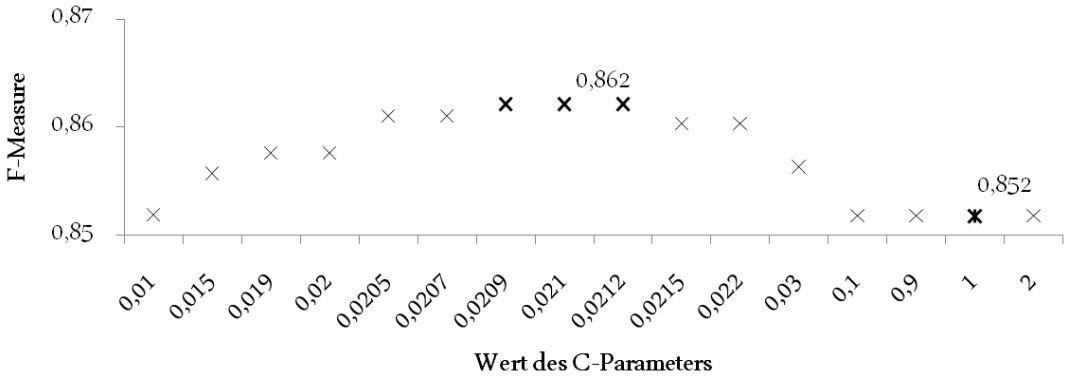

Abbildung 6.10. Ergebnisse der Optimierung des C-Parameters für SVM mit polynomialem Kernel, Nutzung des Porter-Stemming-Algorithmus und Feature-Selektion mit DF, Schwellwert 2 (Eigene Darstellung).

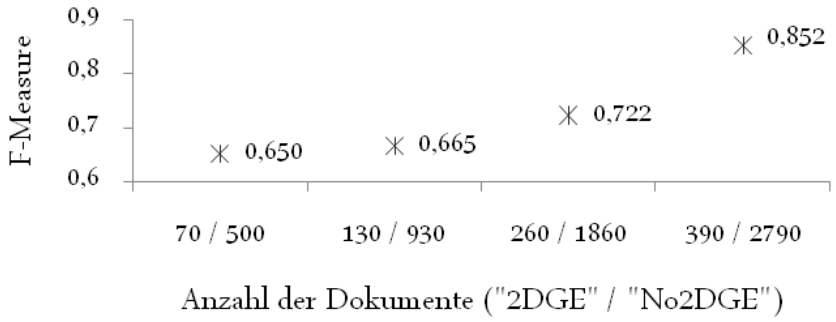

Abbildung 6.11. Auswirkung der Größe der Instanzmenge auf F-Measure für die relevante Klasse bei Verwendung von Support Vector Machines mit polynomialem Kernel und vorangegangenem Stemming sowie Feature-Auswahl mit Document Frequency, Schwellwert 2 (Eigene Darstellung).

7. Evaluierung

In Abschnitt 6.2 wurde die Evaluierung der modellbildenden Verfahren erläutert. In diesem Kapitel wird im ersten Abschnitt evaluiert, inwieweit das Ergebnis

- **neuartig** (in Bezug auf bereits vorhandenes Wissen),
- **valide** (für die Anwendung auf neue Daten),
- **verständlich** und
- **nützlich** für den Endanwender

ist [Vgl. Cle10, S. 9]. Anschließend erfolgt im zweiten Teil eine Analyse des durchgeführten Data-Mining-Prozesses.

7.1. Erfolgsbeurteilung

Support Vector Machines wurden bereits in mehreren Arbeiten des Text Mining erfolgreich verwendet (Vgl. Abschnitt 6.1.3). Der positive Einfluss der Feature-Auswahl mittels Document Frequency wurde ebenfalls in anderen Anwendungsfällen beobachtet (Vgl. Abschnitt 5.3). **Neuartig** ist – soweit bekannt – die Klassifikation von PubMed-Publikationen nach der eingesetzten Methode und unabhängig vom Forschungsgebiet. Die Klassifikation nach dem Forschungsgebiet ist vergleichsweise häufig (Vgl. Abschnitt 7.2). Autoren konzentrieren sich auf die Präsentation ihrer Forschungsergebnisse. Der Einsatz von Methoden ist dabei Mittel zum Zweck, nicht aber Schwerpunkt von Publikationen. Aufgrund der weniger ausführlichen Erläuterung – ausgenommen sind Arbeiten mit Fokus auf Verfahrensbeschreibungen bzw. -erweiterungen – der Methoden, ist die Klassifikation nach diesen schwieriger: Weniger Features sind vorhanden, die Indizien für das jeweilige Verfahren sind. Features, die auf das Forschungsthema deuten – z. B. Names der untersuchten Organismen, Krankheiten o. Ä. – sind deutlich häufiger und erschweren damit die Klassifikation nach der Methode.
Die Auswahl der Instanzmenge wurde unter Berücksichtigung der tatsächlichen Verteilung durchgeführt (Vgl. Abschnitt 5.1.1). Zudem wurde eine vergleichsweise große Instanzmenge für die Evaluierung verwendet. Aus diesen Gründen kann davon ausgegangen werden, dass der Klassifikator **valide** für die Anwendung auf reale Daten geeignet ist.
Die manuelle Identifikation von Kunden durch Suche in PubMed benötigt viel Zeit. Insbesondere weil viele irrelevante Publikationen herausgesucht werden müssen (Vgl. Abschnitt 3.2.2). Der Anteil relevanter Publikationen kann durch Verwendung von Suchbegriffen erhöht werden. Einige Suchbegriffe führen zu einer Trefferliste mit einem relativ hohen Anteil von richtig-positiven Publikationen. Dieser Anteil kann deutlich größer sein als der, der mittels

Klassifikation erzielt werden kann. Allerdings ist dann die absolute Anzahl relevanter Publikationen deutlich geringer: Viele relevante Publikationen würden durch Einschränkung der Suche mit einem Suchbegriff nicht mehr berücksichtigt werden (Vgl. 5.1.1 sowie 6.2.3). Die deutliche Zeitersparnis bei gleichzeitig absolut mehr positiven Publikationen ist für die schnellere Identifikation von Kunden **nützlich**.
Ergebnis der Klassifikation ist die Einteilung der Publikationen in zwei Klassen. Diese sind den Endanwendern – den Vertriebsmitarbeitern von DECODON – ebenso wie PubMed bekannt und entsprechend gut **verständlich**.
Verständlichkeit und Nützlichkeit werden durch Einbettung der Klassifikation in eine Webanwendung zusätzlich erhöht. Sie wird im Kapitel „Entwicklung und Einsatz der Webanwendung 'LeadScout'“ erläutert.

7.2. Prozessanalyse

Beginn des Data-Mining-Prozesses war die **Untersuchung der Datenquelle**. Die Bedeutung und der zeitliche Aufwand für diesen ersten Teilschritt wurde anfänglich unterschätzt. Bei der zukünftigen Durchführung von Data-Mining-Projekten ist darauf zu achten, dieser Phase ausreichend Aufmerksamkeit zu widmen. Andernfalls – wie im vorliegenden Projekt geschehen – ist eine erneute Betrachtung zur Entscheidung über zu verwendende Attribute und Datenvorverarbeitungsmethoden zu einem späteren Zeitpunkt erforderlich.
Bei der Aufwandsschätzung ist im Text Mining der **Aufbau der Instanzmenge** nicht zu unterschätzen. In der Text-Mining-Forschung wird dieser Aspekt häufig unzureichend und meist gar nicht betont. Grund hierfür ist die Verwendung von bereits klassifizierten Sammlungen wie den Reuters-Daten. Sollen allerdings neue Datenquellen betrachtet oder neue Klassen in bekannten Quellen gefunden werden, ist der manuelle Aufwand hoch: In der vorliegenden Arbeit waren etwa fünfzig Stunden notwendig, da in einigen Fällen Volltexte angesehen werden mussten. Hilfreich war die **parallel zum Data-Mining-Prozess verlaufende Entwicklung der Webanwendung**: Die Anwendung konnte genutzt werden, um die manuell klassifizierten Publikationen mit zwei Mausklicks in einer Datenbank abzulegen.
In Abschnitt 5.1.1 wurde erläutert, dass zunächst einige Tests mit einer relativ kleinen Instanzmenge durchgeführt wurden. Die Größe dieser Menge und die Klassenverteilung darin führten zu einer äußerst geringen Aussagekraft über die Performanz bei Anwendung auf neue Daten. Einerseits war diese erste Evaluierung sinnvoll, um Daten und Verfahren besser kennenzulernen. Andererseits musste relativ viel Zeit für wenige Erkenntnisse investiert werden. In zukünftigen Projekten sollte gleich zu Beginn eine **möglichst große Instanzmenge** – mit einer an die tatsächliche Verteilung angenäherten Klassenverteilung – aufgebaut werden.
Ein weiterer erfolgsentscheidender Bestandteil eines Data-Mining-Prozesses ist die **Dokumentation**. Aufgrund der Vielzahl verfügbarer Methoden und der unterschiedlich guten Performanz gleicher Klassifikatoren auf unterschiedlichen Daten sollten möglichst viele Kombinationen getestet werden (Vgl. 6). Diese Forderung führt zur Generierung gewaltiger Datenmengen. Deren Ergebnisse und die Erzeugung dieser müssen genauestens dokumentiert werden, um doppelte Arbeit zu vermeiden.

8. Entwicklung und Einsatz der Webanwendung „LeadScout“

Die Webanwendung “LeadScout“ bietet folgende Funktionen:

1. **Identifikation von Leads**

   Als relevant klassifizierte Publikationen und deren Autoren – die zu identifizierenden potentiellen Kunden – können angesehen werden. Zudem wird geprüft, ob die potentiellen Kunden besonders vielversprechend sind. Die zeitaufwendige Suche in PubMed entfällt.

2. **Übertragung von Leads in die Adressdatenbank von DECODON**

   Der Anwender von LeadScout kann Leads als neue Kontaktobjekte in der Adressdatenbank anlegen lassen. Ist eine Organisationszugehörigkeit aus einer Publikation verfügbar, kann diese ebenfalls übertragen werden. Zeitintensives Kopieren und Einfügen der Kontaktdaten – wie bei der manuellen PubMed-Suche – entfällt.

3. **Einholen von Informationen über Kontaktpersonen**

   Die Publikationen eines neuen Kontakts können angesehen werden. In bipartiten Graphen aus Publikationen und Autoren können auch gemeinsame Arbeiten mit Kunden und die Anzahl der Publikationen aus der relevanten Klasse ermittelt werden. Dies vereinfacht die Entwicklung von Verkaufsstrategien – z. B. durch Nutzung des Empfehlungsmarketings.

4. **Erweiterung der Datenbasis**

   Weitere Publikationen werden aus PubMed extrahiert. Diese können zur Identifikation von Leads und zur kontinuierlichen Vergrößerung der Trainingsmenge verwendet werden. Letzteres wird voraussichtlich den Anteil der True Positives erhöhen (Vgl. 6.2.3). Dadurch müssen Anwender mittelfristig noch weniger Zeit in das Herausfiltern von irrelevanten Publikationen bei der Lead-Suche investieren. Voraussetzung ist, dass auch Klassifikationsergebnisse der als nicht-relevant klassifizierten Publikationen regelmäßig überprüft werden. Mit der erweiterten Trainingsmenge wird der Klassifikator stetig weiter trainiert.

5. **Evaluierung von Datenvorverarbeitung, Feature-Selektion und Klassifikation**

   Die Möglichkeit der Evaluierung ist aus zweierlei Gründen Bestandteil von LeadScout. Einerseits waren diese Funktionen hilfreich bei der Ermittlung der verbleibenden Feature-Zahlen bei Anwendung der Vorverarbeitungsmethoden. Auch kleine Tests

während der Modellbildung waren dank der Webanwendung einfacher durchführbar. Andererseits erleichtert dieser Teil die Übertragung von LeadScout auf zukünftige Text-Mining-Projekte.

Diese Funktionalitäten unterstützen den betriebswirtschaftlichen Nutzen nicht. Sie werden daher nicht weiter erläutert.

Zunächst wird im folgenden Abschnitt ein Überblick über die Architektur von LeadScout gegeben. Anschließend werden in Abschnitt 8.2 die beiden erstgenannten Funktionsbereiche mit Anwendungsszenarien illustriert.

