Masterarbeit

Oliver Steenbuck

Datamining über Studienverlaufsdaten zum Zweck einer Erfolgsprognose

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Oliver Steenbuck

Datamining über Studienverlaufsdaten zum Zweck einer Erfolgsprognose

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachterin: Prof. Dr. Bettina Buth

Abgegeben am . April 2016

Oliver Steenbuck

Datamining über Studienverlaufsdaten zum Zweck einer Erfolgsprognose

Educational Datamining, Datamining, Metrik, Studenten, Prognose, Studienerfolg

Kurzzusammenfassung

In dieser Arbeit werden am Beispiel des Studiengangs Technische Informatik der Hochschule für Angewandte Wissenschaften Methoden des Educational Dataminings genutzt, um Studienerfolgsprognosen zu erstellen. Es wird gezeigt wie Studenten, aufbauend auf den Notendaten des ersten Studienjahrs, in eine erfolgreiche Klasse mit Abschluss und eine nicht erfolgreiche Klasse ohne Abschluss klassifiziert werden können. Auf dieser Klassifikation aufbauend wird untersucht ob ein Erkennen von erfolglosen Langzeitstudenten ebenfalls nach dem ersten Studienjahr möglich ist. Es werden Möglichkeiten diskutiert, statistische Daten zu nutzen um den individuellen Studienfortschritt eines Studenten genauer als derzeit möglich zu definieren.

Oliver Steenbuck

Datamining on past student data to generate predictions about future academic success

Educational Datamining, Datamining, Metric, Students, Prediction, Academic success

Abstract

In this thesis, methods from the field of Educational Datamining are used to generate forecasts of future academic success on the example of the degree course in Technische Informatik at the Hamburg University of Applied Sciences. This work shows how students can be classed into successfull (will have a degree in the future) and not successfull (will have no degree in the future) based on the grades achieved in the first year of study. Expanding on this there is discussion on the topic of classing students into a group of unsuccessfull longtime students. Finally options of using statistical data to give individual students better understanding of their current progress towards a degree are shown.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Listings

Teil I.

Einführung

1. Einleitung

Im Zuge des Bologna-Prozesses mit der Umstellung auf Bachelor-/Masterstudiengänge tritt auch die Frage der Studierbarkeit immer mehr in den Vordergrund. Studierbarkeit ist von der *CHE* Hochschulranking[1] definiert als:

> „... die Vollständigkeit des Lehrangebots hinsichtlich der Studienordnung, die Zugangsmöglichkeiten zu Lehrveranstaltungen, die Abstimmung des Lehrangebots auf die Prüfungsordnung, die Prüfungsorganisation und die Transparenz des Prüfungssystems."

(cheStudierbarkeit)

Ein Resultat der Studierbarkeit ist also die theoretische Möglichkeit, in Regelstudienzeit mit dem Studium abzuschließen. Gesamtgesellschaftlich gewinnt der Begriff der Studierbarkeit mit zunehmendem Fachkräftemangel in der Gesellschaft eine größere Bedeutung. Um frühzeitig einen Einblick in zukünftig zu erwartende Absolventenzahlen zu liefern, will diese Arbeit einen Beitrag zur Studienerfolgsprognose geben. Hier wird dazu exemplarisch der Studiengang Technische Informatik an der Hochschule für Angewandte Wissenschaften Hamburg betrachtet.

1.1. Motivation

Diese Arbeit beschäftigt sich mit Educational Dataming (siehe Kapitel 2.1), um aufbauend auf der Analyse von Studienverläufen Prognosen über zukünftige Studienerfolge abzugeben. Ihre Motivation stammt zum einen aus der persönlichen Erfahrung des Autors und seiner Überzeugung, dass Studieren durch die Anwendung statistischer Mittel erfolgreicher gestaltet werden kann, und zum anderen aus der Erkenntnis, dass die Gesellschaft und der Arbeitsmarkt mehr Informatiker benötigen, als derzeit erfolgreich ausgebildet werden. Hier kann eine Steigerung der Quote von Studenten, die ihr Studium erfolgreich abschließen, einen Beitrag zur Reduktion des Fachkräftemangels liefern. Auf einer anderen Ebene bieten Studienerfolgsprognosen die Möglichkeit, die knappen Ressourcen der Hochschule, die für

[1] http://www.che-ranking.de

Tutorien und Unterstützung zur Verfügung stehen, genauer auf die Zielgruppe zu fokussieren[2].

1.2. Ziele

Ziel dieser Arbeit ist es, aufbauend auf einem Auszug aus dem Studierendeninformationssystem (*Stisys*) der Hochschule für Angewandte Wissenschaften Hamburg mit den Mitteln des **Educational Dataminings** exemplarisch **Studienerfolgsprognosen** zu erstellen und zu validieren. Hierzu wird auf den **Datamining-Prozess** des Knowledge Discovery in Databases (KDD, siehe Kapitel (3.1)) zurückgegriffen. Es werden Überlegungen zur Generalisierbarkeit der gewonnenen Erkenntnisse sowohl über die Zeit auf den betrachteten Studiengang als auch auf andere Studiengänge angestellt. Übergreifendes Ziel dabei ist es, Möglichkeiten aufzuzeigen und zu validieren, um den **Studienplanungsprozess** zu unterstützen. Diese Unterstützung des Planungsprozesses findet auf den beiden unterschiedlichen Ebenen des Studenten und des Departments statt. Auf der Ebene des einzelnen Studenten soll diesem eine präzisere Auskunft über seinen Studienfortschritt gegeben werden, als es derzeit auf Basis der erreichten Credit Points möglich ist. Auf der Ebene des Departments entstehen belastbarere Aussagen über die prognostizierten Abschlussquoten und damit die Studierbarkeit des Studienganges.

1.3. Abgrenzung

In dieser Arbeit werden **keine Handlungsanweisungen** gegeben, vielmehr werden nur Daten und Schlussfolgerungen gezeigt. Die konkrete Verwendung der gewonnenen Erkenntnisse muss in den Gremien der **Selbstverwaltung** angeregt und beschlossen werden. Die Behandlung datenschutzrechtlicher Bedenken ist nicht Teil dieser Arbeit, hier kann lediglich ein Denkanstoß gegeben werden. Die hier gezeigten Algorithmen zur Auswertung können auch nicht direkt auf die vorhandenen Systeme[3] übertragen werden, hierzu wäre die Datenanbindung weiterer Systeme aus der Hochschulumgebung notwendig, deren Daten für diese Arbeit manuell integriert wurden. Die Datenquellen, die in diese Arbeit eingeflossen sind, sind zum überwiegenden Teil **quantitativer** Art (Notendaten, statistische Immatrikulierungsdaten etc...) insofern können alle Aussagen dieser Arbeit auch nur als quantitative Ergebnisse verstanden werden. Es wird das *Wie* und das *Wann* des Studierens betrachtet,

[2] Die Definition der Zielgruppe ist eine hochschulpolitische Frage, die aus dieser Arbeit ausgeklammert wird.
[3] Stisys

und das *Warum* größtenteils ausgeklammert. Es ist anzunehmen, dass eine qualitative Untersuchung von Gründen des erfolgreichen bzw. erfolglosen Studierens weitere Hinweise zur Erhöhung der Studienerfolgsquoten geben kann.

1.4. Gliederung

Diese Arbeit gliedert sich in drei Teile. Dies sind Einleitung, Hauptteil und Schluss. Die **Einleitung** zeigt die Motivation für diese Arbeit (Kapitel 1.1), ihre Ziele (Kapitel 1.2) und was nicht Ziel ist (Abgrenzung, siehe Kapitel 1.3) und die Gliederung. Der zweite große Teil der Einleitung ist der **Forschungsstand** (Kapitel 2), in dem auf den Begriff des Educational Datamining (2.1) eingegangen wird, um dann die verwendeten statistischen Verfahren (2.2) vorzustellen und die verwendeten Entwicklungsumgebungen (2.3) zu beschreiben.

Der Hauptteil beinhaltet die **Analyse** (3) mit ihren grob gegliederten Schwerpunkten Prozessmodelle(3.1), Rahmenbedingungen (3.2) und Ethische Fragen (3.4). Auf die Analyse aufbauend folgt die Beschreibung des **Systemdesigns und der Implementierung** (Kapitel 4), hier werden zuerst der Datenfluss und mit ihm die Komponenten des Systems (4.1) gezeigt. Darauffolgend der Prozess der Datenverdichtung (4.3), d.h. wie der Stisys-Export in die für diese Arbeit verwendeten Datentypen transformiert wurde. Abschließend wird auf die Datenbereinigung (4.4) und die die manuelle Auswertung unterstützende Webapplikation (4.5) eingegangen. Die **Evaluierung** (Kapitel 5) zeigt schließlich die wesentlichen statistischen Ergebnisse dieser Arbeit. Einleitend wird detailliert auf die Ergebnisse der Studienverlaufsprognose nach dem ersten Studienjahr mit den beiden Schwerpunkten Abschluss/kein Abschluss(5.1) und langes erfolgloses Studium (5.2) eingegangen. Es folgen diverse weitere Kennzahlen zum betrachteten Studiengang (5.3) unter anderem Durchfallquoten, Durchschnittliche Prüfungsnoten und Auswertungen der Zulassungsstatistiken[4] der Hochschule. Abschließend findet sich ein Fazit der hier gewonnenen Ergebnisse (5.5).

Der Schluss beinhaltet **Fazit** (Kapitel 6.1) und **Ausblick** (Kapitel 6.2). Hier werden zum einen die Ergebnisse der Arbeit zusammengefasst und ein Fazit gezogen und zum anderen weitergehende Themen angeschnitten, die der Autor als untersuchenswert betrachtet.

[4] hierzu siehe auch die Beschreibung der Statistiken in 3.2.2.3, 3.2.2.4 und 3.2.2.5

2. Forschungsstand

Im folgenden Kapitel wird auf den Begriff des Educational Datamining eingegangen,, eine mögliche Taxonomie desselben vorgestellt und diese Arbeit in sie eingeordnet. Abschließend wird auf die grundlegenden statistischen Verfahren eingegangen, die hier Anwendung finden.

2.1. Educational Datamining

Die Literatur beschreibt Educational Datamining (im Folgenden EDM) als:

> „Educational data mining (EDM) is a field that exploits statistical, machine-learning, and data-mining (DM) algorithms over the different types of educational data. Its main objective is to analyze these types of data in order to resolve educational research issues.“ (Romero und Ventura, 2010)

Also die Anwendung von statistischen Methoden und Verfahren des Datamining auf die speziellen Daten, die im Bereich der schulischen/wissenschaftlichen Aus- und Weiterbildung entstehen. Der Begriff des Datamining wird im EDM also inklusiv verwendet und beinhaltet auch statistische Methoden, die häufig nicht als Teil des Datamining verstanden werden.

Eine mögliche und in dieser Arbeit verwendete Taxonomie des EDM wird ebenfalls in (Romero und Ventura, 2010) gezeigt und bedient sich zur Einordnung in die einzelnen Kategorien der Zielsetzung des konkreten EDM-Vorganges und der genutzten Daten.

1	**Analysis & Visualization**
2	Providing Feedback
3	Recommendation
4	**Predicting Performance**
5	Student Modeling
6	Detecting Behavior
7	Grouping Students
8	Social Network Analysis
9	Developing Concept Map
10	Planning & Scheduling
11	Constructing Courseware

Tabelle 2.1.: Kategorien des EDM; hervorgehoben die Kategorie, in die diese Arbeit sich einordnet

In dieser Taxonomie ordnet sich diese Arbeit primär in den Bereich *Predicting Performance* ein. Dieser wird in (Romero und Ventura, 2010) beschrieben als ein EDM-Prozess mit dem Ziel, eine unbekannte Variable über den Studenten zu prognostizieren, üblicherweise Lernerfolg, Note oder Leistung. Die prognostizierte Variable kann entweder numerisch/kontinuierlich (Regression) oder kategorisch/diskret (Klassifikation) sein. In dieser Arbeit wird primär eine Klassifikation von Studenten unternommen. Wobei der Übergang zur Kategorie *Detecting Behavior* fließend ist. Teil dieser Kategorie ist etwa die Detektion/Prognose von ungewünschtem Verhalten/akademischem Scheitern (Romero und Ventura, 2010) eine Prognose die auch in dieser Arbeit versucht wird.

2.2. Statistische Verfahren

Im Laufe dieser Arbeit wurden diverse statistische Verfahren angewandt, von denen die wesentlichen hier beschrieben werden. In erster Näherung können die verwendeten Verfahren in die zwei Gruppen Klassifikation und Regression eingeordnet werden. Klassifikation ist die Zuordnung eines Objektes zu einer Klasse, basierend auf einer Menge von bereits klassifizierten Datensätzen, aus denen Klassifikationsregeln[1] abgeleitet werden. Regression basiert ebenso auf einer Menge bereits bekannter(gelabelter) Datenpunkte, prognostiziert aber keine Klassenzuordnung, sondern einen numerischen Wert (vgl. Elkan, 2010).

[1] Der Begriff Regel wird an dieser Stelle weitgefasst verwendet und inkludiert auch z.B. Wahrscheinlichkeiten aus einem Bayes-Theorem-basierten Modell.

Die im Folgenden beschriebenen Verfahren werden mit Ausnahme der linearen Regression alle als Klassifikationsverfahren bezeichnet und genutzt (vgl. Elkan, 2010). Es handelt sich um Algorithmen zum überwachten Lernen.

An dieser Stelle wird nur ein kurzer Überblick über die angewendeten Verfahren gegeben. Für eine ausführliche Besprechung wird auf die Standardliteratur zum Thema verwiesen. Einen guten Überblick gibt (Wu u. a., 2008), für einen Einblick in die Implementierungsdetails in RapidMiner bieten sich (Elkan, 2010) und (Land und Fischer, 2012) an. Detaillierte Abhandlungen zu den genutzten Algorithmen finden sich unter anderem in (Runkler, 2015), (Gabriel u. a., 2009) sowie (Cleve und Lämmel, 2014).

2.2.1. k-nächste Nachbarn

Der k-nearest neighbors[2] (kNN) Algorithmus ist ein intuitiver Klassifikationsalgorithmus, der sich umgangssprachlich beschreiben lässt als: *Ein Punkt hat die gleiche Klasse wie die meisten der k ihn umgebenden Punkte*.

Formalisiert und vereinfacht beschrieben für $k = 1$ beschrieben in (Runkler, 2015, S. 100) als:

$$||x - x_k|| = min_{j=1,\dots,n}||x - x_j|| \tag{2.1}$$

Mit dem Attributsvektor des zu klassifizierenden Punktes x dem diesem am nächsten liegenden Attributsvektor x_k, j allen bekannten Vektoren und $||$ als einem geeignetem Abstandsmaß, beispielsweise der Euklidischen[3] oder der Mahalanobis Distanz (vgl. Runkler, 2015, S. 100 f.), (vgl. Wu u. a., 2008).

Primäre Schwachstelle des Algorithmus ist die starke Abhängigkeit von einem korrekt gewählten k, wird k zu klein gewählt, kann das Ergebnis stark vom Rauschen in den Daten beeinflusst werden. Wird k zu groß gewählt, verliert der Algorithmus seine Präzision durch die inkludierten Beispiele aus falschen Klassen. Neben dieser Abhängigkeit besteht ein starker Einfluss des gewählten Abstandsmaßes, das zur Anwendung passen muss. Im Detail muss ein geringer Abstand in der Distanzmetrik eine höhere Wahrscheinlichkeit implizieren, dass die Objekte zur gleichen Klasse gehören. Beispielsweise kann für Dokumente eine Kosinus-Ähnlichkeit genauere Ergebnisse liefern als die Euklidische Distanz (vgl. Wu u. a., 2008).

[2] Deutsch: Nächste-Nachbarn-Klassifikation

[3] häufig auch wie im Englischen als „Manhattan“-Distanz bezeichnet

2.2.2. Lineare Regression

Im Folgenden wird zuerst die Definition der linearen Regression angesprochen und grafisch verdeutlicht, um dann folgend Möglichkeiten zur Analyse der Korrektheit einer linearen Regression zu zeigen.

2.2.2.1. Definition und Erläuterungen

Ein simples und intuitives Modell für die Prognose einer abhängigen Variablen y aus einer oder mehreren unabhängigen Variablen x ist anzunehmen, dass y linear von x abhängt. Aus einer bekannten Menge von Trainingsdatensätzen, die jeweils x und das dazugehörige y beinhalten kann, können dann die Parameter der Gradengleichung 2.3, die diesen linearen Zusammenhang beschreibt, ermittelt werden. Eine Erweiterung auf multiple lineare Regression ergibt sich durch die Erweiterung der Gradengleichung wie in 2.3 gezeigt auf mehrere unabhängige Variablen (hier x_1 und x_2). In den gezeigten Gleichungen ist jeweils ϵ der unbekannte und zufällige Fehler für jede Beobachtung. Eine wesentliche Voraussetzung für die Anwendbarkeit linearer Regression ist, dass der Fehler ϵ über alle Beobachtungen im Mittel an 0 angenähert ist.

$$y = a \cdot x + b + \epsilon \tag{2.2}$$

$$y = a \cdot x_1 + c \cdot x_2 + b + \epsilon \tag{2.3}$$

Eine Weiterentwicklung linearer Regression ist polynomiale Regression. Um nicht linear abhängige Variablen berechnen zu können, werden hier Funktionen höherer Ordnung genutzt (siehe das in 2.4 gezeigte Beispiel). Im Rahmen dieser Arbeit wurde polynomiale Regression nicht genutzt.

$$y = a \cdot x^2 + b + \epsilon \tag{2.4}$$

Abbildung 2.1[4] zeigt beispielhaft das Ziel der linearen Regression: eine Gerade (in der Abbildung rot) zu finden, die durch die Datenpunkte im Plot (blau) führt. Aufgrund des in der Praxis vorhandenen Messfehlers ϵ ist dieses Ziel in der Regel nicht erreichbar. Es wird daher eine Funktion gesucht, die den Fehler (grafisch die Abstände) zwischen den prognostizierten Werten $f(x)$ und den gemessenen Werten y minimiert.

[4] Quelle Wikipedia, Public Domain

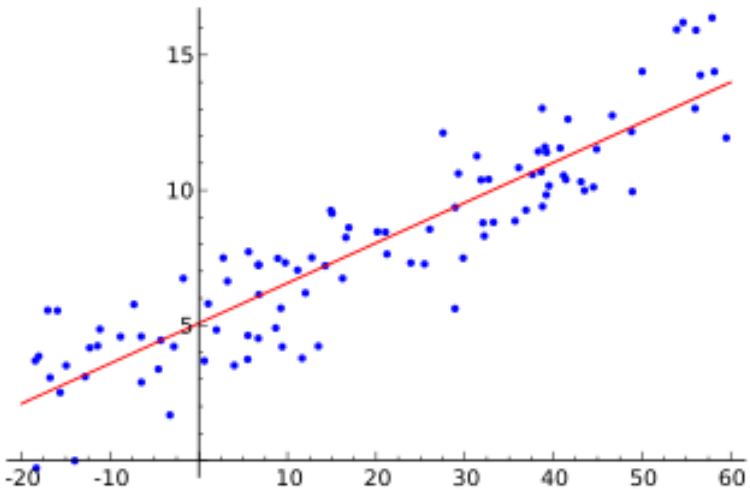


Abbildung 2.1.: Beispiel; lineare Regression

2.2.2.2. Diagnostik der linearen Regression

Lineare Regression (siehe Gleichung 2.2) erfordert, dass der Zusammenhang zwischen abhängigen und unabhängigen Variablen durch eine lineare Funktion ausgedrückt werden kann und der Messfehler ϵ im Median 0 ist.

Abbildung 2.2 zeigt einen Datensatz, der durch lineare Regression nur schlecht zu approximieren ist. Diese Schwierigkeit im gezeigten Beispiel beruht darauf, dass zwei unterschiedliche Distributionen gewählt wurden, um die Grafik zu erstellen.

Abbildung 2.2.: Beispiel; Probleme der linearen Regression

Die korrekte Verteilung des Fehlers ϵ kann unter anderem grafisch durch das Auftragen des beobachteten Fehlers zwischen Vorhersage und Beobachtung im Trainings- bzw. Evaluierungsdatensatz (ϵ = *Residuals*) analysiert werden. Abbildung 2.3 zeigt einen solchen Plot.

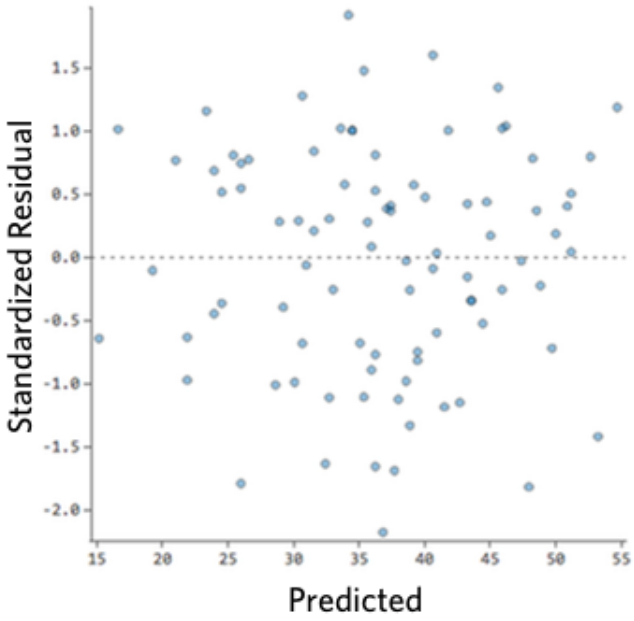


Abbildung 2.3.: Beispiel; Residuals Plot

Auf der x-Achse ist der prognostizierte Wert aufgetragen und diesem gegenüber auf der y-Achse der gemessene Fehler. Wenn die genutzte lineare Regression zur Approximation geeignet ist, sind die Fehler wie in der gezeigten Abbildung zufällig um die 0-Linie der y-Achse verteilt und weisen keine wahrnehmbare Struktur auf.

2.2.3. Logistic Regression

Logistic Regression ist die Anwendung von Regressionsanalyse auf das Problem der binominalen Klassifizierung[5]. Eine lineare Regression ist hier in der Regel nicht durchführbar, da diese eine Normalverteilung der Fehler (siehe 2.2.2.2) erfordert und die Transformation der Prognose in die binominale Verteilung (0/1) nicht verlustfrei möglich ist (Elkan, 2010).

Diese Probleme werden in der logistischen Regression durch die Transformation der abhängigen Variablen zu einem *Logit* gelöst. Hierdurch kann eine Sigmoid-Funktion auf dem ursprünglichen Datensatz dargestellt werden. Die S-Form dieser Funktion ist häufig eine bessere Approximation der binominalen Zielverteilung als eine Gerade und der *Logit* ist eine linear von x abhängige Funktion (Elkan, 2010).

Im Folgenden wird der Logit erklärt und sein Zusammenhang mit der Wahrscheinlichkeit, dass $y = 1$ ist, gezeigt.

[5]Zuweisung eines Datenpunktes zu einer von zwei Klassen

Angenommen, folgende Punkte gelten:

- y ist ein binäres Ereignis (0/1)
- p ist die Wahrscheinlichkeit, dass y eintritt ($y = 1$)
- Es ergibt sich, dass $(1 - p)$ die Wahrscheinlichkeit ist, dass y nicht eintritt ($y = 0$)
- Dann ist $p/(1 - p)$ die Chance, dass y eintritt.

Der *Logit* ist der Logarithmus der Chance also $\log(p/(1-p))$ und linear abhängig von x. Er kann ähnlich einer linearen Regression ausgedrückt werden als: Durch die Transformation ergeben sich die folgenden Gleichungen

$$logit = \log(p/(1 - p)) = b_0 x + b_1 \tag{2.5}$$

oder verallgemeinert für multiple unabhängige Variablen:

$$logit = b_0 x + b_1 x_1 + ... + b_b x_n \tag{2.6}$$

Nach der Berechnung des Logits kann durch Einsetzen desselben in die Gleichung 2.7 die Wahrscheinlichkeit, dass die abhängige Variable $y = 1$ ist, berechnet werden.

$$p = e^{logit}/(1 + e^{logit}) \tag{2.7}$$

Die Werte von $b_1, ..., b_n$ in der Gleichung 2.5 bzw. 2.6 werden in der Praxis durch einen iterativen Ansatz bestimmt.

Eine historische Betrachtung der logistischen Regression kann in (Cramer, 2002) gefunden werden. Die Gleichungen 2.5, 2.6 und 2.7 sowie der Ansatz der Erläuterung wurden aus (Elkan, 2010) übernommen.