8.1. Architektur

LeadScout wurde in der objektorientierten Programmiersprache Python, Version 2.6, geschrieben. Als Web-Framework wurde Django, Version 1.2.1, ausgewählt. Sowohl Python als auch Django wurden aufgrund ihrer weiten Verbreitung und der guten Dokumentation gewählt. Neben dem freien Python-Buch „Dive into Python“ von Mark Pilgrim [Pil04] und dem Django-Buch „The Definitive Guide to Django“ von Holovathy und Kaplan-Moss [HKM09] existieren zahlreiche Online-Dokumentationen, Weblogs und Foren (Vgl. [PSF10], [DSF10], [Dja10]).
Ein weiterer Vorteil von Django ist die Anbindung an eine Datenbank mittels objektrelationaler Mapper. Die Mapper übernehmen die Datenübertragung zwischen dem Objekt und der Datenbank. Die Anwendung der Datenbankabfragesprache Standard Query Language (SQL) ist nicht erforderlich [Vgl. Fow03, S. 165][1].
Für die Datenhaltung wird SQLite verwendet. Diese plattformunabhängige Programmbibliothek implementiert ein relationales Datenbanksystem – abgelegt in einer einzigen Datei. SQLite benötigt keinen externen Server. Die Benutzung ist leicht erlernbar, die Bibliothek einfach zu installieren und zu warten [Vgl. SD10].
Django realisiert das Prinzip der Trennung der Verantwortlichkeiten („separation of concerns“) mittels des Musters **Model-View-Controller (MVC)**:

- **Model**

  Die Datenebene wird durch mehrere Model-Klassen realisiert. Alle Beschreibungen der Daten, deren Attribute und Beziehungen werden hinterlegt. Django ermöglicht die Erstellung beliebig vieler Models. Ein Model ist eine Python-Klasse, die Methoden zur Realisierung der Geschäftslogik enthalten kann. Alle Models werden in „models.py“ abgelegt. Exemplarisch zeigt Abbildung 8.1 das sehr einfache Model „Classification“ aus der im Rahmen dieser Arbeit entwickelten Webanwendung.

  Classification hat lediglich zwei Attribute und eine Methode. Dank des objektrelationalen Mappers wird automatisch eine Tabelle „Classification“ in der verwendeten Datenbank – hier: SQLite – angelegt, die die beiden Attribute als Spalten hat. Zusäzlich wird – falls diese Eigenschaft nicht ausdrücklich einem anderen Attribut zugewiesen wurde – ein Primärschlüssel namens „id“ angelegt und ebenfalls automatisch gefüllt.

[1] Bei Änderungen der Models ist in einigen Szenarien die Aktualisierung mit SQL erforderlich. Dies kommt in der Praxis allerdings sehr selten vor und kann daher vernachlässigt werden.

```
class Classification(models.Model):
    result = models.CharField(max_length=10)
    classifier = models.CharField(max_length=10)

    def __unicode__(self):
        return unicode(self.result) + u' [' + unicode(self.classifier) + u']'
```

Abbildung 8.1. Das Classification-Model (Ausschnitt aus der Webanwendung).

- **View**

  Der Begriff View ist im MVC-Muster ein Synonym für die Präsentationsschicht. Welche Daten in welcher Form präsentiert werden, wird hier festgelegt. In Django wird dieser Teil mit Views und Templates implementiert. Erstere sind in der Datei „views.py“ gespeicherte Methoden. Sie greifen ggf. auf externe Module zu, um die Datenausgabe korrekt zu steuern.[2] Letztere sind mit einer Django-eigenen Template-Sprache dynamisch-gestaltbare HTML-Seiten. Jede View gibt entweder eine HTTP-Response unter Nutzung eines Templates zurück oder leitet die Anfrage an eine andere View weiter.

- **Controller**

  Als Controller wird der Teil der Anwendung bezeichnet, der die Steuerung der Präsentation übernimmt. In Django wird dies durch URLConf realisiert: In der Datei „urls.py“ werden alle verfügbaren URLs mit der jeweils auszuführenden View hinterlegt [Vgl. HKM09, S. 73f].

In Django ist die oberste Hierarchieebene eine Website. Der Name der Website ist im vorliegenden Projekt **„lfsite“**. Der Name setzt sich aus „lf“ – als Abkürzung für den ursprünglichen Namen von LeadScout „LeadFinder“ – und dem Zusatz „site“ – als Kurzform für Website – zusammen. Als Django-Website muss lfsite die folgenden Module beinhalten:

Manage Manage wird von Django automatisch erzeugt und bedarf keinerlei Bearbeitung. Das Modul beinhaltet unter anderem Methoden zum Starten der Anwendung oder zum Synchronisieren der Datenbank mit den Models-Klassen.

Settings In diesem Modul werden Variablen gesetzt, die das Arbeiten mit Django erleichtern. Beispielsweise werden in dem Dictionary „DATABASES“ alle Informationen zur verwendeten Datenbank angegeben.

Tests Dieses Modul beinhaltet die Tests für die Klassen und Funktionen von LeadScout.

Urls Urls enthält die URL-Konfiguration – URLConf genannt. In der Liste „urlpatterns“ werden die verfügbaren URLs mit den aufzurufenden Views gespeichert. Alternativ zur Angabe einer View kann eine Umleitung auf eine andere URL definiert werden.

[2] Die Entwickler von Django weisen darauf hin, dass diese Aufteilung je nach Verständnis von MVC anders sein kann. Djangos Views könnten – aufgrund der enthaltenen Logik zur Steuerung der anzuzeigenden Daten – auch Controller im Sinne des MVC-Modells sein [Vgl. HKM09, S. 73].

Views Dieses Modul enthalt alle Views, die unabhängig von einer Django-Anwendung sind. Im Beispiel von LeadScout ist dies lediglich eine View, die auf die View zum Ausgeben der Startseite von Leadretrieval weiterleitet.

Zusätzlich können jeder Website beliebig viele Anwendungen zugeordnet werden. Lfsite hat bisher genau eine Anwendung: **„Leadretrieval“** enthält neben anwendungsspezifischen Models, Views und Templates, sogenannte Templatetags, die unterstützende Logik für die Darstellung unter Nutzung von Templates bereitsstellen.
Weiterhin werden Module verwendet, die in dem Paket „businesslogic“ abgelegt sind. Neben der Klasse ADBFormHandler zum Anlegen von neuen Kontakten in der Adressdatenbank der DECODON GmbH beinhaltet businesslogic die Module:

Classification Zur Klassifikation der Publikationen werden in diesem Teil von LeadScout Weka-Subprozesse[3] aufgerufen. Weka (verwendet in Version 3.6.3) ist eine Sammlung von Algorithmen für das Data Mining, geschrieben in Java. Die Weka-Klassen sind – unter der GNU General Public License – frei verfügbar. Weka wurde aufgrund der guten Dokumentation sowie der Fülle an anwendbaren Verfahren ausgewählt. Die Verfahren Naïve Bayes, IBk – die Weka-Variante für k-Nearest-Neighbors – und Support Vector Machines mit polynomialem, normalisiertem polynomialen und RBF-Kernel wurden verwendet.

Error Dieses Modul enthält Unterklassen der Python-Klasse „Exception“, die für das Werfen von Exceptions verwendet werden.

Evaluation Hier werden Klassen implementiert, die der Evaluierung von Datenvorverarbeitung, Feature-Selektion und Klassifikation dienen.

Graphs Das Modul „Graphs“ ermöglicht die Darstellung von Publikationen und deren Autoren in bipartiten Graphen sowie deren Ausgabe in der Graph Modelling Language (GML).

Parsing Dieses Modul von LeadScout übernimmt das Analysieren der XML-Dokumente aus PubMed, die HTTP- bzw. Batch-Requests sowie die Aufgaben im Zusammenhang mit den HTML-Formularen der Adressdatenbank. Für XML- und HTML-Parsing wird das Paket „BeautifulSoup“ verwendet. Die HTML-Formulare werden unter Nutzung des Moduls „ClientForm“ ausgelesen und gefüllt.

Preprocessing Die Aufgaben der Datenvorbereitung und Featureselektion werden hier abgearbeitet. Der Porter-Stemming-Algorithmus aus dem „Natural Language Toolkit (NLTK)“-Paket für Python wird benutzt.

[3] Der kNN-Klassifikator wurde zunächst unabhängig von Weka implementiert und für erste Tests (Vgl. Abschnitt 5.1.1) genutzt. Aufgrund der besseren Geschwindigkeit der Weka-Variante und zur Vereinheitlichung der Struktur im Modul „Classification“ wurde schließlich zur Verwendung der Weka-Implementation übergegangen.

Die Module benutzen externe, freie Python-Module für das Parsing („BeautifulSoup“), das Arbeiten mit HTML-Formularen („ClientForm“) und das Auslesen von HTML-Tabellen („table_parser“). Abbildung 8.2 zeigt eine Übersicht über die wichtigsten Bestandteile von LeadScout. In einigen Fällen – beispielsweise bei views und templates – wurde auf eine vollständige Aufzählung zugunsten der Übersichtlichkeit verzichtet. In diesen Fällen sind die weiteren Teile durch drei Punkte angedeutet.

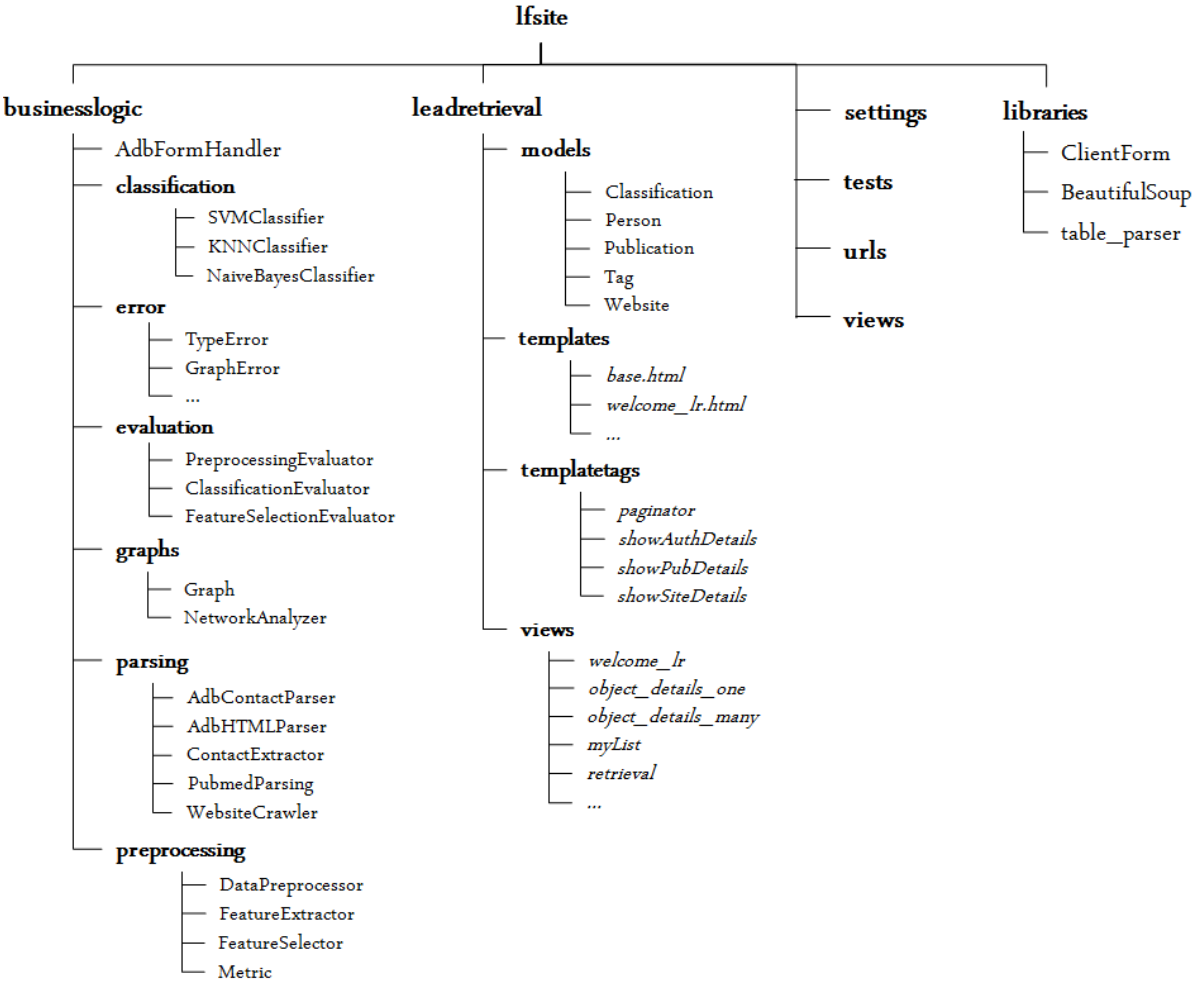

Abbildung 8.2. Übersicht über die Module von LeadScout. Paket- bzw. Modulnamen sind fett gedruckt. Klassen beginnen stets mit einem Großbuchstaben. Funktionen und HTML-Templates sind kursiv gedruckt. Die HTML-Templates haben zudem die Endung „.html“ (Eigene Darstellung).