2.2.4. Entscheidungsbäume (Decision Tree)

Der hier beschriebene Algorithmus für Entscheidungsbäume ist C4.5, ein Nachfolger von CLS (Hunt u. a., 1966) und ID3 (Quinlan, 1979). Decision Trees[6] erstellen Klassifikatoren aus einer mit Klassen annotierten Objektmenge S, in der jedem Objekt S_i ein Attributsvektor v zugewiesen ist, in dem der folgende Algorithmus iterativ durchlaufen wird:

1. Wenn alle Objekte in S zur gleichen Klasse gehören oder S klein ist, ist der Baum ein Blatt, das mit der Klasse, der die meisten Objekte aus S angehören, gelabelt ist.

[6] Entscheidungsbäume

2. Sonst wird ein Test auf einem einzelnen Attribut mit zwei oder mehr möglichen Ergebnissen gewählt und dieser Test als Wurzel des Baumes definiert. Die Zweige des Baumes sind Partitionen von $S_1, S_2, \ldots$ von S mit je einem Zweig für jedes mögliche Ergebnis des gewählten Tests.

3. Der Algorithmus wird rekursiv auf die einzelnen Partitionen $S_1, S_2, \ldots$ angewendet.

Zwei häufig genutzte Heuristiken zur Auswahl des geeigneten Tests in Schritt 2 sind Informationsgewinn[7] zur Minimierung der totalen Entropie[8] aller Partitionen $S_1, S_2, \ldots$ und die Rate des Informationsgewinns, die anders als der reine Informationsgewinn keine Tendenz zu Merkmalen mit möglichst vielen Partitionen aufweist (vgl. Wu u. a., 2008).

Vorteile von Entscheidungsbäumen sind eine intuitive Erfassbarkeit der Ergebnisse für Menschen und dass nicht zwingend alle Attribute genutzt werden müssen. Nachteilig können optimal oft nur diskrete Attribute verarbeitet werden. Bei der Verwendung kontinuierlicher Attribute liegen die Klassengrenzen immer partitionsweise parallel zu den Achsen des Koordinatensystems, was die Erfassung kompliziert nichtlinear verlaufender Klassen nicht effizient abbildet (vgl. Runkler, 2015, S. 102-106) (vgl. Wu u. a., 2008).

Abbildung 2.4 zeigt beispielhaft einen Entscheidungsbaum, der im Rahmen dieser Arbeit zur Studienerfolgsprognose erstellt wurde.

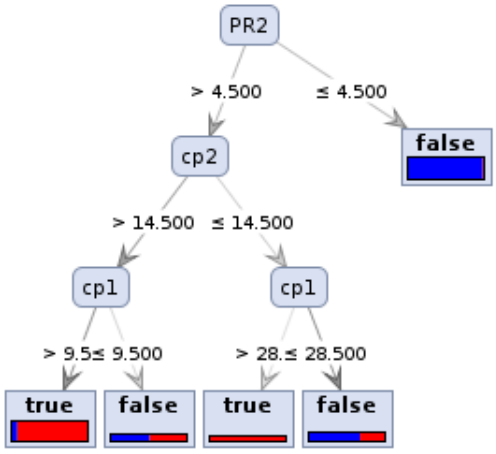


Abbildung 2.4.: Beispiel; Entscheidungsbaum

[7] Synonym für: Kullback-Leibler-Divergenz
[8] ein Maß der Unordnung

Beginnend mit der Note für *PR2* wird, falls die Note $> 4,5$ ist, weiter evaluiert, sonst wird der Student der Gruppe *false* (kein Abschluss) zugewiesen. Wenn die Note $\leq 4,5$ ist, wird auf *cp2* (der erreichten Menge von CPs im zweiten Semester) geprüft, ob mehr oder weniger als $14,5$ CPs erreicht wurden und abschließend auf *cp1* (der erreichten Menge von CPs im ersten Semester). Die Farbe des Balkens in den Blättern zeigt die Verteilung der Klassen true=blau=Abschluss und false=rot=kein Abschluss. Die Höhe der einzelnen Balken ist analog zur Kardinalität der enthaltenen Menge von Datensätzen.

2.2.5. Neuronales Netz (Auto MLP)

Künstliche neuronale Netze[9] bauen auf dem Konzept einer biologischen Nervenzelle[10] auf. Dieses Konzept von vielen einfachen Nervenzellen, die miteinander verbunden ein komplexes Verhalten ergeben, wird in Abbildung 2.5 grafisch gezeigt (Elkan, 2010).

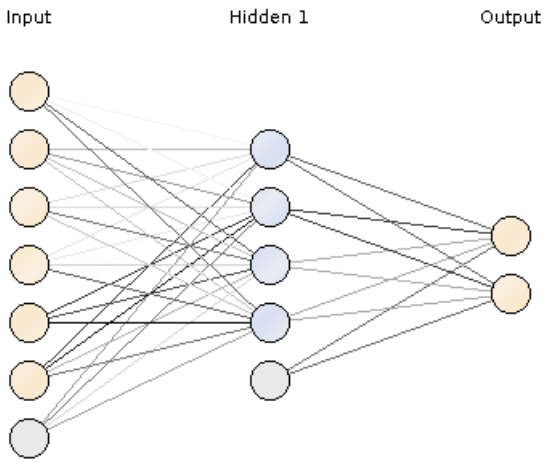


Abbildung 2.5.: Beispiel; Neuronales Netz

Neben dem *Input-* und *Output-Layer*, die nur jeweils Aus-/Eingänge haben, enthalten neuronale Netze eine variable Anzahl von *hidden* Layern[11], in dem die einzelnen Neuronen Ein- und Ausgänge haben. Durch diese Form werden Gleichungen, die einen nichtlinearen, komplexen Zusammenhang zwischen Ein- und Ausgängen des gesamten Netzes aufweisen, abgebildet (Elkan, 2010). Abbildung 2.6 zeigt ein simples neuronales Netzwerk:

[9] im Folgenden: Neuronale Netze
[10] engl.: neuron
[11] Im hier gezeigten Beispiel nur 1 Hidden Layer

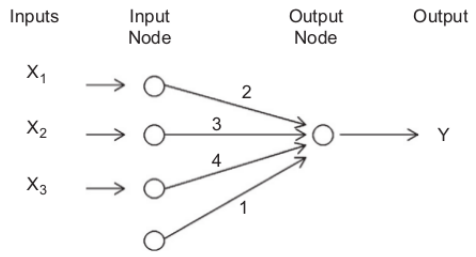


Abbildung 2.6.: Beispiel; einfaches neuronales Netz (Elkan, 2010)

In dem gezeigten simplen neuronalen Netzwerk ohne *hidden* Layer wird die untenstehende Gleichung 2.8 (entnommen aus (Elkan, 2010)) abgebildet. Wobei die Eingänge X_1, X_2 und X_3 und die Konstante (in der Abbildung ganz unten) das *Input-Layer* bilden und direkt auf die *Output-Node* und damit den Ausgang Y abbilden. An den Kanten des neuronalen Netzes sind die konstanten Faktoren der Gleichung aufgetragen.

$$Y = 1 + 2X_1 + 3X_2 + 4X_3 \tag{2.8}$$

In dieser Arbeit wird AutoMLP[12] genutzt, um die verschiedenen Faktoren des neuronalen Netzes, wie die Anzahl von *hidden* Layers und die Lernrate, in einer Mischung aus stochastischem und evolutionärem Prozess automatisch zu ermitteln (Thomas Breuel, 2010).

2.2.6. Naives Bayes

Eine Klassifikation durch einen naiven Bayes-Klassifikator ist probabilistisch und baut auf Bayes Theorem auf, es gilt für zwei Zufallsereignisse A und B, entnommen aus (vgl. Runkler, 2015):

$$p(A|B) \cdot p(B) = p(B|A) \cdot p(A) \tag{2.9}$$

wobei $p(A)$ die (A-priori)-Wahrscheinlichkeit ist, dass ein Ereignis x eintritt und $p(A|B)$ die bedingte Wahrscheinlichkeit, dass A unter der Bedingung eintritt, dass B bereits eingetreten ist.

Durch Umformung ergibt sich die Klassifikationswahrscheinlichkeit für die Klasse i[13] durch die Gleichung 2.10 (entnommen aus (vgl. Runkler, 2015)) unter der Annahme, dass die einzelnen Merkmale des Merkmalsvektors x unabhängig voneinander sind.

[12] Multi Layer Perceptrons
[13] j ist die jeweils andere Klasse

$$p(i|x) = \frac{p(i) \cdot \prod_{k=i}^{p} p(x^{(k)}|i)}{\sum_{j=1}^{c} p(j) \cdot \prod_{k=1}^{p} p(x^{(k)}|j)} \tag{2.10}$$

Durch Einsetzen von „Objekt ist Teil der Klasse/ist kein Teil der Klasse“ und „Objekt hat Merkmalsvektor x“ können aus einem gegebenen klassifizierten Datensatz die Wahrscheinlichkeiten für die Stichprobe bestimmt werden. Mit diesen bekannten Wahrscheinlichkeiten können dann nicht klassifizierte Datensätze klassifiziert werden (vgl. Runkler, 2015) (vgl. Wu u. a., 2008).

2.2.7. SVM

Support Vektor Maschinen (vgl. Conway und White, 2012, S. 275-284) sind derzeit eine der am häufigsten getesteten Methoden zur Klassifizierung. SVM finden auf linear separierbaren Daten eine optimale Lösung. Das Optimum wird hier definiert als die Funktion, die die vorhandenen Lerndaten mit maximalem Abstand trennt. Die optimale Funktion $F(x)$ hat also einen maximalen Abstand zu den bekannten Daten $X = x_1, \ldots, x_n$. Hierdurch wird häufig eine generalisierbare Lösung gefunden (vgl. Wu u. a., 2008).

Für nicht linear separierbare Daten wird eine Kernel-Funktion verwendet, um die zu klassifizierenden Daten $X = x_1, \ldots, x_n \in \mathbb{R}^p$ auf einen höherdimensionalen Datensatz $X' = x'_1, \ldots, x'_n \in \mathbb{R}^q$ mit $q > p$ abzubilden und in der besser geeigneten Dimension q eine näherungsweise lineare Grenze zu finden. Abbildung 2.7 zeigt diese Strategie an einem Beispielbild (vgl. Runkler, 2015, S. 98 ff.), (vgl. Wu u. a., 2008).

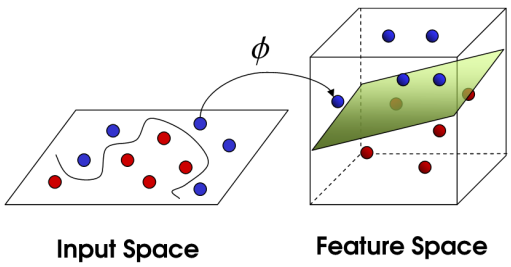


Abbildung 2.7.: Beispiel; SVM (Stackoverflow)

Es ist zu sehen, wie durch das Verschieben in eine höhere Dimension eine korrekte, in der Ursprungsdarstellung nichtlineare Lösung gefunden werden kann.

Durch Erweiterungen können Daten erkannt werden, die auch in einer höheren Dimension nicht komplett linear trennbar sind, oder SVM zur Regressionsanalyse verwendet werden

(vgl. Wu u. a., 2008). Probleme mit SVMs treten häufig bei großen Datenmengen auf, da naive Berechnungsmethoden für SVMs nicht effizient sind. Hier existieren iterative Lösungen, die in endlicher Zeit lösbar sind (vgl. Tsang u. a., 2005). Weiterhin sind die Ergbnisse, die eine SVM liefert, häufig nicht intuitiv nachvollziehbar, da im Gegensatz zu z.B. einem Entscheidungsbaum häufig keine grafische Darstellung existiert (vgl. Runkler, 2015).

2.3. Statistik-IDEs und Sprachen

Im Folgenden wird auf die in dieser Arbeit verwendeten statistischen Sprachen (R) und IDEs (RapidMiner, RStudio) eingegangen. Es wird jeweils der Hintergrund zum verwendeten Werkzeug und ein Anwendungsbeispiel gezeigt.