Wird die Unterscheidung nach Django-spezifischen und externen Modulen berücksichtigt, kann die Architektur von LeadScout wie in Abbildung 8.3 darstellgestellt werden. Neben eigens entwickelten Modulen, die mit einigen externen Paketen zusammenarbeiten, werden alle notwendigen Bestandteile einer Django-Website genannt.

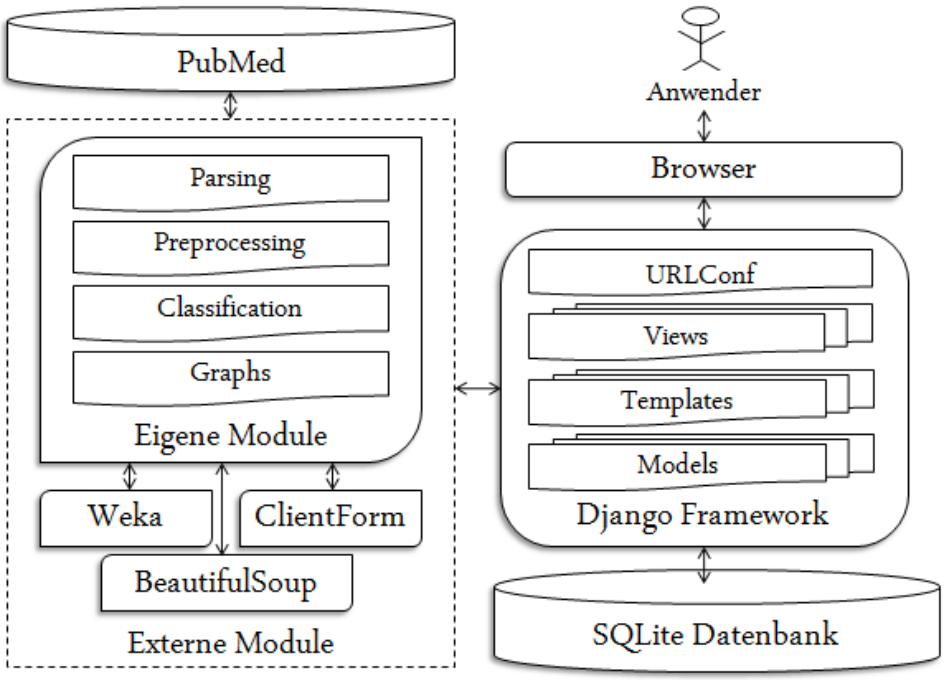

Abbildung 8.3. Architektur von LeadScout. Dargestellt sind – zugunsten der Übersichtlichkeit – lediglich die wichtigsten Module (Eigene Darstellung).

8.2. Anwendungsszenarien und Ausblick

8.2.1. Identifikation potentieller Kunden

Abbildung 8.4 zeigt die Startseite von LeadScout. Die Aufmerksamkeit des Anwenders wird auf die beiden zentralen Anwendungsmöglichkeiten gerichtet:

1. **„Latest relevant Publications “ und Bild von einer Nadel im Heuhaufen**

   Dieser Link führt zu einer Liste von mit SVM als relevant (Klasse „2DGE“) klassifizierten Publikationen (Vgl. Abbildung 8.5). Die Klassifikation darf noch nicht durch einen Anwender bestätigt oder geändert worden sein.

2. **„Authors tagged with ’LEAD’“ und Bild von einem Goldklumpen**

   Die durch Klick auf diesen Link erscheinende Liste von Autoren ist das Ergebnis der Kombination der Data-Mining-Resultate mit den Erfahrungswerten aus dem Vertrieb: Ein Autor bekommt automatisch den Tag „LEAD“ und ist damit als besonders vielversprechender potentieller Kunde gekennzeichnet, wenn er

   - noch kein Kunde ist,
   - wenigstens einmal Autor einer Publikation der Klasse „2DGE“ war,
   - wenigstens eine Publikation mit einem Kunden veröffentlicht hat (Kunde kann als Empfehler fungieren) und

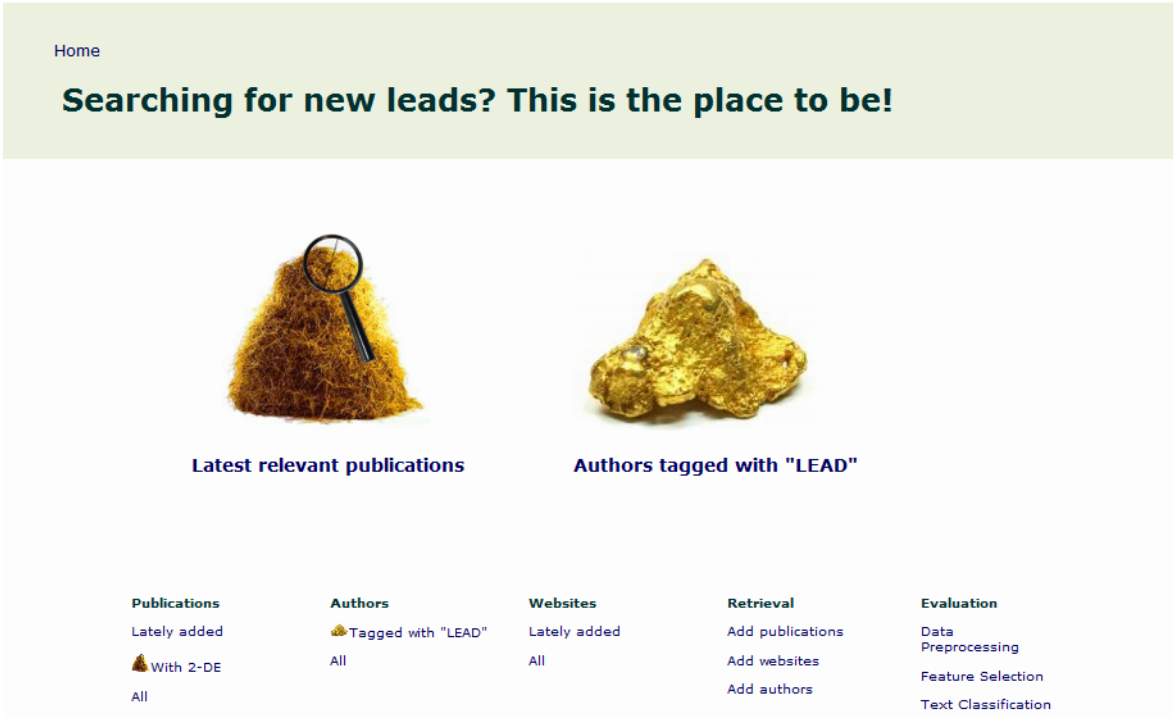

Abbildung 8.4. Startseite von LeadScout (Screenshot).

- wenigstens einmal ohne einen Kunden publiziert hat (Ausschluss von Kollegen eines Kunden, die nur mit diesem 2D-Gele machen).

Will der Anwender die relevanten Publikationen durchsuchen, klickt er auf den ersten Link. Eine Liste von Veröffentlichungen – getrennt durch eine Leerzeile – erscheint. In jedem dieser Absätze ist links die PMID (als Link zu PubMed), das Erstellungsdatum, der Titel, die Autoren, die Affiliation, MeSH-Begriffe, ein Link zur Detailansicht sowie ein Button zum Löschen der Publikation verfügbar. Rechts kann ein beliebiger Tag angehängt oder die Klassifikation geändert bzw. bestätigt werden.
Die Klassifikation ist ein Django-Model mit den Attributen „result“ und „classifier“. Ersteres ist die Klasse („2DGE“, “No2DGE“), letzteres gibt an, ob die Publikation vom Anwender („user“) oder von Support Vector Machines („svm“) klassifiziert wurde. Wird die Klassenzugehörigkeit vom Anwender abgelehnt bzw. bestätigt, erhält die Publikation zusätzlich die Klassifikation „2DGE[user]“ bzw. „No2DGE[user]“. Die Publikation taucht nicht mehr in der Liste der neuen relevanten Veröffentlichungen auf.
Abbildung 8.5 zeigt eine Liste relevanter Publikationen. Trotz der guten Performanz von SVM ist eine manuelle Verifikation erforderlich. Die Klassifikation der ersten Publikation mit der PMID „20212449“ kann mit den angezeigten MeSH-Begriffen und Titel nicht sicher bestätigt werden. Der Aufruf der Detailseite ist erforderlich. Auf dieser Seite wird zusätzlich der Abstract angezeigt (Vgl. Abbildung 8.6). Ein Link zu einem Graphen aus der Publikation und deren Autoren (Link „Graph“) sowie zu einem erweiterten Graphen („Graph plus one“), der zusätzlich alle anderen Publikationen der Coautoren enthält, ist verfügbar. Die Klassifikation

Abbildung 8.5. Ausschnitt aus einer Liste relevanter Publikationen aus LeadScout. Zu jeder Publikation werden Klasse, PMID, CreationDate, Titel, Autoren, Affiliation und MeSH-Begriffe angegeben. Die Nadel mit dem Heuhaufen vor der Angabe der Klasse der Publikation weist drauf hin, dass die Publikation relevant ist (Screenshot).

kann durch Lesen des Abstracts verifiziert werden. Der Anwender kann sich auf deren Autoren konzentrieren. Autorennamen sind Links zur Detailseite des Autors. Im Beispiel ist die Email-Adresse von „Marco Prunotto“ angegeben. Er scheint für den Inhalt hauptverantwortlich zu sein. Seine Detailseite ist aufzurufen (Vgl. Abbildung 8.7). Dort ist vermerkt, dass dieser Autor noch nicht in der Adressdatenbank der DECODON GmbH ist. Kunden würden mit der Notiz „Author is already a customer.“ angezeigt werden. Dieser Hinweis fehlt. „Marco Prunotto“ wurde also erfolgreich als potentieller Kunde identifiziert.
Alternativ zum Durchsuchen der relevanten Publikationen besteht die Option, durch Klick auf den zweiten Link auf der Startseite eine Liste der Autoren mit dem Tag „LEAD“ aufzurufen. Dies ist eine Teilmenge der Menge aller Autoren relevanter Publikationen. Die zusätzlich zu erfüllenden Kriterien sollen besonders interessante Leads sichtbar machen. Sie wurden durch Analyse der Netzwerke von Neukunden aufgestellt. Abbildung 8.8 zeigt eine Seite der Lead-Liste.
Der bereits im ersten Beispiel entdeckte Wissenschaftler „Marco Prunotto“ hat den Tag „LEAD“. Auf seiner Detailseite (Vgl. 8.7) ist dies durch das Bild eines Goldklumpens und

Abbildung 8.6. Detailseite für Publikation „20212449“ (Screenshot).

den Satz „Marco Prunotto is a lead!“ schnell zu erkennen. Abbildung 8.9 zeigt einen bipartiten Graphen aus seinen Publikationen (Quadrate mit jeweiliger PMID beschriftet) und Coautoren (Kreise mit PubMed-Namen beschriftet). Relevante Publikationen sind in grün, Kunden in rot dargestellt. Marco ist in orange hervorgehoben. Marco ist noch kein Kunde. Publikation „19540948“ wurde in Zusammenarbeit mit dem Kunden „Andrea Urbani“ erstellt. Publikation „20212449“ ist der Klasse „2DGE“ zugeordnet. Marco Prunotto ist demnach ein äußerst vielversprechender Lead.

8.2.2. Übertragung von Leads in die Adressdatenbank

Identifizierte potentielle Kunden können aus der Webanwendung heraus in die Adressdatenbank von DECODON überführt werden. Auf der Detailseite des im vorherigen Abschnitt identifizierten Leads „Marco Prunotto“ sind dessen gespeicherte Publikationen zu sehen. Die angegebene Organisationszugehörigkeit kann dem Autor durch Klick auf den Button „Update with Selected Affiliation“ zugeordnet werden. Abbildung 8.10 zeigt das erscheinende Formular. Im Beispiel sind Land und Email-Adresse noch in einer Zeile. Sie können nicht korrekt zugeordnet werden. Dies kann durch Auswahl von „SPLIT“ und Klick auf den Button „Submit“ korrigiert werden. Die Maske sieht nun wie in Abbildung 8.11 zu sehen aus. Wurden

alle Attribute zugeordnet und mit Klick auf den Button „Submit“ bestätigt, erscheint erneut die Detailseite für „Marco Prunotto“. Die selektierten Daten sind nun unter „Current Author Information“ angegeben. Mit Klick auf den Button „Add Author 1964 to ADB“[4] wird der Autor zur Adressdatenbank hinzugefügt. Auf der Detailseite erscheint nun der Hinweis „Marco Prunotto is already in ADB: link“. Der Autor Marco Prunotto wurde erfolgreich in die Adressdatenbank von DECODON überführt.

8.2.3. Informationsgewinn durch Ansicht von bipartiten Publikationen-Autoren-Graphen

Ein weiteres Anwendungsszenario ist das Ansehen von bipartiten Graphen aus Autoren und deren Publikationen und Coautoren, um Zusammenhänge erkennen zu können. In der Webanwendung werden zu jedem Autor nicht nur dessen Publikationen als Liste angezeigt (Vgl. Abbildung 8.12). Auch bipartite Graphen sind unter “See Author’s Network“ aufrufbar: Der Link „Graph“ öffnet einen bipartiten Graphen in dem freien Graph-Editor „yEd“. In dem Graph ist der Autor mit seinen Veröffentlichungen und seinen Coautoren zu sehen.[5] Unter „Graph plus one“ kann ein Graph aufgerufen werden, der zusätzlich alle Publikationen der Coautoren sowie deren Coautoren beinhaltet. Wird ein neuer Kontakt geknüpft, kann dieser in der Webanwendung gesucht und sein Graph angesehen werden. Verbindungen zu bekannten Personen oder sogar Kunden können sehr schnell entdeckt und bei der Planung der Verkaufsstrategie berücksichtigt werden.
Das Hinzufügen weiterer Publikationen – sofern in PubMed vorhanden – ist über „Fetch more publications“ möglich. Der Graph wird entsprechend erweitert. Zusätzlich zu den Publikationen des Autors, für den der Graph aufgerufen wird, können die Publikationen seiner Coautoren angezeigt werden. Abbildung 8.13 zeigt beispielhaft einen solchen Graphen („Graph plus one“). Die Vertriebsmitarbeiter von DECODON können mithilfe der Graphen schnell einen Überblick über die Tätigkeit eines Kontaktes erlangen. Gemeinsame Arbeiten mit Bestandskunden sind schnell erkennbar. Verkaufsstrategien können mit diesem Wissen optimiert werden.

8.2.4. Erweiterung der Datenbasis

Die Verbesserung der Performanz von Support Vector Machines und das kontinuierliche Finden neuer potentieller Kunden erfordert die stetige Erweiterung der Datenbasis. Über den Link „Add Publications“ (in der Fußzeile auf jeder Seite der Website unter „Retrieval“) ist ein Formular zum Hinzufügen von Publikationen aufrufbar. Zwischen den Varianten

- Eingabe eines Suchbegriffs für PubMed sowie der Anzahl der hinzuzufügenden Publikationen aus der Trefferliste,

[4] ADB ist die unter DECODON-Mitarbeitern verwendete Abkürzung für Adressdatenbank. Die Nennung der ID anstelle des Namens des Autors ist aus technischen Gründen erforderlich.

[5] Hat der Autor Namensvetter kann nicht eindeutig der Name dem jeweiligen Individuum zugeordnet werden (Vgl. Abschnitt 4.1 sowie 4.2). Daher werden Autoren mit identischen Namen behandelt als wären sie dieselbe Person, d.h. alle Publikationen von Namensvettern erscheinen in demselben Graphen. Aufgabe des Anwenders bleibt die kritische Betrachtung der Daten.

- Angabe der PMID oder
- Auswahl aller bzw. einer Gruppe von Kunden deren neueste Publikationen gesucht werden sollen.

kann gewählt werden.
Die erste Variante ist die empfohlene für das standardmäßige Erweitern: Als Suchbegriff sollte „electrophoresis“ eingegeben werden. Der Anwender kann auswählen, ob die gefundenen Publikationen als „2DGE“, „No2DGE“, gar nicht oder durch SVM klassifiziert werden sollen. Letzteres ist empfehlenswert, um die Liste der neuen und relevanten Publikation (Vgl. Abschnitt 8.2.1) zu erweitern. Ist das Einlesen und Klassifizieren abgeschlossen, erscheint eine Seite mit den PMIDs aller neuen Publikationen – verlinkt mit ihrer Detailseite. Der Anwender kann diese nun nacheinander anschauen oder die Identifikation neuer Kunden fortsetzen. Die neuesten Einträge werden nun berücksichtigt.
Zunächst ist die wöchentliche Erweiterung um 200 Publikationen angedacht. Dies würde etwa 25 neuen relevanten Veröffentlichungen entsprechen (Vgl. Abschnitt 5.1.1 sowie 6.2.3). In Abhängigkeit von der Anwendungshäufigkeit und des daraus resultierenden Bedarfs sind Anzahl und Häufigkeit anzupassen.
Voraussetzung für die Optimierung der Performanz von Support Vector Machines ist, dass nicht nur die als relevant klassifizierten Publikationen angesehen werden. Verifikation der Klassifikationsergebnisse der vermeintlich irrelevanten Veröffentlichungen ist ebenso erforderlich. Anreize für die Anwender – beispielsweise in Form von Nutzungsrechten für die Bereiche der Anwendung, die primär der Identifkation von Leads dienen – sind zu schaffen.

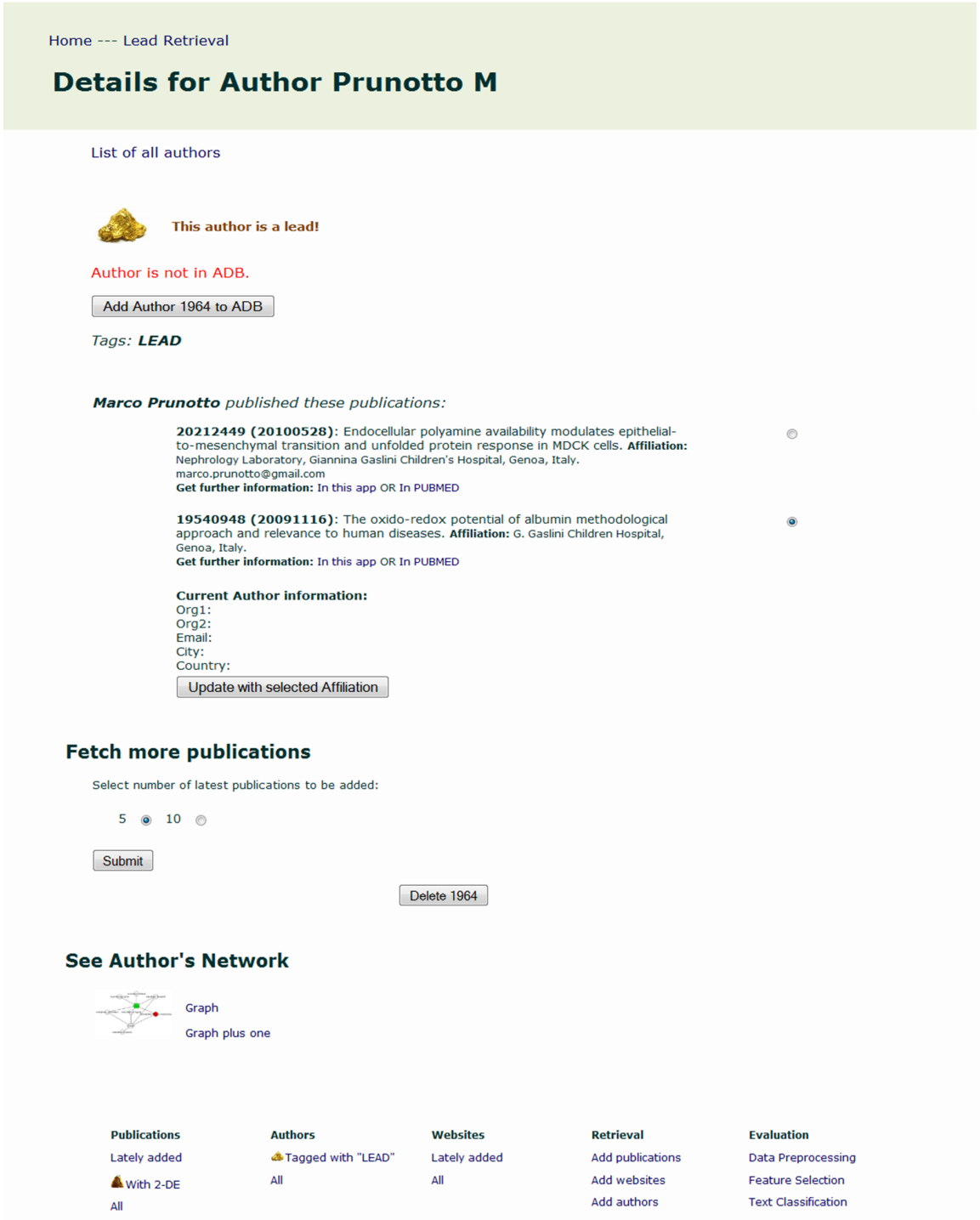

Abbildung 8.7. Detailseite für den Autor „Marco Prunotto“ (Screenshot).

Abbildung 8.8. Seite der Liste von Leads (Screenshot).

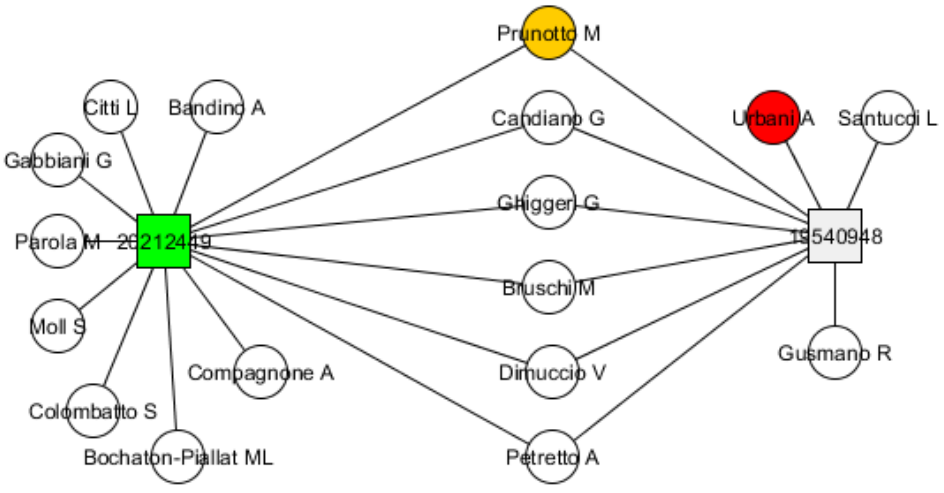

Abbildung 8.9. Graph aus Publikationen und Autoren (Eigene Darstellung).

Abbildung 8.10. Maske zur Auswahl der Attributwerte für ein zu erstellendes ADB-Objekt (Screenshot).

Abbildung 8.11. Maske zur Auswahl der Attributwerte für ein zu erstellendes ADB-Objekt (Screenshot).