2.3.1. RapidMiner

RapidMiner ist eine Entwicklungsumgebung für Datamining- und Analyseprozesse. In RapidMiner werden diese in einer grafischen Benutzeroberfläche in Form von Prozessgraphen(*Flowdesign*) erzeugt. Intern wird der Prozessaufbau als XML erzeugt und persistiert. Auf diesen Prozessgraphen können während der Ausführung klassische Mittel der Programmentwicklung genutzt werden. So können Teile des Graphen zusammengefasst und als Einheit (*Building Block*) weiterverwendet werden. Auf dem Graphen können Haltepunkte (*Breakpoints*) gesetzt werden, um während der Prozessdurchführung durch Inspektion von Teilergebnissen zur Defektlokalisierung beizutragen (Elkan, 2010), (Land und Fischer, 2012), (Rapid-i, 2010).

RapidMiner ist aus einem Projekt der Universität Dortmund, dort als *YALE* (Yet Another Learning Environment), hervorgegangen und wird mittlerweile in Version 7 als kommerzielles Produkt von der durch ehemalige Studenten der Universität gegründeten Firma *RapidMiner* vertrieben und entwickelt. Trotz dieser Kommerzialisierung ist die jeweils vorherige Version von RapidMiner[14] unter einer Open-Source-Lizenz verfügbar (RapidMiner), (Elkan, 2010).

2.3.1.1. RapidMiner Beispiel

Abbildung 2.8[15] zeigt beispielhaft einen Prozessgraphen, der aus dieser Arbeit stammt und mit RapidMiner entwickelt wurde.

[14] derzeit also 6.x

[15] Abbildung ist in Anhang A.3.1 in Vergrößerung gezeigt

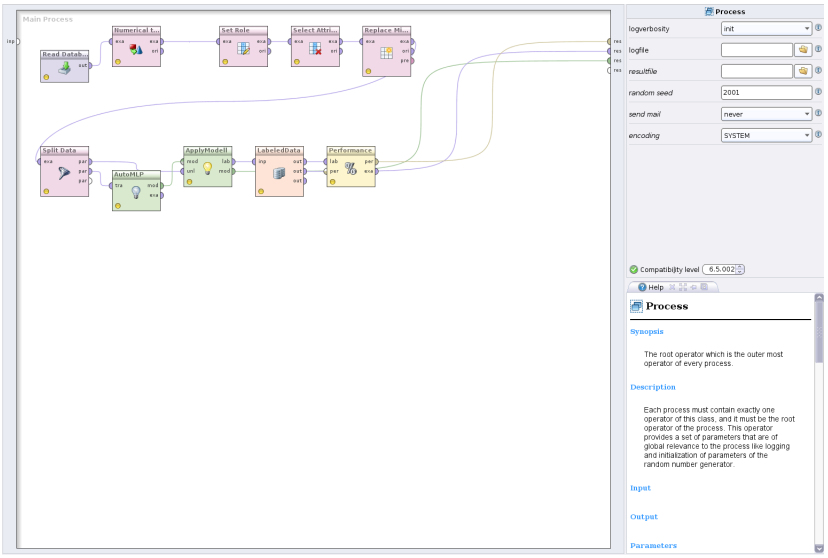


Abbildung 2.8.: Beispiel; RapidMiner Flowdesign

Zu sehen ist ein Prozessgraph, in dem eine Klassifikation der Studierenden mithilfe eines durch AutoMLP (Thomas Breuel, 2010) trainierten neuronalen Netzes durchgeführt wird. Gut zu erkennen ist an dieser Stelle, wie sich die grafische Entwicklung in RapidMiner von der konsolenbasierten in R unterscheidet, siehe auch 2.3.2. Hierbei befindet sich in der Abbildung links die grafische Darstellung des Prozessgraphen und rechts oben ein Konfigurationsfeld für den Prozess bzw. bei entsprechender Auswahl einen Operator. Rechts unten ist ein Fenster mit der Onlinehilfe des ausgewählten Operators zu sehen. Nicht abgebildet ist links neben der Prozessdarstellung die Operator- und Prozess-/Datenwahl. Im Detail sind die Elemente der grafischen Benutzeroberfläche in der Herstellerdokumentation (Rapid-i, 2010) beschrieben.

Im Detail ist[16] zu erkennen, dass zuerst Daten aus der Datenbank geladen werden (*Read Database*) und dann ein Attribut vom Typ *Numerical* in den Typ *Binominal* transformiert wird (*Numerical to Binominal*), das so transformierte Attribut ist das Ziel der Klassifizierung, das Label: *isGraduated*. Danach werden für die einzelnen Attribute Rollen vergeben (*Set Role*), es werden also RapidMiner-Metadaten über die genutzten Daten gegeben, nämlich welches Attribut das Klassifikationsziel (*isGraduated*) ist und welches Attribut eine ID(*id*) ist. Dann werden die in der Klassifikation genutzten Attribute ausgewählt (*Select Attribute*) und fehlende Werte durch eine 0 ersetzt (*Replace Missing Values*). Die Daten werden nachfolgend in zwei gleichgroße Partitionen geteilt (*Split Data*). Eine Partition wird zur Modellbildung genutzt (*AutoMLP*), die andere, um das erstellte Modell durch Anwendung zu testen (*Apply*

[16] Abbildung ist in Anhang A.3.1 in Vergrößerung gezeigt

Model). Das Ergebnis dieser Operationen ist ein mit Labeln versehener Datensatz (*Labeled Data*), auf dem durch einen Statistik-Operator eine Erfolgsmessung durchgeführt wird (*Performance*).

2.3.1.2. Funktionalität

RapidMiner unterstützt unter anderem alle in 2.2 beschriebenen Verfahren. Neben diesen der Klassifikation zuzuordnenden Methoden werden diverse weitere aus den Gebieten Regression, Clustering, Assoziation, Anomalieerkennung und Textmining unterstützt. Neben diesen bereits implementierten Verfahren können weitere durch die Anwender erstellt und mithilfe der in RapidMiner integrierten Funktionen mit anderen geteilt werden. Außer diesen werden Funktionalitäten im Bereich Social Media (z.B. Twitter-Anbindung) angeboten (Land und Fischer, 2012). Im Firmenverbund sind Anwendungen zur Portierung der in RapidMiner erstellten Prozesse auf dedizierte Server (RapidMiner-Server) und BigData-Plattformen (Hadoop) vorhanden. In dieser Arbeit wurden primär die in 2.2 beschriebenen Methoden und Verfahren angewendet.

2.3.2. R

R (The R Foundation) ist eine Programmiersprache zur Beschreibung statistischer Anwendungen. R ohne Erweiterungen wird interaktiv über eine Kommandozeile bedient bzw. führt Skripte aus. In dieser Arbeit wurde als grafische Entwicklungsumgebung für R RStudio (RStudio) verwendet.

2.3.2.1. R Beispiel

Listing 2.1 zeigt beispielhaft ein in R implementiertes Programm zur Anzeige der CPS nach 3 Semestern pro Kohorte.

Modell Fit in R

```
1 #library load
2 library(RMySQL)
3 library(caret)
4
5 #db connection
6 con <- dbConnect(MySQL(),
7     user = 'thesis',
8     password = 'xxx',
```

```
9     host = 'localhost',
10    dbname='student_data')
11
12 #load data
13 kohorte3Semester <- dbGetQuery(conn = con, statement = "
      select kohorte as k, sum(cp)/(select count(*) from
      students where kohorte = k) as avg, (select bewerber_
      erfolgs_quote from C2 where kohorte = k) as quote from
      big  where hochschulsemester <= 3 and kohorte > '2006WS'
      and kohorte < '2012SS' and bestanden = 1 group by kohorte
      ;")
14
15 #configure panel
16 panel=function(x, y,labels) {
17      panel.xyplot(x, y,);
18      ltext(x=x, y=y, labels=kohorte3Semester$k,pos=1, offset
      =1, cex=0.8); panel.lmline(x,y)};
19
20 #plot
21 xyplot(avg ~ quote, kohorte3Semester, grid = TRUE, type = c(
      "p", "r"), col.line = "orange", lwd = 1, xlab = '
      Bewerbungserfolgsquote', ylab = 'CPs', main="CP nach 3
      Semester zu Bewerbungserfolgsquote", panel=panel)
```

In Zeile 2 und 3 werden Bibliotheken zur Datenbankanbindung (*RMysql*) und Anzeigekonfiguration (*calibrate*) geladen. Zwischen den Zeilen 6 bis 10 wird die Verbindung mit der Datenbank aufgebaut und folgend in Zeile 13 verwendet, um Daten abzurufen. Die Konfiguration der Anzeigefläche findet in Zeile 16 bis 18 statt und wird abschließend in Zeile 21 genutzt, um die abgerufenen Daten anzuzeigen. In Zeile 21 ist die Erstellung des Regressionsmodells durch $avg \sim quote$ innerhalb der Anzeigefunktion gezeigt. Das Modell ist hier mit der abhängigen Variablen *avg* und unabhängigen Variablen *quote* gebildet.

2.3.2.2. RStudio

Abbildung 2.9[17] greift das in 2.3.2.1 gezeigte Beispiel auf und bettet es in RStudio ein.

[17] Abbildung ist in Anhang A.3.2 in Vergrößerung gezeigt

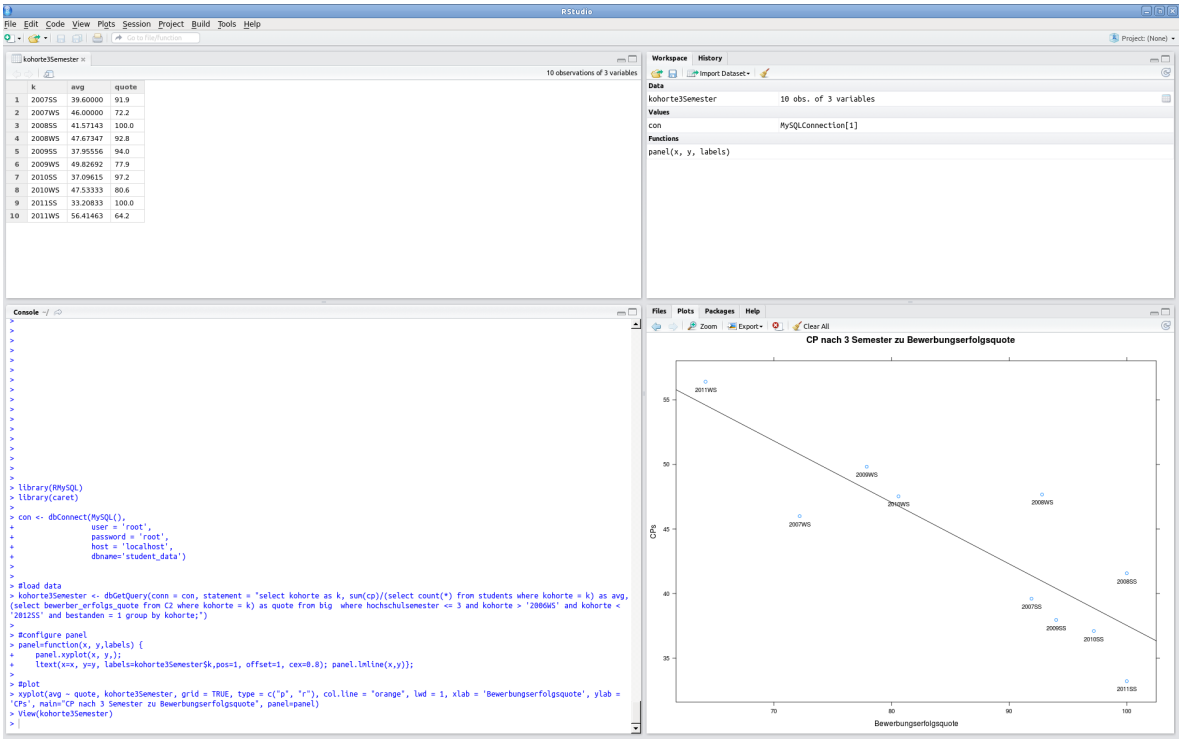


Abbildung 2.9.: Beispiel; RStudio

Gezeigt ist die RStudio-Entwicklungsumgebung mit ihrer klassischen Vierteilung in unterschiedliche Arbeitsbereiche, wobei hier die Ansichten *Source* (links oben), *Workspace* (rechts oben), *Console* (links unten) und *Plot* (rechts unten) gezeigt sind. In der *Source-Ansicht* werden Details des in der *Workspace-Ansicht* ausgewählten Objektes gezeigt, im Beispiel handelt es sich um das *Data Frame kohorte3Semester*, das die drei Spalten Kohorte(*k*), Durchschnittliche CPs (*avg*) und Bewerbungserfolgsquote (*quote*) beinhaltet. In der Ansicht *Console* ist der bereits in Listing 2.1 gezeigte R-Quellcode und im *Plot* der durch diesen Quellcode erzeugte Datenplot mit der dazugehörigen Regressionsgeraden.

Teil II.

Hauptteil

3. Analyse

In der Analyse wird einleitend auf den Prozess des Knowledge Discovery in Databases eingegangen, der die Arbeit des Dataminings strukturiert (Kapitel 3.1). Es werden dann die Rahmenbedingungen dieser Arbeit betrachtet (Kapitel 3.2), neben einer ausführlichen Betrachtung zur vorhandenen Datenbasis (Kapitel 3.2.2.1, 3.2.2.3, 3.2.2.4 und 3.2.2.5) sind hier die Faktoren, die die Stetigkeit der Daten beeinflussen, geführt (Kapitel 3.2.3) sowie verschiedene Perspektiven des Studienerfolgs (Kapitel 3.2.1) dargestellt. Im Anschluss wird auf die unterschiedlichen am Studienerfolg interessierten Stakeholder (Kapitel 3.2.4) eingegangen. Es folgen außerhalb der Rahmenbedingungen vergleichende Betrachtungen (Kapitel 3.3) aus der Literatur, ein Anschnitt auf die ethischen Fragen, die mit der statistischen Verarbeitung personenbezogener Daten einhergehen (Kapitel 3.4) und abschließend ein Fazit, das die in dieser Arbeit primär betrachteten Thesen darstellt (Kapitel 3.5).

3.1. Knowledge Discovery in Databases (KDD)

Datamining und Wissensfindung in Datenbanken (Knowledge Discovery in Databases)[1] ist ein Modell des Dataminings, das Schritte der vorbereitenden Untersuchungen und der Datentransformationen inkludiert. Wobei beide Begriffe (Datamining und KDD) häufig auch synonym verwendet werden. Ziel des KDD ist es, *neues*, *nützliches* und *interessantes* Wissen zu erzeugen. Im Folgenden werden die drei genannten Zielattribute[2] und die einzelnen Schritte des KDD erläutert.

3.1.1. Zieldefinition

Die Literatur fordert für das Ergebnis des KDD, dass das gefundene Wissen die Attribute *neues*, *nützliches* und *interessantes* erfüllt. Diese werden im Folgenden kurz erläutert.

[1] im Folgenden: KDD

[2] *neues*, *nützliches* und *interessantes*

> „Neues Wissen: Hier sind meist implizite, bisher unbekannte Erkenntnisse gemeint. Data Mining Verfahren erzeugen in der Regel sehr viele Informationen. Der Anspruch, das Wissen solle wirklich neu sein, erfordert zum Beispiel den Abgleich gefundener Informationen mit schon gespeichertem Wissen."

(Beierle und Kern-Isberner, 2008, S. 144)

> „Nützliches und interessantes Wissen: Das entdeckte Wissen soll im behandelten Kontext relevant sein. Im ökonomischen Bereich wird man daher den KDD-Prozess meist durch betriebswirtschaftliche Parameter wie Umsatz, Gewinn usw. steuern. Bei wissenschaftlichem KDD kommen eher qualitative Gütekriterien wie Spezifität und Generalität zum Einsatz."

(Beierle und Kern-Isberner, 2008, S. 144)

> „In verständlicher Form: Die neue, nützliche und interessante Information muss dem Benutzer auch als solche präsentiert werden. D.h., der Benutzer soll auf Anhieb den Wert des KDD-Ergebnisses erkennen können. Dies verlangt die Aufbereitung gefundener Informationen in lesbarer und anschaulicher Weise."

(Beierle und Kern-Isberner, 2008, S. 144)

Diese Forderungen für das Ergebnis des KDD sind nicht isoliert, sondern im Verbund zu betrachten und definieren so Rahmenbedingungen für das Ergebnis. Neben dieser Funktion beeinflussen sie auch wesentlich den Prozess des KDD, der dem Anwender[3] im Optimum die Möglichkeit gibt, interaktiv mit diesem umzugehen.

3.1.2. Schritte des KDD

Der Prozess des KDD besteht aus 8 Schritten, die im Folgenden erläutert werden. Die unten stehende Tabelle zeigt diese Schritte.

1. Hintergrundwissen und Zielsetzung
2. Datenauswahl
3. Datenbereinigung
4. Datenreduktion und -projektion
5. Modellfunktionalität
6. Verfahrensauswahl

[3] häufig ein Experte der behandelten Domäne

7. Datamining
8. Interpretation

(vgl. Beierle und Kern-Isberner, 2008, S. 143 ff.)

Die Schritte des KDD sind im Einzelnen, nach (Fayyad u. a., 1996):

Hintergrundwissen und Zielsetzung

Es wird ein Verständnis der Applikationsdomäne und des relevanten bestehenden Wissens angestrebt. Aufbauen hierauf wird die Zielsetzung des KDD-Prozesses aus der Sicht des Anwenders formuliert.

Datenauswahl

Es muss der Datensatz, auf dem der KDD-Prozess arbeiten soll, erzeugt werden. Dies kann entweder ein Datensatz sein, der für diesen Prozess neu erstellt wurde, oder ein Fokus bzw. eine Untermenge eines bereits bestehenden Datensatzes.

Datenbereinigung

Die Daten müssen gesäubert und vorverarbeitet werden. Basis-Operationen in dieser Phase sind das Entfernen von Rauschen in den Daten und hierzu das Erstellen bzw. Beschaffen von Informationen, die notwendig sind, um Datenrauschen zu modellieren.. Abschließend ist das Definieren einer Strategie, um nicht vorhandene Attribute in einzelnen Datensätzen sowie Veränderungen über Zeit in den Daten zu behandeln.

Datenreduktion und -projektion

Durch Reduktion oder Transformation werden die vorhandenen Daten noch einmal komprimiert.

Modellfunktionalität

Es wird eine zum Ziel des KDD-Prozesses passende Methode des Datamining ausgewählt, beispielsweise Klassifikation, Regression oder Clustering.

Verfahrensauswahl

In einer exploratorischen Analyse werden Modelle und Hypothesen sowie Methoden und Algorithmen ausgewählt, mit denen nach Mustern in den Daten gesucht wird.

Datamining

In der Datamining-Phase werden die gewählten Methoden/Verfahren genutzt, um Muster in den Daten zu finden. Dies schließt beispielsweise Entscheidungsbäume, Regressionen und Cluster ein.

Interpretation

Abschließend werden die gefundenen Muster interpretiert. Diese Phase kann auch die Visualisierung der extrahierten Muster beinhalten, um sie leichter verständlich zu machen.

Vom Schritt der Interpretation kann zu jeder der vorhergehenden Phasen zurückgesprungen werden, um in einem iterativen Prozess eventuelle Schwächen der getroffenen Entscheidungen zu beheben. In einem nicht explizit enthaltenen neunten Schritt wird das neu gewonnene Wissen verwertet. Dies kann in Form einer direkten Reaktion auf das neue Wissen geschehen oder in der Weiterverarbeitung in anderen Informationssystemen oder auch durch die Dokumentation für interessierte weitere Parteien (Fayyad u. a., 1996).

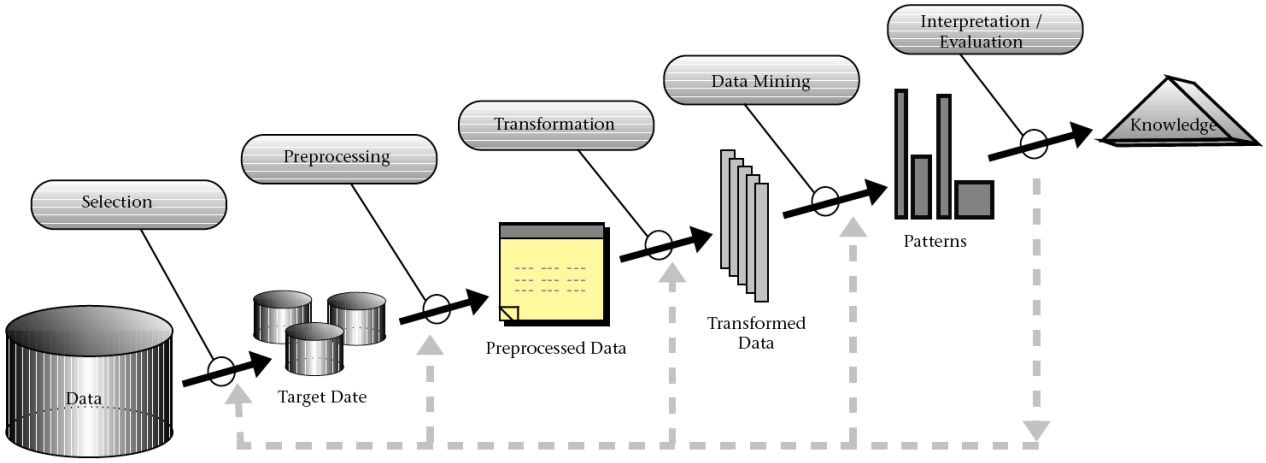


Abbildung 3.1.: KDD nach Fayyad (Fayyad u. a., 1996)

Die Abbildung 3.1 zeigt den Kern des KDD-Prozesses nach (Fayyad u. a., 1996) mit englischen Namen für die einzelnen Schritte, nicht abgebildet sind die Schritte Hintergrundwissen, Modellfunktionalität, Verfahrensauswahl. Diese sind in diesem Modell integriert in die umliegenden Schritte. Abgebildet sind also die Datenauswahl (Selection), die Datenbereinigung (Preprocessing), die Datenreduktion und Projektion (Transformation), Datamining (Data Mining) und Interpretation (Interpretation / Evaluation), die von Daten (Data) hin zum Wissen (Knowledge) führen.

3.1.3. Einordnung dieser Arbeit in den Prozess des KDD

Im Folgenden werden die Phasen und Konzepte des Datamining, die in Kapitel 3.1 aus der Literatur erarbeitet wurden, den Teilen dieser Arbeit zugeordnet. Die Abgrenzung zwischen den einzelnen Teilen des KDD-Prozesses ist hier nicht trennscharf, sondern stellt nur die Schwerpunkte der einzelnen Phasen dar.

Hintergrundwissen und Zielsetzung sind in den Gliederungspunkten 3.2.2 (Hintergrundwissen) und 1.2 (Ziele) konzentriert. Die **Datenauswahl** und Beschreibung wird im Kapitel 3.2.2.1 im Rahmen der Datenbasis beschrieben. **Datenbereinigung** ist wie in KDD-Prozessen üblich ein großer Teil der notwendigen Arbeit (Fayyad u. a., 1996) und wird in den Kapiteln 4.4 und 4.3 behandelt. Eine weitere **Datenreduktion und -projektion** findet im Rahmen dieser Arbeit nicht statt. Die **Modellfunktionalität** wird in den Grundlagen (Kapitel

2.1) gezeigt. Das Kapitel 5 kombiniert die Phasen **Verfahrensauswahl** und **Datamining**. Ansätze einer **Interpretation** finden sich ebenfalls in 5 und dem Ausblick (6.2).

3.2. Rahmenbedingungen

Im Folgenden werden die Rahmenbedingungen dieser Arbeit betrachtet. Einleitend werden verschiedene Perspektiven des Studienerfolgs (Kapitel 3.2.1) dargestellt. Es folgt eine ausführliche Betrachtung der vorhandenen Datenbasis (Kapitel 3.2.2.1, 3.2.2.3, 3.2.2.4 und 3.2.2.5). Am Ende des Kapitels werden die aus unterschiedlichen Motivationen am Studienerfolg interessierten Stakeholder des Prozesses betrachtet (Kapitel 3.2.4) und abschließend wird auf die Stetigkeit der Datenbasis eingegangen (Kapitel 3.2.3).

Neben einer ausführlichen Betrachtung zur vorhandenen Datenbasis (Kapitel 3.2.2.1, 3.2.2.3, 3.2.2.4 und 3.2.2.5) sind hier die Faktoren, die die Stetigkeit der Daten beeinflussen, aufgeführt (Kapitel 3.2.3) sowie verschiedene Perspektiven des Studienerfolgs (Kapitel 3.2.1) dargestellt.