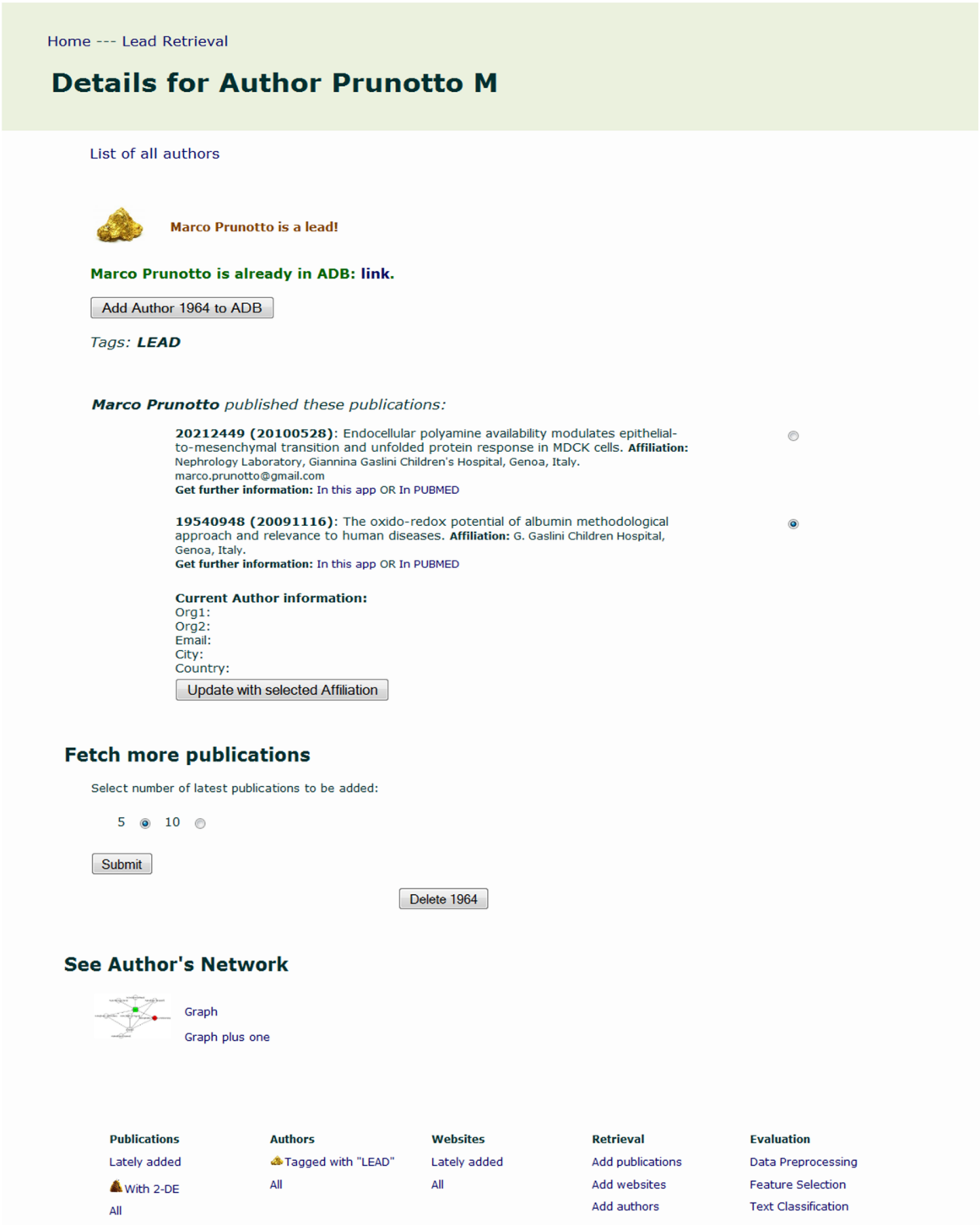

Abbildung 8.12. Detailseite für den Autor „Marco Prunotto“ (Screenshot).

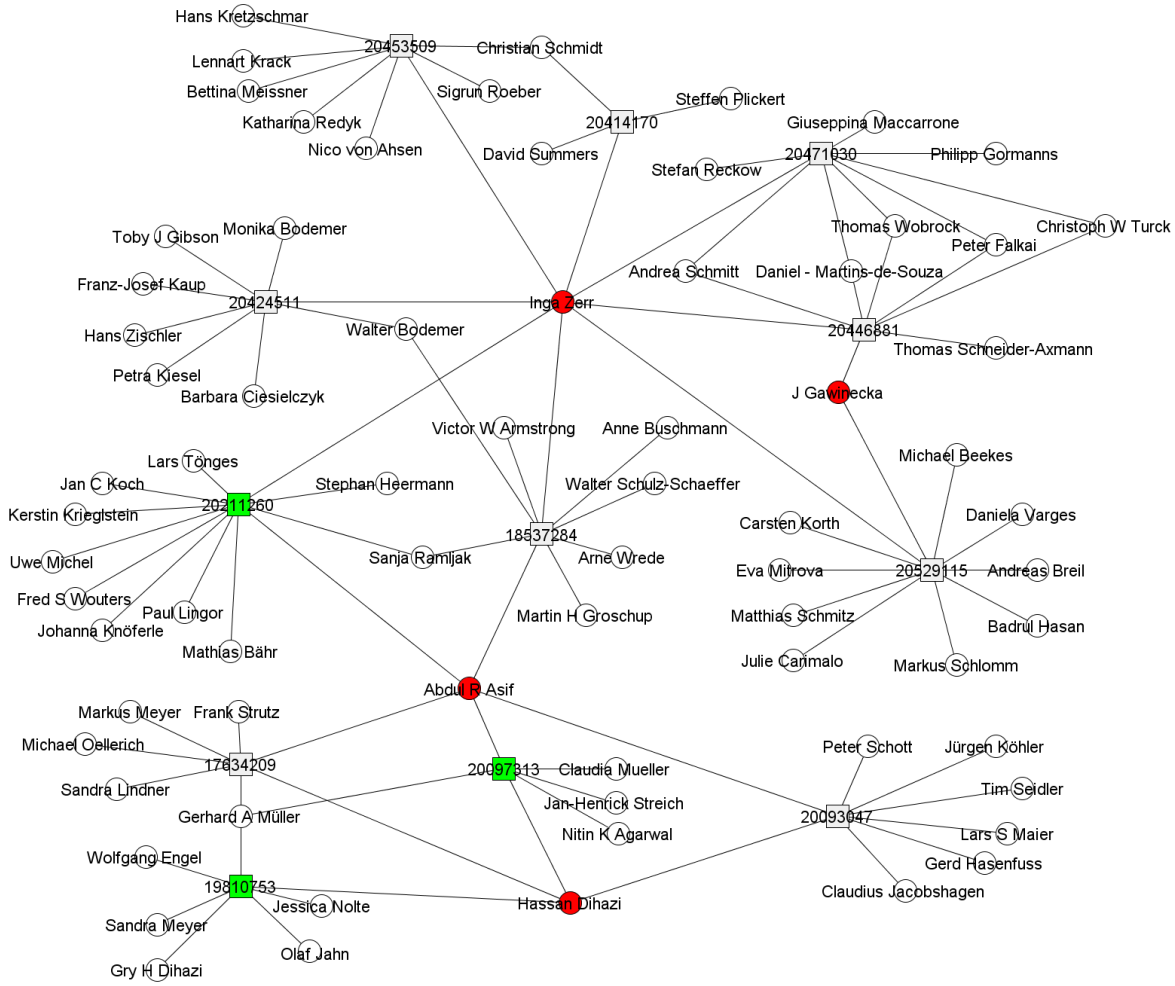

Abbildung 8.13. Darstellung von Autoren und Publikationen im bipartiten Graphen (Visualisierung einer in der Webanwendung erzeugten GML-Datei mit dem yEd Graph Editor)

9. Verwandte Arbeiten

Data Mining wird in den unterschiedlichsten Bereichen mit zunehmender Häufigkeit angewandt [Vgl. WF05, S. 22ff]. Viele freie Werkzeuge wie das in dieser Arbeit verwendete Weka sowie Rapidminer von Rapid-I oder auch KNIME von KNIME.com können benutzt werden. Von den beiden letztgenannten Anwendungen sind zudem Versionen mit erweiterten Funktionalitäten – z. B. zur Prozess- und Leistungsoptimierung – sowie mit Gewährleistungsansprüchen käuflich zu erwerben. Inzwischen bieten Unternehmen Data Mining auch als Dienstleistung an: Das Spin-Off-Unternehmen der Universität Leipzig „TextTech Informationsmanagement und Texttechnologie GmbH“ bietet – neben diversen Modulen zur Unterstützung des Text Mining, wie einem Werkzeug für die Lemmatisierung – Text-Mining-basierte Auftragsarbeiten an. Beispielsweise wurden Produktbeschreibungen klassifiziert, um diese automatisiert in Produktkataloge zu integrieren. Der manuelle Aufwand konnte um etwa 97 Prozent reduziert werden [Vgl. Gmb10b].
In Abschnitt 4.2 wurde bereits auf die große Bedeutung von PubMed in der Text-Mining-Forschung hingewiesen. Beispielhaft sei auf die Arbeiten von Sahay und Zaremba et al. hingewiesen: Sahay nutzte Support Vector Machines zum Herausfiltern von Publikationen aus dem Fachgebiet der Epidemiologie [Vgl. Sah]. Zaremba et al. zeigen, wie Verfahren des Text Mining zur automatisierten Extraktion von Forschungsergebnisse aus Publikationen verwendet werden können [Vgl. Zar+09].
GoPubMed ist ein kostenfreier Service der Transinsight GmbH aus Dresden (`http://www.gopubmed.com`). Publikationen werden basierend auf MeSH-Begriffen und GeneOntology (GO) durchsucht. Die Suchergebnisse können nach Forschungsgebiet, Autoren bzw. Organisationen, Veröffentlichungsort und -datum sortiert werden [Gmb10c]. „Electrophoresis, Gel, Two-Dimensional“ ist der MeSH-Begriff, der zu einer Trefferliste ohne falsch-positive Publikationen führt. Einerseits vereinfacht dies die Suche, da falsch-positive nicht herausgefiltert werden müssen. Andererseits entgehen dem Suchenden viele Treffer, da nur wenigen Publikationen MeSH-Begriffe zugeordnet werden (Vgl. Abschnitt 4.2.3). Zudem werden MeSH-Begriffe nach dem Schwerpunkt der Arbeit vergeben. Relevanten Publikationen wird nicht zwingend „Electrophoresis, Gel, Two-Dimensional“ zugeordnet. Stattdessen wird bei z. B. die untersuchte Erkrankung, der Organismus o. ä. betont: Den beiden ersten Publikationen zum Thema „Zwei-dimensionale Gelelektrophorese“ - die Arbeiten von [O'F75] und [Klo75], den Erfindern der Technik - wurden MeSH-Begriffe aber nicht „Electrophoresis, Gel, Two-Dimensional“ zugeordnet. Unter den neuesten 50 relevanten Publikationen von Kunden der DECODON GmbH ist nur 29 der MeSH-Begriff „Electrophoresis, Gel, Two-Dimensional“ zugeordnet [Stand: August 2010]. Die Untersuchung der ersten 20 Veröffentlichungen der Monate Januar 2009 bis einschließlich Juni 2010, die Treffer auf den Begriff „electrophoresis“ waren, ergab: 78 Prozent der positiv klassifizierten Publikationen hatten keinen MeSH-Term „Electrophoresis, Gel, Two-Dimensional“.

Abbildung 9.1. Startseite des Internet-Dienstes „biomedExperts“ (Screenshot).

Die Collexis Holding Inc. bietet mit **biomedExperts** einen kostenfreien Service zum Finden von Experten an. Die Startseite ist in Abbildung 9.1 zu sehen. Wissenschaftler können hier Experten aus ihrem Forschungsgebiet suchen. Die Vernetzung ist – wie in anderen sozialen Netzwerken – möglich [Vgl. Inc10]. Das Portal ermöglicht auch die Suche nach Experten für die zweidimensionale Gelelektrophorese. Allerdings ist die Nutzung zum Finden potentieller Kunden nicht sinnvoll: Die Suche nach Experten liefert lediglich die Namen der 50 derzeit weltweit führenden Wissenschaftlern in dem Gebiet. Sie sind DECODON bereits bekannt. Einige sind Kunden, andere werden regelmäßig mit den Instrumenten des Direktmarketing kontaktiert.

Auch im Marketing ist der Einsatz von Data Mining äußerst vielfältig (Vgl. [TSK06, S. 1f], [WF05, S. 26ff]). Ling und Li beschreiben beispielsweise, wie Kundendaten zur Ermittlung der Zielgruppe für ein Mailing klassifiziert werden können [Vgl. LLL98]. Das Themengebiet „Business Intelligence“ wäre ohne Data Mining schwer denkbar.

Trotz des vielfältigen Einsatzes von Text-Mining-Verfahren und der häufigen Verwendung zur Unterstützung des Marketing ist die Anwendung auf PubMed-Einträge zum vereinfachten Finden von potentiellen Kunden – soweit bekannt – noch nicht Thema einer anderen Arbeit gewesen.

10. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden zunächst die Grundlagen des Data Mining sowie die betriebswirtschaftlichen Grundlagen betrachtet. Dadurch wurde in die Thematik eingeführt und die Basis für die weiteren Ausführungen geschaffen. Nach einer Erläuterung der Datenvorverarbeitung wurden die evaluierten modellbildenden Verfahren – k-Nearest-Neighbors, Support Vector Machines und Naive Bayes – beschrieben. Anschließend wurden diese Verfahren mit PubMed-Publikationen evaluiert. Dabei hat sich gezeigt, dass Support Vector Machines mit polynomialem Kernel die besten Resultate liefern. Um die Performanz dieses Klassifikators weiter zu verbessern, wurden die Parametereinstellungen angepasst. Eine Verbesserung konnte erzielt werden. Die Evaluierung der gewonnenen Erkenntnisse hat gezeigt, dass die Ergebnisse neuartig, verständlich, nützlich und valide sind. Ebenso wurde dargestellt, wie die Ergebnisse des Data Mining in einer Webapplikation nutzbar gemacht wurden. Die Anwendung ist für den Einsatz im Tagesgeschäft geeignet. Die Betrachtung verwandter Arbeiten konnte schließlich zeigen, dass die vorliegende Arbeit einzigartig ist.
Die Nutzung der Resultate in der Webanwendung LeadScout kann in Zukunft weiter optimiert werden. Insbesondere sind folgende Optimierungen geplant:

1. **Entwicklung weiterer Regelkombinationen für die Optimierung der Lead-Beurteilung.**

   Die in Abschnitt 8.2.1 beschriebene Bewertung von Leads durch Anwendung einer Kombination aus erfahrungsbasierten Regeln sollte weiter ausgebaut werden. Beispielsweise könnten neue Regeln durch die weitere Analyse der Netzwerke von Bestandskunden (Vgl. Beispiel in 8.2.3) entwickelt werden.

2. **Automatisierte Zuordnung von Kontaktdaten zu den jeweiligen Autoren.**

   Bisher ist die manuelle Zuordnung der Affiliation – sie ist lediglich in ihre Bestandteile zerlegt – notwendig. Der freie Webservice „OpenCalais“ dient der automatisierten Textanalyse: OpenCalais liefert für jede Eingabe möglichst passende Metadaten. Beispielsweise würde die Eingabe „University of California“ unter anderem zur Ausgabe, dass eine Organisation im Text genannt ist, führen. Die in PubMed-Publikationen angegebene Affiliation könnte in die durch Kommata getrennten Bestandteile zerlegt und mit OpenCalais den Attributen der Autoren automatisch zugeordnet werden. Erste Tests mit OpenCalais waren vielversprechend. Die Verwendung erfordert allerdings noch weitere Optimierung: Z. B. werden einige Organisationen nicht zuverlässig als solche erkannt. Zudem ist OpenCalais lediglich für Bezeichnungen in englischer Sprache geeignet. Einige Autoren geben die Affiliation in ihrer Sprachen an. Die Verwendung von OpenCalais würde die Übersetzung erfordern.

3. **Automatisierte Suche nach Kontaktdaten von Leads.**

   Leider geben einige Autoren keine Kontaktdaten oder Organisationszugehörigkeit (Affiliation) in ihren Publikationen an. Um diese Autoren als potentielle Kunden ansprechen zu können, ist die manuelle Internet-Recherche nach Kontaktdaten erforderlich. Zukünftig soll diese Suche automatisiert werden. Angedacht ist die Nutzung eines Webservices, der die Recherche übernimmt und Links zu Seiten mit Kontaktdaten zurückliefert.

   Ein solcher Webservice zum Finden von Websites von Personen war „Ahoy – The Homepage Finder“. Leider wurde Ahoy seit 2000 nicht weitergeführt und ist inzwischen veraltet [Vgl. SLE10]. Arbeiten wie die von Fatemieh et al. lassen baldige Verfügbarkeit neuer Services nach dem Vorbild von Ahoy erhoffen: Fatemieh et al. stellen eine Anwendung vor, die unter Nutzung der Suchmaschine „Google“ und aufgestellter Heuristiken Homepages findet [Fat+05].

   Ebenso ist die Evaluierung von Online-Diensten wie „123people“ geplant. Ziel ist die Nutzung eines Webdienstes, um dem Anwender Kontaktdaten von Autoren präsentieren zu können. Der Anwender kann diese – z. B. durch Ansehen der Website – prüfen und ggf. übernehmen.

4. **Suche nach Autoren mit besonderen Rollen im Netzwerk.**

   Die Nutzung der Graphen soll hierfür weiter ausgebaut werden: Mit der sozialen Netzwerkanalyse sollen Autoren identifiziert werden, die im Netzwerk besondere Rollen innehaben. Welche Autoren sind Konnektoren, verbinden z. B. aufgrund ihrer methodischen Kompetenz Autoren unterschiedlicher Fachrichtungen? Welche Autoren haben ähnliche Rollen wie umsatzstarke Kunden? Vertriebsstrategien könnten mit diesem Wissen optimiert werden: Autoren mit wahrscheinlicher Multiplikatorenwirkung würden beispielsweise zuerst kontaktiert werden. Zusätzlich skizzieren Mimno und McCallumn in [MM07] eine Vorgehensweise zum Finden einflussreicher Autoren aus großen Datenmengen. Auch die Evaluierung dieser Methode scheint vielsprechend zu sein.

5. **Erweiterung um die Klassifikation von Websites.**

   Die Klassifikation von Websites ist mittels Wiederverwendung der bereits für die PubMed-Publikationen eingesetzten Verfahren angedacht. Django-Models und entsprechende Views sind bereits vorhanden. Zu einigen Autoren wurden bereits Links zu deren Websites aus der Adressdatenbank extrahiert. Erste Webcrawling-Versuche führten zu einem sehr hohen Anteil irrelevanter Seiten (z. B. über 99 Prozent bei Start auf einer Universitäts-Homepage und Verfolgen aller Links). Zur Optimierung des Crawlings sollten Heuristiken für die Link-Verfolgung entwickelt werden.

Zudem ist – mit der Eroberung weiterer Nischen im Bereich der Lebenswissenschaften – die Anpassung zum Finden von Publikationen mit anderen verwendeten Methoden denkbar. Ebenso ist vorstellbar, die Anwendung anderen Bioinformatik-Unternehmen zur Verfügung zu stellen.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Es wurden keine anderen als die angegebenen Quellen und Hinweise verwandt. Die vorliegende Arbeit wurde bisher noch keiner anderen Prüfungsbehörde vorgelegt und noch nicht veröffentlicht.

Wismar, 15. Januar 2011 .....

A. Anhang

A.1. PubMed Stop words

Tabelle A.1 listet alle für PubMed von der NLM angegebenen stop words auf:

A	a, about, again, all, almost, also, although, always, among, an, and, another, any, are, as, at
B	be, because, been, before, being, between, both, but, by
C	can, could
D	did, do, does, done, due, during
E	each, either, enough, especially, etc
F	for, found, from, further
H	had, has, have, having, here, how, however
I	i, if, in, into, is, it, its, itself
J	just
K	kg, km
M	made, mainly, make, may, mg, might, ml, mm, most, mostly, must
N	nearly, neither, no, nor
O	obtained, of, often, on, our, overall
P	perhaps, pmid
Q	quite
R	rather, really, regarding
S	seem, seen, several, should, show, showed, shown, shows, significantly, since, so, some, such
T	than, that, the, their, theirs, them, then, there, therefore, these, they, this, those, through, thus, to
U	upon, use, used, using
V	various, very
W	was, we, were, what, when, which, while, with, within, without, would

Tabelle A.1. PubMed Stop words – alphabetisch sortiert (Quelle: [NCB10b]).

A.2. Ergebnisse der Klassifikation

Eine Übersicht über die durchschnittlichen F-Measures nach 10-facher Kreuzvalidierung für alle evaluierten Verfahrens- und Parameterkombination ist in der Tabelle in Abschnitt A.2.1 dokumentiert. Die durchschnittlichen Erfolgsraten sind in Abschnitt A.2.2 gelistet.

A.2.1. F-Measures

	Naïve	kNN			SVM			
					PK			
	Bayes	k=1	k=21	k=101	p = 1	p = 2	NPK	RBF
Manuell	0,774	0,778	0,773	0,778	0,813	0,796	0,838	0,657
All	0,286	0,000	0,000	0,000	0,835	0,776	0,814	0,797
All_W	0,289	0,000	0,000	0,000	0,836	0,736	0,800	0,798
All_S	0,322	0,005	0,000	0,005	0,851	0,803	0,846	0,830
All_SW	0,322	0,000	0,000	0,000	0,836	0,785	0,827	0,823
DF2	0,285	0,010	0,000	0,010	0,839	0,786	0,808	0,800
DF2_W	0,287	0,010	0,000	0,010	0,828	0,753	0,799	0,803
DF2_S	0,323	0,015	0,000	0,025	0,852	0,816	0,844	0,836
DF2_SW	0,322	0,015	0,000	0,020	0,833	0,796	0,823	0,824
DF3	0,284	0,010	0,000	0,010	0,836	0,789	0,810	0,806
DF3_W	0,287	0,010	0,000	0,010	0,831	0,765	0,802	0,803
DF3_S	0,323	0,020	0,000	0,035	0,849	0,817	0,846	0,836
DF3_SW	0,323	0,015	0,000	0,025	0,836	0,803	0,823	0,832
DF10	0,277	0,005	0,000	0,015	0,827	0,803	0,818	0,818
DF10_W	0,275	0,024	0,000	0,024	0,826	0,779	0,807	0,817
DF10_S	0,302	0,048	0,000	0,058	0,848	0,830	0,849	0,840
DF10_SW	0,309	0,053	0,000	0,063	0,830	0,816	0,832	0,833
DF100	0,473	0,275	0,019	0,005	0,478	0,721	0,730	0,524
DF100_W	0,456	0,319	0,005	0,005	0,478	0,724	0,726	0,521
DF100_S	0,450	0,346	0,053	0,005	0,478	0,733	0,744	0,519

	Naïve	kNN			SVM			
					PK			
	Bayes	k=1	k=21	k=101	p = 1	p = 2	NPK	RBF
DF100_SW	0,449	0,315	0,044	0,005	0,478	0,728	0,740	0,517
TF*IDF80	0,524	0,005	0,000	0,000	0,743	0,417	0,399	0,525
TF*IDF80_W	0,514	0,005	0,000	0,000	0,723	0,380	0,299	0,522
TF*IDF80_S	0,524	0,005	0,000	0,000	0,738	0,457	0,442	0,508
TF*IDF80_SW	0,522	0,005	0,000	0,000	0,700	0,425	0,406	0,505
TF*IDF75	0,504	0,005	0,000	0,005	0,698	0,473	0,481	0,526
TF*IDF75_W	0,502	0,000	0,000	0,000	0,736	0,440	0,439	0,520
TF*IDF75_S	0,534	0,005	0,000	0,005	0,747	0,523	0,494	0,518
TF*IDF75_SW	0,523	0,008	0,000	0,008	0,744	0,467	0,458	0,510
TF*IDF66	0,534	0,015	0,000	0,015	0,698	0,467	0,514	0,534
TF*IDF66_W	0,516	0,082	0,000	0,082	0,759	0,493	0,481	0,530
TF*IDF66_S	0,527	0,010	0,005	0,010	0,706	0,552	0,536	0,521
TF*IDF66_SW	0,521	0,037	0,005	0,037	0,754	0,533	0,503	0,516
TF*IDF52	0,496	0,318	0,000	0,318	0,766	0,627	0,621	0,531
TF*IDF52_W	0,491	0,247	0,000	0,247	0,772	0,602	0,558	0,531
TF*IDF52_S	0,510	0,328	0,000	0,328	0,759	0,667	0,625	0,518
TF*IDF52_SW	0,520	0,227	0,005	0,227	0,754	0,625	0,582	0,518
TF*IDF18	0,415	0,420	0,100	0,420	0,677	0,730	0,752	0,514
TF*IDF18_W	0,427	0,444	0,104	0,444	0,732	0,728	0,743	0,514
TF*IDF18_S	0,420	0,418	0,101	0,418	0,706	0,759	0,761	0,514
TF*IDF18_SW	0,443	0,393	0,138	0,393	0,731	0,738	0,742	0,512

Tabelle A.2. F-Measures aller Varianten nach 10-facher Kreuzvalidierung.