3.2.1. Studienerfolg

Ein wesentlicher Indikator für das Funktionieren eines Studienganges ist der Studienerfolg der immatrikulierten Studenten. Studienerfolg ist in diesem Zusammenhang ein von der Perspektive abhängender Begriff, der in unterschiedlichen Gruppen unterschiedlich definiert wird. Im Folgenden wird auf einige gebräuchliche Maße des Studienerfolgs eingegangen. In dieser Arbeit wird der Studienerfolgsbegriff größtenteils synonym mit dem Abschluss verwendet, Abweichungen von diesem Begriff sind an den betreffenden Stellen dokumentiert.

Gebräuchliche Maße für den Studienerfolg sind unter anderem:

- Abschluss erreicht
- Abschlussnote
- Regelstudienzeit
- Berufsaussichten

Die einfachste, weil binäre Sicht auf den Erfolg im Studium fragt nur, ob der im Studiengang angestrebte Abschluss erreicht wurde oder nicht. Etwas komplexer ist die Frage danach, ob ein Abschluss in Regelstudienzeit erreicht wurde, und nicht, ob dies der Intention des Studierenden entsprach oder nicht. Die Abschlussnote stellt ein Ordnungskriterium für die Studenten der Klassen Abschluss/kein Abschluss dar und erlaubt damit eine Sortierung innerhalb

der groben Klassen nach dem Erfolg des einzelnen. Retrospektiv können die Berufsaussichten, die mit dem Studium verbunden sind, betrachtet werden.

Von diesen Facetten des Studienerfolgs wird hier primär die erste, das Erreichen des Abschlusses betrachtet. Dies, da das Kriterium Abschluss/kein Abschluss sich aus den vorliegenden Daten ergibt und berechenbar ist. Eine explorative Analyse der Daten hat gezeigt, dass das Kriterium der Regelstudienzeit sich nur schwer auf den betrachteten Studiengang anwenden lässt, da viele Studierende arbeiten und eher Attribute des Teilzeitstudiums aufweisen (vgl. Meisel, 2005), (vgl. Meisel, 2014). Über den späteren Berufserfolg gibt es keine mit den vorliegenden Daten verbundenen Erkenntnisse.

3.2.2. Beschreibung des Studienganges und der Daten

Soweit es dort nicht anderes explizit dokumentiert ist, stehen die in den Abschnitten 3.2.2.3, 3.2.2.4 und 3.2.2.5 beschriebenen Daten der von der Hochschule für Angewandte Wissenschaften geführten Statistiken jeweils kohortenweise ab dem Sommersemester 2007 zur Verfügung.

3.2.2.1. Datenbasis

Die Datenbasis für diese Veröffentlichung ist ein Abzug des Studentenverwaltungssystems der Informatik (Stisys). Insgesamt liegen 31787 Datensätze vom Sommersemester 2004 bis zum Wintersemester 2015 vor.

Jeder dieser Datensätze entspricht einer benoteten Prüfung oder einer unbenoteten Prüfungsvorleistung. Unbenotete Prüfungsvorleistungen sind einer benoteten Prüfung zugeordnet und müssen von den Studenten erfolgreich abgelegt werden, bevor sie an der zugeordneten Prüfung teilnehmen dürfen.

Benotete Prüfungen (Klausuren oder mündliche Prüfungen) werden zum Abschluss eines Moduls erbracht und auf einer Skala von $0 - 15$ bewertet. Tabelle 3.1 zeigt die möglichen Noten, im weiteren Verlauf werden Notenpunkte von $0 - 15$ verwendet.

Notenpunkte	Note	Note
15	0,7	ausgezeichnet
14 bis 13	1,0 und 1,3	sehr gut
12 bis 10	1,7, 2,0 und 2,3	gut
09 bis 07	2,7, 3,0 und 3,3	befriedigend
06 bis 05	3,7 und 4,0	ausreichend
04 bis 00	4,3, 4,7, 5,0 und 6,0	nicht ausreichend

Tabelle 3.1.: Mögliche Noten an der HAW

In dieser Arbeit wurden nur die Studenten betrachtet, die vom Sommersemester 2004 bis zum Sommersemester 2011 immatrikuliert wurden. Da primär Studienverlaufsdaten untersucht werden, wurden hierdurch nur Studenten untersucht, für die ein Verlauf mit realistischer Chance einen Abschluss zu erreichen vorhanden ist. Es handelt sich hierbei um 18461 Prüfungsleistungen, die von 645 Studierenden erbracht werden. Von diesen haben 256 einen Abschluss erreicht und analog dazu 389 keinen Abschluss.

In den frühen Phasen dieser Arbeit wurde mit einem Datensatz, der nur vom Sommersemester 2004 bis zum Sommersemester 2012 erhoben wurde, gearbeitet. Dieser Datensatz war auch deutlich unschärfer erhoben und enthielt alle Studenten, die in der Technischen Informatik in diesem Zeitraum eine Prüfung erbracht haben. Im Gegensatz hierzu ist der für die endgültige Version dieser Arbeit verwendete Datensatz mit Daten aus dem hochschulweiten Immatrikulationssystem (Helios) aufbereitet worden und enthält nur Studenten, deren Immatrikulation nach dem Sommersemester 2004 erfolgt ist.

3.2.2.2. Studiengang

Der 6-semestrige Studiengang Technische Informatik wird an der Hochschule für Angewandte Wissenschaften Hamburg mit einem Bachelor of Science abgeschlossen. Pro Semester (alle 6 Monate) wird er von etwa 60 Studenten begonnen. Das Curriculum besteht schwerpunktmäßig aus Inhalten der Informatik und der Elektrotechnik, während Akzente durch Betriebswirtschaft und gesellschaftswissenschaftliche Kurse gesetzt werden.

Insgesamt werden von den 180 ECTS[4] *Credit Points*, die für einen Bachelor nötig sind, ca. 7% in nichttechnischen Fächern (BWL, Geisteswissenschaften), 15% in Wahlpflichtkursen und Projekten und die verbleibenden 78% in technischen und mathematischen Kursen erbracht.

[4] European Credit Transfer and Accumulation System

Der Studiengang ist in einer Technik und Informatik Fakultät angesiedelt, in der zwei weitere Bachelor-Abschlüsse mit Informatikbezug (Angewandte und Wirtschaftsinformatik) erworben werden können. Alle drei Studiengänge dienen als Vorbereitung für den konsekutiven Masterstudiengang Informatik.

3.2.2.3. Übersicht der Zulassungs- und Immatrikulierungsverfahren (C1)

Die „Abschlussübersicht zum Zulassungs- und Immatrikulationsverfahren“ (C1 Statistik) zeigt, wie viele Studenten in jedem Studiengang immatrikuliert sind. Diese Zahl wird aufgeschlüsselt nach 1. Fachsemester und höheren Fachsemester sowie Beurlaubte und Austauschstudierende. Die Statistik wurde in dieser Arbeit nur genutzt, um auf die Effektgröße von beurlaubten Studenten zu schließen. In frühen Phasen dieser Arbeit wurde sie auch genutzt, um die Datenbereinigung zu validieren indem die korrekten Zahlen der Statistik mit den durch Heuristiken bereinigten Notendaten verglichen wurden. Dieser letzte Punkt wurde schließlich durch einen aktualisierten Datensatz unnötig.

3.2.2.4. Statistik zur Bewerbungserfolgsquote (C2)

Die Statistik C2 wird von der Hochschule betitelt als „Bewerbungen, Zulassungen, Immatrikulationen 1. Fachsemester, Ablehnungen (NC)“. Hier werden die Daten zur Gesamtzahl der zugelassenen Studenten und Bewerbungen in Verbindung mit der Kapazität jedes Studienganges geführt. Die neu zugelassenen Studenten werden jeweils nach ihrer Zulassung in das 1. Fachsemester oder ein höheres Semester aufgeschlüsselt. Die gesamten Bewerbungen in zulassungsberechtigte (alle formalen Bedingungen erfüllend) und nicht zulassungsberechtigte. An dieser Stelle muss der Begriff der Zulassung von dem der Immatrikulation abgegrenzt werden. Die Zulassung erlaubt es einem Studenten sich zu immatrikulieren. Dieses Recht nehmen nicht alle zugelassenen Studenten wahr. Grund hierfür ist häufig eine Mehrfachbewerbung an unterschiedlichen Hochschulen[5].

Aus der Anzahl der Zulassungen und der Anzahl der formal korrekten Bewerbungen wird die in dieser Arbeit verwendete Bewerbungserfolgsquote gebildet. Diese gibt an, welchem Prozentsatz der formal korrekten Bewerbungen ein Studienplatz angeboten wurde. Eine hohe Bewerbungserfolgsquote deutet also darauf hin, dass die Hochschule weniger Auswahl an potenziellen Studenten hatte als bei einer niedrigeren Quote.

[5] Laut mündlichen Berichten der zuständigen Stellen. Die Hochschule erhebt keine Statistiken zum Grund der Nicht-Annahme eines Studienplatzes.

3.2.2.5. Statistik zur Art der Hochschulzugangsberechtigung (C3)

Die von der Hochschule für Angewandte Wissenschaften Hamburg geführte Statistik C3 stellt die Hochschulzugangsberechtigung (HZB) der neu immatrikulierten Studenten aufgeschlüsselt nach den im Hamburger Hochschulgesetz (HmbHG) §37 und §38 geregelten Berechtigungen zum Studium in grundständigen Studiengängen dar. Sie gliedert die neu immatrikulierten Studierenden in die folgenden Gruppen (die entsprechenden Paragrafen des HmbHG werden jeweils vor den Gruppen geführt):

§**37(1) S.1 Nr. 1 u. 2 und** §**39** Allgemeine Hochschulreife

§**37(1) S.2** Fachhochschulreife

§**37(1) S.2** Fachgebundene Hochschulreife

§**37(4)** Künstlerische Befähigung ohne andere HZB

§**38** Berufsausbildung plus 3 Jahre Praxis plus Eingangsprüfung

§**37(1) S.1 Nr. 3-7** Meister/Fachwirt/weitere

Die Kategorien entsprechen umgangssprachlich dem Abitur, der Fachhochschulreife (schulischer und praktischer Teil), der fachgebundenen Hochschulreife (Fachakademie, Berufskollegs etc.), einer künstlerischen Zugangsprüfung (für die Technische Informatik nicht verwendet), der Berufserfahrung plus einer Eingangsprüfung und der beruflichen Fortbildung bis zum Meister oder einem ähnlichen Abschluss. Für den hier betrachteten Studiengang sind in dieser Reihenfolge relevant: Allgemeine Hochschulreife, Fachhochschulreife und Fachgebundene Hochschulreife. Die restlichen Kategorien kommen entweder gar nicht (künstlerische Befähigung) oder nur in Einzelfällen (Berufsausbildung/Meister) im Untersuchungszeitraum vor.

3.2.3. Stetigkeit der Datenbasis

Eine wesentliche Problematik in den erhobenen Daten sind während des Untersuchungszeitraumes erfolgte Veränderungen der Rahmenbedingungen des Studierens. Diese können dazu führen, dass retrospektiv aus historischen Daten gewonnene Erkenntnisse nicht auf aktuelle Studienabläufe und Situationen übertragbar sind. In den betrachteten Daten sind mehrere Arten von strukturellen Veränderungen über die Zeit der Untersuchung erfolgt. Zum einen ein Wechsel der Prüfungsordnung zum Wintersemester 2008 und zum anderen der Wechsel der Lehrenden zwischen den Semestern. Auf beide Fälle wird im Folgenden eingegangen.

3.2.3.1. Wechsel der Lehrenden

Das akademische Personal der Hochschule, das die Lehre durchführt[6], unterliegt zum einen einer Fluktuation durch Zu-/Abgänge, zum anderen wird das gleiche Modul in der Regel im Wechsel durch unterschiedliche Lehrende durchgeführt. Für diese Arbeit standen keine Daten darüber zur Verfügung, wann welcher Lehrende welches Modul durchgeführt hat. Hier wird davon ausgegangen, dass Unterschiede zwischen den Lehrenden im Aggregat zufällig über die Semester verteilt sind und sich nicht auswirken. Eine Validierung dieser Annahme in einer späteren Arbeit wäre durch das manuelle Einbringen der historischen Stundenplandaten möglich und wurde hier sowohl aus Zeit- als auch aus Datenschutzgründen[7] nicht durchgeführt.

3.2.3.2. Wechsel der Prüfungsordnung

Der Untersuchungszeitraum dieser Arbeit berührt zwei Prüfungsordnungen und eine Aktualisierung einer Prüfungsordnung. Er beginnt mit der Prüfungsordnung 2004 (PO2004), die zum Wintersemester des Jahres eingeführt wurde. Zum Wintersemester 2008 wurde die Prüfungsordnung 2008 eingeführt (PO2008) und im November 2010 aktualisiert. Die Prüfungsordnungen können jeweils auf den Webseiten der Hochschule für Angewandte Wissenschaft Hamburg[8] heruntergeladen werden. Im Folgenden wird auf die für diese Arbeit wesentlichen Unterschiede zwischen den Prüfungsordnungen eingegangen, der Einführungsprozess für eine neue Prüfungsordnung beschrieben und die Auswirkung auf diese Arbeit analysiert.

Unterschiede

Die PO2004 war die erste Bachelorprüfungsordnung der HAW im Studiengang Technische Informatik. Module sind in der PO2004 üblicherweise mit 5 Credit Points versehen, was einem Studienaufwand von 125-150 Stunden pro Modul entspricht und zu 6 oder 7 Prüfungsleistungen im Semester führt. Im Gegensatz hierzu sind in der PO2008 Prüfungen in der Mehrzahl mit 6 Credit Points Arbeitsaufwand versehen, was 150-180 Stunden entspricht und zu 5 oder 6 Prüfungsleistungen im Semester führt. Ziel der Umstellung war es, unter anderem die Studierbarkeit der Technischen Informatik durch eine geringere Anzahl von Klausuren und mehr selbstständige Studienarbeit zu erhöhen.

[6] im wesentlichen Professoren

[7] Eine solche Validierung würde eine Art der individualisierten Leistungskontrolle der Lehrenden darstellen.

[8] im Folgenden HAW

Einführungsprozess

Eine neue Prüfungsordnung wird semesterweise beginnend im ersten Semester eingeführt, sodass alle Studenten, die im Sommersemester 2008 immatrikuliert wurden, noch die Möglichkeit hatten, ihren Erstversuch in jedem Modul in der PO2004 durchzuführen. Es existiert jeweils eine Äquivalenzliste, die zeigt, welche Prüfungsleistungen aus einer neuen Prüfungsordnung welchen aus einer alten Prüfungsordnung entsprechen. Nach einer von den Gremien der Selbstverwaltung bestimmten Frist[9], nachdem eine neue Prüfungsordnung eingeführt wurde, müssen sich Studenten, die noch in der abgelösten Prüfungsordnung immatrikuliert sind, umschreiben (d.h. in die neuere Prüfungsordnung wechseln) lassen. Hier kann der Fall eintreten, dass Prüfungen, für die keine Äquivalenzleistungen existieren, neu abgelegt werden müssen.

Analyse

Wesentlicher relevanter Unterschied zwischen den Prüfungsordnungen (POs) sind die Unterschiede im ersten Studienjahr, das im Rahmen dieser Arbeit als Prognosegrundlage genutzt wurde. Hier wurden aufbauend auf den Modulbeschreibungen[10] und den Modulen der Prüfungsordnungen, dort wo es Abweichungen zwischen den POs im ersten Studienjahr gibt, die einzelnen Module zu Modulgruppen zusammengefasst. Der Durchschnittswert der Noten aller Module einer Gruppe wurde dann als Note der Gruppe verwendet. Dieses Vorgehen hat im Wesentlichen nur die mathematischen Fächer sowie Automatentheorie und Formale Sprachen betroffen.

3.2.4. Stakeholder

Im Folgenden wird um Kontext zu geben, auf die unterschiedlichen Parteien eingegangen, die ein Interesse am Studienerfolg (entweder des Einzelnen oder des Studienganges) oder der Studienfortschrittsmessung haben. Dies sind hier Politik (Abschnitt 3.2.4.1), Studierende (Kapitel 3.2.4.2), Prüfungsausschuss (Abschnitt 3.2.4.3,), Studienfachberater (Abschnitt 3.2.4.4), Akkreditierungsbeauftragte (Abschnitt 3.2.4.5) und die für die Studiengangsentwicklung zuständigen Gremien (Abschnitt 3.2.4.6).

[9] ca. 4 Jahre

[10] https://www.haw-hamburg.de/en/ti-i/studium/angewandte-informatik-ba/modulhandbuecher.html

3.2.4.1. Politik

In der Hamburger Politik ist neben der allgemeinen Förderung der Hochschulen ein weiteres Ziel die Sicherung des Bedarfs an Fachkräften in der Metropolregion Hamburg. In der Informatik besteht hier im Moment eine Lücke. Die Hamburger Hochschulen bilden ca. 400 Informatiker pro Jahr aus. Laut Zahlen der IHK sind derzeit ca. 50.000 Informatiker in der Metropolregion beschäftigt. Bei einer durchschnittlichen Lebensarbeitszeit von 25 Jahren als Informatiker ergibt dies einen Bedarf von ca. 2000 neu auszubildenden Informatikern pro Jahr. Die fehlenden Stellen werden durch Zuzug und Fachfremde (Mathematiker, Physiker etc.) gedeckt ((Statistikamt Nord), 2015).

Der Politik stehen zur Steuerung der Hochschulen primär die Zielvereinbarungen, die mit diesen geschlossen werden, sowie das Budget, das durch die Wissenschaftsbehörde gesteuert wird, zur Verfügung.

3.2.4.2. Studierende

Für den Großteil der Studierenden gilt, dass sie ein Interesse daran haben, das aufgenommene Studium erfolgreich abzuschließen. Je nach Studiengang sind sekundäre Faktoren wie Abschlussnote und Studiendauer mehr oder weniger hoch für den späteren Berufseinstieg priorisiert. Es muss auch davon ausgegangen werden, dass es einen gewissen Prozentsatz an immatrikulierten Studierenden gibt, die das Studium aus fachfremden Gründen betreiben und nicht zwingend einen Abschluss zum Ziel haben. Für diese Studierenden stellen zum Beispiel Aufenthaltstitel oder der spätere Wechsel in einen anderen grundständigen Studiengang der Hochschule Gründe für die Immatrikulation dar.

Für diejenigen Studierenden, die als Studienziel den Abschluss haben, wird hier angenommen, dass sie von einer korrekteren Einschätzung ihres Studienfortschrittes, als dieser im Moment gegeben werden kann, profitieren würden. Eine Studienfachberatung, die eine solche Einschätzung geben könnte, können schon aus Kapazitätsgründen nicht alle Studierenden individuell erhalten.

Speziell für den hier betrachteten Studiengang Technische Informatik kann davon ausgegangen werden, dass aufgrund der Arbeitsmarktsituation auch ein grundständiger (Bachelor-) Studiengang zum erfolgreichen Berufseinstieg qualifiziert. Ein Masterstudium nehmen hier primär solche Studierenden auf, die an der Materie Interesse haben. In anderen Studiengängen ist die Fokussierung auf die Zulassung zum Masterstudium (und damit auf die Noten des Bachelorstudiums) deutlich höher. Das klassische Beispiel sind hier Studiengänge, die zu einer Tätigkeit im öffentlichen oder im öffentlich alimentierten Dienst (Sozialberufe)

hinführen. Hier ist im Gegensatz zur Informatik das Erreichen eines fortgeschrittenen teritären Abschlusses (Master oder vergleichbar) häufig notwendige Voraussetzung für eine Karriere im höheren Dienst.

3.2.4.3. Prüfungsausschuss

Aufgabe des Prüfungsausschuss ist es:

- Auf die Einhaltung der Regeln der Prüfung- und Studienordnung zu achten
- Exmatrikulation
- Einhaltung von Fristen
- In Streitfällen zu entscheiden
- Anmeldungen zu Bachelor- und Masterarbeiten durchzuführen
- Anerkennung von Leistungen, die an anderen Hochschulen erbracht wurden.

Dem Prüfungsausschuss wird hierbei bei Routinetätigkeiten vom Fakultätsservicebüro (FSB) sowie bei der Anerkennung von Leistungen von den Studienfachberatern zugearbeitet. Hierdurch konzentriert sich die tatsächliche Arbeit des Ausschusses auf die Bearbeitung von Streitfällen. Historisch wurden auch Sondergenehmigungen durch den Ausschuss erteilt, um Leistungsschranken aus der PO nach dem ersten Studienjahr und zur Wiederholung von Prüfungen nach Antrag auszusetzen. Diese Aufgaben sind inzwischen durch Änderungen der Prüfungsordnungen nicht mehr notwendig[11].

3.2.4.4. Studienfachberater

Die Studienfachberatung hat unter anderem folgende Aufgaben:

- Anerkennung von Studienleistungen anderer Hochschulen/Studiengänge
- Leistungsbescheinigung nach §48 BAföG
- In der PO vorgeschriebene Beratung über den individuellen Studienverlauf bei Überschreitung der Regelstudienzeit um mehr als 2 Semester
- Durch den Studenten initiierte Beratung bei Studienproblemen

[11] Die Änderung erfolgte mit der PO2014, also nach dem in dieser Arbeit betrachteten Zeitraum

Die Rolle des Studienfachberaters wird von mehreren Professoren des Studiengangs ausgefüllt. Sie führen sowohl Beratungen für Gruppen im Rahmen der Orientierungswochen für neu immatrikulierte Studierende durch als auch individuelle Beratungen. Themen der individuellen Beratung sind unter anderem die Anerkennung von Leistungen anderer Hochschulen bzw. anderer Studiengänge und die Beratung bei Überschreiten der Regelstudienzeit um mehr als zwei Semester.

3.2.4.5. Akkreditierung

Derzeit findet die Akkreditierung der Studiengänge alle 3 Jahre durch eine externe Agentur statt[12]. Die Fakultät plant mittelfristig eine Umstellung auf eine Systemakkreditierung, bei der dann nur noch das System der Qualitätskontrolle akkreditiert wird und nicht mehr einzelne Programme.

3.2.4.6. Studiengangsentwicklung

Der Studienreformausschuss (SRA) leitet seine Existenz aus §10 Absatz 3 der Fakultätsordnung der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg vom 7. Oktober 2005 (zuletzt geändert am 21. Dezember 2006) ab[13]. Aufgabe der Studiengangsentwicklung ist, unter Betrachtung der aktuellen und prognostizierten Ergebnisse den Studiengang und damit die Prüfungsordnung weiterzuentwickeln. Hierzu arbeitet er mit den Studiengangskoordinatoren und den durch sie gebildeten Ausschüssen zusammen.

3.3. Vergleichende Betrachtungen

In der Literatur finden sich Arbeiten, die in Teilen der hier vorgestellten ähneln. Besonders erwähnenswert sind hier (Golding und Donaldson, 2006) und (Borrego u. a., 2005) da sie Thesen behandeln, die auch in dieser Arbeit betrachtet werden. So stellt (Golding und Donaldson, 2006) die beiden Thesen H_{0^2} und H_{0^3} auf.

- H_{0^2}: Performance in 1st year programming and computer science courses does not have an impact on student's performance : rejected

[12] Programakkreditierung

[13] siehe: http://www.haw-hamburg.de/en/fakultaeten-und-departments/ti/fakultaetsservicebuero/pruefungsordnungen.html

- H_{0^3}: gender and age does not determine student's level of success in computer science : accepted

Hier decken sich die Ergebnisse aus der Literatur mit den Aussagen dieser Arbeit (siehe auch: (Steenbuck, 2014)). In der beschriebenen Literatur wird angenommen, dass die technischen Veranstaltungen des ersten Studienjahres einen Einfluss auf den Studienerfolg haben, Geschlecht und Alter im Gegensatz dazu keine Rolle spielen. Diese Aussagen decken sich mit den evaluierten Daten in dieser Arbeit (Kapitel 5).

In (Borrego u. a., 2005) wird ein Unterschied zwischen erfolglos aus dem Studium abgehenden weiblichen und männlichen Studenten festgestellt. Hier gehen erfolglose weibliche Studenten mit besseren Noten von der Hochschule ab als erfolglose männliche Studenten. Diesen Zusammenhang gibt es in dem hier untersuchten Studiengang nicht[14].