A.2.2. Erfolgsraten

	Naïve	kNN			SVM			
					PK			
	Bayes	k=1	k=21	k=101	p = 1	p = 2	NPK	RBF
Manuell	92,054	94,249	95,031	94,249	95,695	94,591	96,059	92,121
All	38,679	87,736	87,736	87,736	96,038	95,126	95,629	95,346
All_W	39,434	87,736	87,736	87,736	96,038	94,434	95,409	95,283
All_S	48,679	87,767	87,736	87,767	96,352	95,566	96,132	95,881
All_SW	48,834	87,736	87,736	87,736	96,006	95,252	95,849	95,723
DF2	38,145	87,799	87,736	87,799	96,101	95,283	95,535	95,377
DF2_W	38,931	87,799	87,736	87,799	95,849	94,717	95,377	95,377
DF2_S	48,616	87,830	87,736	87,862	96,384	95,786	96,164	96,006
DF2_SW	48,616	87,830	87,736	87,862	95,881	95,440	95,786	95,755
DF3	38,176	87,799	87,736	87,799	96,069	95,314	95,566	95,503
DF3_W	39,088	87,799	87,736	87,799	95,912	94,906	95,472	95,346
DF3_S	48,647	87,862	87,736	87,925	96,289	95,786	96,195	96,006
DF3_SW	48,836	87,830	87,736	87,893	95,943	95,535	95,786	95,912
DF10	35,692	87,767	87,736	87,830	95,849	95,535	95,692	95,723
DF10_W	35,377	87,893	87,736	87,893	95,786	95,094	95,503	95,597
DF10_S	45,346	88,050	87,736	88,082	96,226	96,038	96,289	96,069
DF10_SW	45,189	88,082	87,736	88,145	95,818	95,755	95,975	95,849
DF100	79,748	86,604	87,862	87,736	91,164	93,742	94,088	91,887
DF100_W	78,836	88,145	87,736	87,736	91,164	93,742	93,962	91,855
DF100_S	79,843	85,409	87,987	87,736	91,164	93,742	94,214	91,824
DF100_SW	79,308	85,000	87,956	87,736	91,164	93,711	94,088	91,792
TF*IDF80	80,849	87,767	87,736	87,767	94,308	90,912	90,723	91,887
TF*IDF80_W	81,685	87,767	87,736	87,767	94,182	90,535	90,314	91,855
TF*IDF80_S	81,195	87,767	87,736	87,767	94,245	91,352	90,975	91,667

	Naïve	kNN			SVM			
					PK			
	Bayes	k=1	k=21	k=101	p = 1	p = 2	NPK	RBF
TF*IDF80_SW	82,013	87,767	87,736	87,767	93,994	91,069	90,692	91,604
TF*IDF75	79,748	87,767	87,736	87,767	94,057	91,478	91,352	91,887
TF*IDF75_W	79,717	87,736	87,736	87,736	94,296	91,132	91,132	91,824
TF*IDF75_S	81,667	87,767	87,736	87,767	94,277	91,918	91,509	91,792
TF*IDF75_SW	82,107	87,799	87,736	87,799	94,245	91,352	91,186	91,667
TF*IDF66	81,164	87,830	87,736	87,830	93,994	92,107	91,761	91,950
TF*IDF66_W	81,604	87,736	87,736	87,736	94,528	91,667	91,447	91,887
TF*IDF66_S	81,258	87,799	87,767	87,799	94,120	92,138	91,887	91,824
TF*IDF66_SW	81,855	87,736	87,767	87,736	94,497	92,013	91,509	91,761
TF*IDF52	79,423	79,119	87,736	79,119	94,434	93,270	93,225	91,981
TF*IDF52_W	78,994	81,258	87,736	81,258	94,623	92,925	92,226	91,981
TF*IDF52_S	81,386	82,862	87,736	82,862	94,245	93,585	92,830	91,761
TF*IDF52_SW	81,667	81,478	87,767	81,478	94,120	93,082	92,296	91,761
TF*IDF18	73,176	88,365	88,365	88,365	93,396	93,711	94,340	91,761
TF*IDF18_W	75,440	88,019	88,239	88,019	93,648	93,553	94,088	91,792
TF*IDF18_S	75,629	86,226	88,302	86,226	93,082	94,214	94,371	91,761
TF*IDF18_SW	77,673	84,623	88,522	84,623	93,522	93,774	93,994	91,730

Tabelle A.3. Ermittelte Erfolgsraten nach 10-facher Kreuzvalidierung.

A.3. Installation von LeadScout

Zunächst sind folgende die nachfolgenden Anwendungen zu installieren:

- Python 2.6 mit den Paketen
  - NLTK
  - matplotlib
  - numpy
- Django 1.2.1
- SQLite 3.7.3
- Weka 3.6.3
- yED Graph Editor 3.5 oder 3.6

LeadScout selbst kann dann nach Kopieren des Ordners „lfsite“ auf den entsprechenden Rechner verwendet werden.

A.4. Übersicht über die Dokumente auf der beiliegenden CD

Auf der beiliegenden CD sind folgende Daten enthalten:

- Ordner „thesis“: Vorliegende Arbeit als TeXnicCenter-Projekt, inklusive der Arbeit im PDF-Format (Datei „thesis.pdf“).
- Ordner „Quellen“:
  - Datei „mybib.bib“: BibTex-Datei mit allen verwendeten Quellen.
  - Online-Quellen und zitierte Artikel aus Zeitschriften, benannt nach BibTex-ID.
- Ordner „lfsite“: LeadScout-Programmcode.
- Ordner „Resultate“:
  - Datei „ResultateKlassifikation.xls“: Tabellen mit allen Ergebnissen (Erfolgsrate, Precision, Recall, F-Measure) aus allen Iterationen der zehnfachen Kreuzvalidierung aller Verfahrenskombinationen.
  - Ordner „I1“, „I2“, ... „I10“: Ergebnisse aus allen Iterationen der zehnfachen Kreuzvalidierung (eine Datei je Iteration und Kombination) aller Verfahrenskombinationen (Bewertungsmaße und Konfusionsmatrizen).

Literatur

[Aiz00] Akiko Aizawa. “The feature quantity: an information-theoretic perspective of tfidf-like measures”. In: *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*. Athens, GR: ACM Press, New York, US, 2000, S. 104–111.

[Alp10] Ethem Alpaydin. *Introduction to Machine Learning (Second Edition)*. München, 2010.

[Ber+07] Matthias Berth u. a. “The state of the art in the analysis of two-dimensional gel electrophoresis images”. In: *Applied Microbiology Biotechnology* (2007), 1223–1243.

[Ber08] Richard A. Berk. *Statistical Learning from a Regression Perspective*. Philadelphia, Pennsylvania, USA, 2008.

[BFU08] Rainer Busch, Wolfgang Fuchs und Fritz Unger. *Integriertes Marketing: Strategie- Organisation- Instrumente*. Wiesbaden, 2008.

[BK10] Jean Beney und Cornelis H. A. Koster. “SVM Paradoxes”. In: *Perspectives of Systems Informatics* (2010), S. 86–97.

[Boc98] Jürgen Bock. *Bestimmung des Stichprobenumfangs. Für biologische Experimente und kontrollierte klinische Studien*. München, Wien, 1998.

[Bou+10] Remco R. Bouckaert u. a. *WEKA Manual*. Abgerufen am 12.10.2010. Hamilton, New Zealand, 2010. URL: `http://sourceforge.net/projects/weka/files/documentation/3.6.x/WekaManual-3-6-3.pdf/download`.

[Cam+07] Nathalie Camelin u. a. “Speech Mining in Noisy Audio Message Corpus”. In: *Proceedings of 10th International Conference on Spoken Language Processing (Interspeech 2007 ICSLP, Antwerp, Belgium)*. Avignon, France, 2007, 31–57.

[Cha+00] Pete Chapman u. a. *CRISP-DM 1.0. Step-by-step data mining guide*. Abgerufen am 12.10.2010. 2000. URL: `http://www.crisp-dm.org/CRISPWP-0800.pdf`.

[Che73] Herman Chernoff. “The Use of Faces to Represent Points in K-Dimensional Space Graphically”. In: *Journal of the American Statistical Association* 68, No. 342 (1973), S. 361–368.

[Che99] Michael R. Chernik. *Bootstrap Methods. A Practitioner’s Guide.* Diamond Bar, California, USA, 1999.

[Cle10] Jürgen Cleve. *Wissensmanagement: Wissensextraktion*. Skript für das Modul Wissensextraktion, Fernstudium Wirtschaftsinformatik. Wismar, 2010.

[Das+07] Anirban Dasgupta u. a. *Feature Selection Methods for Text Classification*. Abgerufen am 12.10.2010. 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.8428&rep=rep1&type=pdf.

[Dja10] DjangoSites.org. *djangosites - Latest Additions*. Abgerufen am 12.10.2010. 2010. URL: http://www.djangosites.org/.

[DRB10] Kerstin Denecke, Thomas Risse und Thomas Bähr. *Topic Classification Using Limited Bibliographic Metadata*. Abgerufen am 12.10.2010. 2010. URL: http://www.cacaoproject.eu/fileadmin/media/AT4DL/paper-03.pdf.

[DSF10] Django-Software-Foundation. *Django documentation*. Abgerufen am 12.10.2010. 2010. URL: http://docs.djangoproject.com/en/1.2/.

[Fat+05] Omid Fatemieh u. a. *Home Page Finder*. Abgerufen am 12.10.2010. 2005. URL: http://www.cs.uiuc.edu/homes/sfatemi2/documents/DB_Report.pdf.

[Fin08] Klaus-J. Fink. *Empfehlungsmarketing. Der Königsweg zur Neukundengewinnung*. Wiesbaden, 2008.

[For07] George Forman. "Feature Selection for Text Classification". In: *Computational Methods of Feature Selection*. 2007.

[Fow03] Martin Fowler. *Patterns of Enterprise Application Architecture*. Boston, MA, USA, 2003.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro und Padhraic Smyth. "From Data Mining to Knowledge Discovery in Databases". In: *AI Magazine* (1996), S. 37–54.

[FS07] Ronen Feldman und James Sanger. *Text Mining Handbook. Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA, 2007.

[Gmb10a] EMC Deutschland GmbH. *EMC-Studie: jährlich erzeugte Datenmenge steigt bis 2020 um Faktor 44*. Abgerufen am 12.10.2010. 2010. URL: http://germany.emc.com/about/news/press/2010/20100504-01.htm.

[Gmb10b] Texttech GmbH. *Klassifizierung elektronischer Produktkataloge*. Abgerufen am 12.10.2010. 2010. URL: http://www.texttech.de/Anwendungen/Produktkataloge/tabid/71/language/de-DE/Default.aspx.

[Gmb10c] Transinsight GmbH. *GoPubMed - Ontology-based Literature search - Help*. Abgerufen am 12.10.2010. 2010. URL: http://gopubmed.com/web/gopubmed/www/GoPubMed/Search/index.html.

[HCL03] Chih-Wei Hsu, Chih-Chung Chang und Chih-Jen Lin. *A Practical Guide to Support Vector Classification*. Abgerufen am 12.10.2010. 2003. URL: http://www.csie.ntu.edu.tw/~/cjlin/papers/guide/guide.pdf.

[Hel06] Michaela Hellerforth. "Praxisforum: Möglichkeiten des Markteintritts, Konkurrenzanalyse und Akquise". In: *Handbuch Facility Management für Immobilienunternehmen*. 2006, S. 527–567.

[Her07] Olaf Herden. "Data Mining". In: *Taschenbuch Datenbanken*. München: Thomas Kudraß, 2007.

[HJI06] DaRue A. Prieto und Haleem J. Isaaq. “Diagnostic Proteomics”. In: *Proteomics for Biological Discovery*. Frederick, Maryland, USA, 2006, S. 247–276.

[HKK99] Eui-Hong (Sam) Han, Karypis und Vipin Kumar. *Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification*. Abgerufen am 12.10.2010. 1999. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.8898&rep=rep1&type=pdf.

[HKM09] Adrian Holovathy und Jacob Kaplan-Moss. *The Definitive Guide to Django. Web Development Done Right. Second Edition*. New York, NY, USA, 2009.

[HM04] Tom Howley und Michael G. Madden. “The Genetic Evolution of Kernels for Support Vector Machine Classifiers”. In: *In 15th Irish Conference on Artificial Intelligence*. 2004.

[HNP05] Andreas Hotho, Andreas Nürnberger und Gerhard Paaß. “A Brief Survey of Text Mining”. In: *LDV Forum - Zeitschrift für Computerlinguistik und Sprachtechnologie* (2005), S. 19–62.

[Hol04] Heinrich Holland. *Direktmarketing*. München, 2004.

[HS09] Christopher D. Manning und Prabhakar Raghavan und Hinrich Schütze. *Introduction to information retrieval*. Cambridge, England, 2009.

[IM06] Waldemar Pförtsch und Indrajanto Müller. *Die Marke in der Marke: Bedeutung und Macht des Ingredient Branding*. Pfortzheim, 2006.

[Inc10] Collexis Holdings Inc. *Explore and expand your scientific network*. Abgerufen am 12.10.2010. 2010. URL: http://www.biomedexperts.com/Portal/AboutBME.aspx.

[Int10] Interbrand. *Best Global Brands. 2010 Rankings*. Abgerufen am 12.10.2010. 2010. URL: http://www.interbrand.com/de/knowledge/best-global-brands/best-global-brands-2008/best-global-brands-2010.aspx.

[JC08] Uwe Lämmel und Jürgen Cleve. *Künstliche Intelligenz (3., neu bearbeitete Auflage)*. Wismar, 2008.

[JEC06] Pavel Gromov und Irina Gromov und Julio E. Celis. “Proteomic Analysis by Two-Dimensional Polyacrylamide Gel Electrophoresis”. In: *Proteomics for Biological Discovery*. Copenhagen, Denmark, 2006, S. 19–46.

[Joa98] Thorsten Joachims. “Text Categorization with Support Vector Machines: Learning with many relevant features”. In: *Proceedings of ECML-98, 10th European Conference on Machine Learning*. Chemnitz, 1998, S. 137–142.

[KA01] Aleksander Kolcz und Joshua Alspector. “SVM-based Filtering of E-mail Spam with Content-specific Misclassification Costs”. In: *Proceedings of the Workshop on Text Mining (TEXTDM 2001*. 2001.

[KHP05] Hyunsoo Kim, Peg Howland und Haesun Park. “Dimension Reduction in Text Classification with Support Vector Machines”. In: *Journal of Machine Learning Research* 6, 2005 (2005), S. 33–53.

[KLM96] Leslie Pack Kaelbling, Michael Littman und Andrew Moore. “Reinforcement Learning: A Survey”. In: *Journal of Artificial Intelligence Research* 4 (1996), S. 237–285.

[Klo75] Joachim Klose. “Protein mapping by combined isoelectric focusing and electrophoresis of mouse tissues. A novel approach to testing for induced point mutations in mammals”. In: *Humangenetik* (1975), S. 231–243.

[KSZ05] Moshe Koppel, Jonathan Schler und Kfir Zigdon. “Determining an author’s native language by mining a text for errors”. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. Ramat-Gan, Israel, 2005, S. 624–628.

[KWA97] J. Kivinen, M. Warmuth und P. Auer. “The Perceptron algorithm versus Winnow: linear versus logarithmic mistake bounds when few input variables are relevant”. In: *Artificial Intelligence 97* (1997), S. 325–343.

[Leh09] Franz Lehner. *Wissensmanagement - Grundlagen, Methoden und technische Unterstützung*. München, 2009.

[LLL98] Charles Ling, Charles X. Ling und Chenghui Li. “Data Mining for Direct Marketing: Problems and Solutions”. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI Press, 1998, S. 73–79.

[Mak+99] John Makhoul u. a. “Performance Measures For Information Extraction”. In: *Proceedings of DARPA Broadcast News Workshop*. 1999, S. 249–252.

[Man08] Uwe Manschwetus. “Aspekte der Markenführung”. In: *Kulturmarketing*. München / Wien, 2008.

[MH06] Hans H. Bauer und Gregor Stokburger und Maik Hammerschmidt. *Marketing Performance: Messen - Analysieren Optimieren*. Wiesbaden, 2006.

[MI10] Institut für Mittelstandsforschung (IfM). *KMU-Definition des IfM Bonn*. Abgerufen am 12.10.2010. Bonn, 2010. URL: `http://www.ifm-bonn.org/index.php?id=89`.

[Mil10] MilwardBrown. *Top 100 Most valuable global brands 2010*. Abgerufen am 12.10.2010. 2010. URL: `http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/5781/pdf/imm5781.pdfhttp://www.millwardbrown.com/Libraries/Optimor_BrandZ_Files/2010_BrandZ_Top100_Report.sflb.ashx`.

[Mit97] Tom M. Mitchell. *Machine Learning*. New York, New York, USA, 1997.

[MM07] David Mimno und Andrew McCallum. “Mining a Digital Library for Influential Authors”. In: *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. Amherst, MA, USA, 2007, S. 105–106.

[MTJ06] Roman Tesar Michal Toman und Karel Jezek. *Influence of Word Normalization on Text Classification*. Abgerufen am 12.10.2010. 2006. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.6363&rep=rep1&type=pdf.

[NCB10a] NCBI. *PubMed Help*. Abgerufen am 12.10.2010. 2010. URL: http://www.ncbi.nlm.nih.gov/bookshelf/picrender.fcgi?book=helppubmed&part=pubmedhelp&blobtype=pdf.

[NCB10b] NCBI. *PubMed Stop Words*. Abgerufen am 12.10.2010. 2010. URL: http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=helppubmed&part=pubmedhelp&rendertype=table&id=pubmedhelp.T43.

[NIH10] NIH. *Fact Sheet - Medical Subject Headings (MeSH)*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/pubs/factsheets/mesh.html.

[NLM10a] NLM. *Alphabetical List of the MEDLINE/PubMed Elements and their Attributes*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/bsd/licensee/elements_alphabetical.html.

[NLM10b] NLM. *Fact Sheet. MEDLINE*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/pubs/factsheets/medline.html.

[NLM10c] NLM. *MEDLINE / PubMed Character Set*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/databases/dtd/medline_characters.html.

[NLM10d] NLM. *MEDLINE / PubMed XML Element Descriptions and their Attributes*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/bsd/licensee/elements_descriptions.html.

[NLM10e] NLM. *MEDLINE: Number of Citations to English Language Articles; Number of Citations Containing Abstracts*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/bsd/medline_lang_distr.html.

[NLM10f] NLM. *Required Data Elements in MedlineCitationSet*. Abgerufen am 12.10.2010. 2010. URL: http://www.nlm.nih.gov/bsd/licensee/elements_required.html.

[O'F75] Patrick H. O'Farrell. "High Resolution Two-Dimensional Electrophoresis of Proteins". In: *The Journal of Biological Chemistry* (1975), S. 4007–4021.

[OHS10] OHSU. *OHSUMED Test Collection*. Abgerufen am 12.10.2010. 2010. URL: http://ir.ohsu.edu/ohsumed/ohsumed.html.

[OKO07] Kirke Herrmann und Özgür Kurtulmus-Onigkeit. *Was ist Lemmatisierung und wie wird sie maschinell durchgeführt?* Abgerufen am 12.10.2010. 2007. URL: http://www.cl.uni-heidelberg.de/courses/archiv/ws06/morph/Lemmatisierung.ppt.

[PA01] Symposium Nlprs Pages und Akiko Aizawa. “Akiko Aizawa Linguistic Techniques to Improve the Performance of Automatic Text Categorization”. In: *In Proceedings 6th NLP Pac. Rim Symp. NLPRS-01*. Tokyo, Japan, 2001, S. 307–314.

[Pil04] Mark Pilgrim. *Dive into Python*. New York, NY, USA, 2004.

[Pla99] John C. Platt. “Fast training of support vector machines using sequential minimal optimization”. In: *Advances in kernel methods: support vector learning*. Cambridge, MA, USA, 1999, S. 185–208.

[Por80] Martin F. Porter. “An algorithm for suffix stripping”. In: *Program, 14, no. 3* (1980), S. 130–137.

[PSF10] Python-Software-Foundation. *The Python Community*. Abgerufen am 12.10.2010. 2010. URL: http://www.python.org/community/.

[Rab+10] Thierry Rabilloud u. a. “Two-dimensional gel electrophoresis in proteomics: past, present and future”. In: *Journal of Proteomics* (2010), S. 0–0.

[Reu10a] Reuters. *Reuters Corpus - Statistics*. Abgerufen am 12.10.2010. 2010. URL: http://about.reuters.com/researchandstandards/corpus/statistics/index.asp.htm.

[Reu10b] Reuters. *Reuters Corpus - What is available*. Abgerufen am 12.10.2010. 2010. URL: http://about.reuters.com/researchandstandards/corpus/available.asp.htm.

[Rü99] Bernhard Rüger. *Test- und Schätztheorie. Band I: Grundlagen*. München, 1999.

[Sah] Saurav Sahay. *Classifying PubMed documents using Support Vector Machines*. Abgerufen am 12.10.2010. Atlanta, GA, USA, URL: http://www.cc.gatech.edu/~/ssahay/CS7001-3.pdf.

[Sal97] Steven L. Salzberg. “On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach”. In: *Data Mining and Knowledge Discovery, 1, 1997* (1997), 317–328.

[Sch04] Alexander Schimansky. “Der Wert der Marke: Markenbewertungsverfahren für erfolgreiches Marketing”. In: München, 2004.

[Sch07] Christine Schauer. “Mitarbeiter als Markenbotschafter - Mit Leidenschaft die Marke vertreten”. In: *Die neue Macht des Marketing: Wie sie ihr Unternehmen mit Emotion, Innovation und Präzision profilieren*. Wiesbaden: Ralf T. Kreutzer und Wolfgang Merkle, 2007.

[SD10] SQL-Developers. *About SQLite*. Abgerufen am 12.10.2010. 2010. URL: http://www.sqlite.org/about.html.

[Seb01] Fabrizio Sebastiani. *Machine Learning in Automated Text Categorization*. Abgerufen am 12.10.2010. 2001. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.4939&rep=rep1&type=pdf.

[SLE10] Jonathan Shakes, Marc Langheinrich und Oren Etzioni. *Ahoy – The Homepage Finder*. Abgerufen am 12.10.2010. 2010. URL: http://www.cs.washington.edu/research/ahoy/.

[Tan05] Songbo Tan. "Neighbor-weighted K-nearest neighbor for unbalanced text corpus". In: *Expert Systems with Applications 28* (2005), 667–671.

[TM05] Franz-Rudolf Esch und Thorsten Möll. "Kognitionspsychologische und neuroökonomische Zugänge zum Phänomen Marke". In: *Moderne Markenführung - Grundlagen - Innovative Ansätze - Praktische Umsetzungen*. Wiesbaden, 2005.

[TS99] Ricky K. Taira und Stephen G. Soderland. *A Statistical Natural Language Processor for Medical Reports*. Abgerufen am 12.10.2010. 1999. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2232848/pdf/procamiasymp00004-1007.pdf.

[TSK06] Pang-Ning Tan, Michael Steinbach und Vipin Kumar. *Introduction to Data Mining*. New York, NY, USA, 2006.

[VCP09] Abhilash Venugopal, Raghothama Chaerkady und Akhilesh Pandey. "Application of mass spectrometry-based proteomics for biomarker discovery in neurological disorders". In: *Annals of Indian Academy of Neurology* (2009), S. 3–11.

[Vit04] Thorsten Vitt. *Seminar Advanced Data Warehousing: Text-Clustering*. Abgerufen am 12.10.2010. 2004. URL: https://zope.informatik.hu-berlin.de/forschung/gebiete/wbi/teaching/archive/ws0304/dwh/arbeiten/07_TVitt-paper.pdf.

[Voß+04] Werner Voß u. a. *Taschenbuch der Statistik*. München, 2004.

[WF05] Ian H. Witten und Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA, 2005.

[Win08] Peter Winkelmann. *Marketing und Vertrieb: Fundamente für die marktorientierte Unternehmensführung*. München, 2008.

[Wir05] Bernd W. Wirtz. *Integriertes Direktmarketing. Grundlagen - Instrumente - Prozesse*. Wiesbaden, 2005.

[WK09] W John Wilbur und Won Kim. "The Ineffectiveness of Within - Document Term Frequency in Text Classification". In: *Information Retrieval* (Okt. 2009), S. 509–525.

[WK91] Sholom M. Weiss und Casimir A. Kulikowski. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. San Francisco, CA, USA, 1991.

[WM07] Ralf T. Kreutzer und Wolfgang Merkle. *Die neue Macht des Marketing: Wie sie ihr Unternehmen mit Emotion, Innovation und Präzision profilieren*. Wiesbaden, 2007.

[Yan98] Yiming Yang. *An Evaluation of Statistical Approaches to Text Categorization*. Abgerufen am 12.10.2010. 1998. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.2516&rep=rep1&type=pdf.

[Yer03] F. Yergeau. *UTF-8, a transformation format of ISO 10646*. United States, 2003.

[YL99] Yiming Yang und Xin Liu. *A re-examination of text categorization methods*. Abgerufen am 12.10.2010. 1999. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.9519&rep=rep1&type=pdf.

[YP97] Yiming Yang und Jan O. Pederson. "A comparative study on feature selection in text categorization". In: *Proceedings of the 14th International Conference on Machine Learning*. Abgerufen am 12.10.2010. 1997, S. 412–420.

[Zar+09] Sam Zaremba u. a. "Text-mining of PubMed abstracts by natural language processing to create a public knowledge base on molecular mechanisms of bacterial enteropathogens". In: *BMC Bioinformatics* (2009). URL: http://www.biomedcentral.com/content/pdf/1471-2105-10-177.pdf.