3.4. Ethische Fragen

Die Entwicklung von Systemen, die personenbezogene Daten auswerten, erfordert immer auch eine Betrachtung der (ungewollten) Konsequenzen dieser Nutzung personenbezogener Daten. Die in dieser Arbeit beschriebenen Analysen finden auf anonymisierten Daten statt. Ein späterer Einsatz einiger Teile der entwickelten Funktionalität würde aber eine Deanonymisierung bzw. ein Arbeiten auf nicht anonymisierten Daten notwendig machen. Im Folgenden werden einige absehbare Problematiken angesprochen. Diese Aufzählung kann schon aus dem Grund, dass dieser Arbeit kein konkreter Einsatz zugrundeliegt, nicht abschließend sein, sie sollte nur als initialer Startpunkt einer späteren vollständigen Technikfolgenabschätzung verstanden werden. Die angesprochenen Punkte sind unberechtigter Zugriff auf die Daten (Datensicherheit), berechtigter Zugriff (Autorisierung), ungewollte und Rückkopplungseffekte (selbsterfüllende Prophezeiung) und Konflikte in der Zielsetzung (Zielkonflikte).

3.4.1. Datensicherheit

Ein elementarer Punkt ist die Sicherheit der datenhaltenden Systeme gegenüber nicht autorisierten Zugriffen. Je nach Ausprägung, in der die Methoden aus dieser Arbeit oder ähnlichen in den universitären Betrieb übernommen werden, stellen sich hier unterschiedliche Herausforderungen. Wenn die verwendeten personenbezogenen Daten in ihren Ursprungssystemen verbleiben und auch nur dort zusätzliche Kennzahlen persistiert werden, wo schon Ursprungsdaten vorhanden sind, ist die zusätzliche Angriffsfläche wahrscheinlich geringer als

[14] Durchschnittsnote erfolglos Studierender: M(8, 4), W(8, 37)

dort, wo komplett neue Systeme geschaffen werden. Nichtsdestotrotz wird eine Überprüfung der ursprünglichen vorhandenen Sicherheitskonzepte unter Einbeziehung der neu geschaffenen Daten angeraten .

3.4.2. Autorisierung

Eine der Fragen, die in der Hochschule unter Teilnahme aller betroffenen Statusgruppen gestellt werden muss, ist bei der algorithmischen Prognoseerstellung, wer auf die resultierenden Daten zugreifen darf und wie dieser Zugriff ausgeprägt sein wird. Als leitende, aber möglicherweise orthogonal zueinander stehende Prinzipien treten hier zum einen der Nutzen der Analyse und zum anderen die anzustrebende Datensparsamkeit auf. Es scheint sinnvoll, zurückhaltenden Umgang gerade mit personalisierten Prognosen zu üben die per Definition mit einer Prognoseungenauigkeit behaftet sind.

Unterschiedliche Statusgruppen, die in der Zugangskontrolle vorstellbar sind, sind unter anderem Professoren, Studienfachberater und Studierende. Denkbar wäre so zum Beispiel, eine Prognose nur über einen Studienfachberater verfügbar zu machen. Hierdurch können die Daten mit einer fachgerichteten Beratung verbunden werden.

3.4.3. Selbsterfüllende Prophezeiung

Bei vorausschauender Klassifikation von Studenten entstehen mehrere Probleme, die unter den Begriff der selbsterfüllenden Prophezeiung zusammengefasst werden können. So kann zum einen durch Rückkopplungseffekte die Prognosegenauigkeit gesenkt werden, wenn die Mitglieder der Hochschule ihr Verhalten an prognoserelevanten Faktoren ausrichten, bzw. die Prognose als unumstößliche Wahrheit aufnehmen und dadurch ihr Verhalten verändern. Das beobachtete System reagiert also auf die Beobachtung und invalidiert durch diese Reaktion bisher gültige Annahmen.

Der andere zu befürchtende Effekt betrifft die Auswirkung einer negativen oder positiven Prognose auf das Selbstwertgefühl des Studenten. Studien haben gezeigt, dass ein Schüler/Student, der eine negative Prognose erhält, schlechtere Leistungen erbringt als ein Student, dem eine positive Prognose ausgestellt wurde. Hier kann eine Art „Vorverurteilung“ für Studierende entstehen, denen nach relativ kurzer Studiendauer eine negative Prognose ausgestellt wird.

3.4.4. Zielkonflikte

Es muss eine klare, unter allen Statusgruppen der Hochschule abgestimmte Zielsetzung für ein Prognoseverfahren geben, das eingesetzt wird, um studienrelevante Entscheidungen zu treffen. Gibt es eine solche Zielsetzung nicht, besteht die Gefahr, dass personenbezogene Prognosen eingesetzt werden, um Ziele anzustreben, die den Interessen des jeweilig betrachteten Studenten entgegenlaufen. So könnte beispielsweise die Prognose, ob ein Student einen Abschluss erhält oder nicht, dazu genutzt werden, Ressourcen der Hochschule auf die prognostizierten “High Performer“ zu konzentrieren.

3.5. Fazit

Aus der Analyse der derzeitigen Stisysverwendung und der am Prozess Beteiligten wurden für diese Arbeit als wesentliche zu untersuchende Fragestellungen ausgewählt, ob die im Folgenden genannten Prognosen aus den Daten des ersten Studienjahres möglich sind.

- Prognose des Studienerfolgs
- Prognose der Studiendauer

Im Folgenden wird auf diese beiden Prognosen eingegangen. Neben diesen stellt sich auch die Frage, ob Studierenden ein Maß für den Grad ihrer Zielerreichung hinsichtlich des Abschlusses gegeben werden kann, das umfangreicher ist als das Zählen von Credit Points.

3.5.1. Prognose des Studienerfolgs

Auf der Ebene der Studiengangsentwicklung und des Qualitätsmanagements wird der Studienerfolg einer Kohorte prognostiziert, um unter anderem den Erfolg einer neuen Prüfungsordnung abzuschätzen. Dies ist wünschenswert, da eine retrospektive Analyse erst nachdem der Großteil der Studenten einer Kohorte exmatrikuliert[15] ist, also nach vier bis fünf Jahren möglich ist.

Ebenso ermöglicht ein frühzeitiges Erkennen von Problemen mit der Studierbarkeit einer Prüfungsordnung Maßnahmen unterhalb der Grenze, bei der eine Anpassung der Prüfungsordnung nötig wäre, zu ergreifen. Eine solche Anpassung ist beispielsweise die Änderung von Prüfungsmodalitäten[16].

[15] nach einem Abschluss oder dem Studienabbruch
[16] etwa der Zeitpunkt, zu dem der Prüfende bekannt gegeben wird

3.5.2. Prognose der Studiendauer

Die Frage nach der Studiendauer der einzelnen Studierenden dient der Fokussierung von Ressourcen der Hochschule. So ist denkbar, dass gezielt solche Studierende angesprochen werden, deren Studienerfolg in Frage steht, für die aber eine lange Studiendauer prognostiziert wird. Ziel einer solchen Ansprache könnte sein, die Studierenden mit den Hilfestellungen vertraut zu machen, die die Hochschule anbietet (Zentrale Studierendenberatung, Fachtutorien etc.).

3.5.3. Studienfortschrittsmessung für den einzelnen Studierenden

Primär durch die Studienfachberater wird eine Kontextualisierung der individuellen Studienleistungen erbracht. Hierbei wird der Studienfortschritt des einzelnen Studenten in der Studienberatung durch den beratenden Professor im Vergleich mit den Kommilitonen des einzelnen betrachtet und mit diesen verglichen. Diese Aufgabe führen die Studienfachberater primär durch die ihnen innewohnende Erfahrung über typische Studienverläufe und punktuell unterstützt durch Anzeige der Notendaten des einzelnen Studieren in Stisys durch.

Dem einzelnen Studierenden wird keine Möglichkeit gegeben, seinen Studienfortschritt zu messen, außer durch das Zählen der erreichten Credit Points. Diese stellen wegen der in der Praxis deutlich unterschiedlich herausfordernden Prüfungen für unterschiedliche Module nur einen ungenauen Näherungswert dar.

4. Design/Implementierung

Im Kapitel Design/Implementierung wird zuerst auf den Entwurf und dort primär auf die Datenbank und den ETL[1]-Prozess mit dem dem die Datenbank erstellt wurde, eingegangen. Danach wird die in Ruby on Rails entwickelte Oberfläche zur Datenvisualisierung und Exploration gezeigt und abschließend auf Probleme der Übertragbarkeit von Daten zwischen Prüfungsordnungen eingegangen.

4.1. Datenfluss/Anwendungsdesign

Dieses Kapitel gibt einen abstrakten Überblick über den Datenfluss und die Komponenten des für diese Arbeit verwendeten Systems. Abbildung 4.1 zeigt die Komponenten des Systems, die im Folgenden erläutert werden. Der Datenfluss ist grob von links nach rechts.

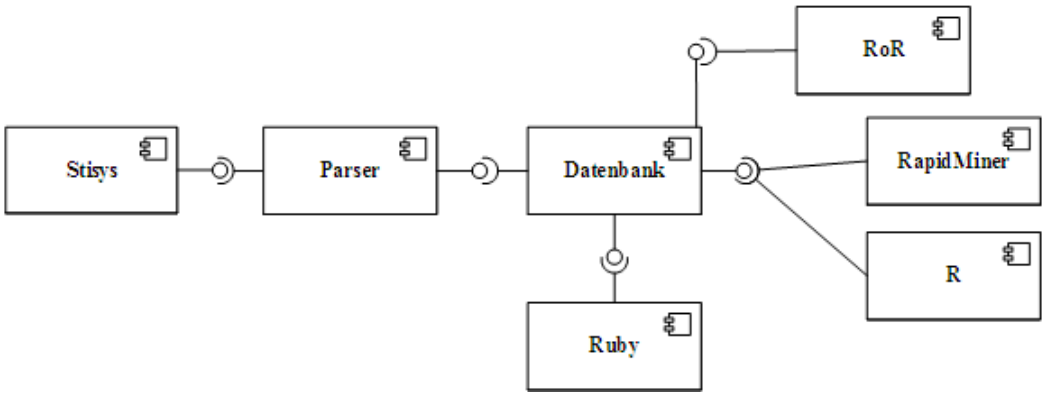


Abbildung 4.1.: Technische Architektur/Datenfluss

Die einmalig in einem CSV-ähnlichen Format aus Stisys exportierten Prüfungsdatensätze werden mittels eines in Antlr generierten Parsers[2] weiterverarbeitet. Ausgabeformat des Parsers ist SQL, das als SQL-Skript persistiert wird. Im Parser findet die erste simple Anreiche-

[1] Extract Transform Load
[2] siehe Listing A.1 für die Parser-Grammatik

rung der Daten mit berechneten Attributen[3] statt. Anschließend läuft ein bash-Skript[4], das die Datenbank anlegt, die vom Parser abgelegten Daten einfügt und die weitere Anreicherung der Daten mit berechneten Attributen startet. Die weitere Anreicherung findet entweder direkt in der Datenbank mittels MySQL *stored procedures* oder per Ruby Skript statt. Die Datenbank wird in Kapitel 4.2 beschrieben.

Lesend auf die Datenbank greifen die drei in Abbildung 4.1 als *RoR*, *RapidMiner* und *R* bezeichneten Komponenten zu. Die Ruby on Rails (RoR) Komponente stellt die gespeicherten Daten in übersichtlicher Form bzw. grafisch auf einer Webseite dar, sie wird in Kapitel 4.5 näher beschrieben. RapidMinder und R greifen direkt auf die Datenbank zu, um statistische Auswertungen durchzuführen. Grundlagen zu den in dieser Arbeit verwendeten statistischen Verfahren sind in Kapitel 2.2 erläutert. Details zu den Möglichkeiten und Anwendungsbeispiele zu RapidMiner und R sind in Kapitel 2.3 gezeigt.

4.2. Datenbank

Im Folgenden wird auf das Design der in dieser Arbeit genutzten Datenbank eingegangen. Zuerst wird das Datenmodell in einer an ein Entity-Relationship-Modell (ERM) angelehnten Form in Abbildung 4.2 gezeigt, hier sind sinnvolle Teile der Datenbank farblich voneinander abgesetzt. Danach werden die einzelnen Module des Datenmodells Studentendaten (4.2.1), Statistische Daten (4.2.2) und Kurs- und Prüfungsdaten (4.2.3) erläutert. Listing A.2.2 im Anhang zeigt das SQL, mit dem die Datenbankstruktur generiert werden kann bzw. wurde.

[3] z.B. Alter bei Immatrikulierung
[4] siehe Listing im Anhang: A.2.3

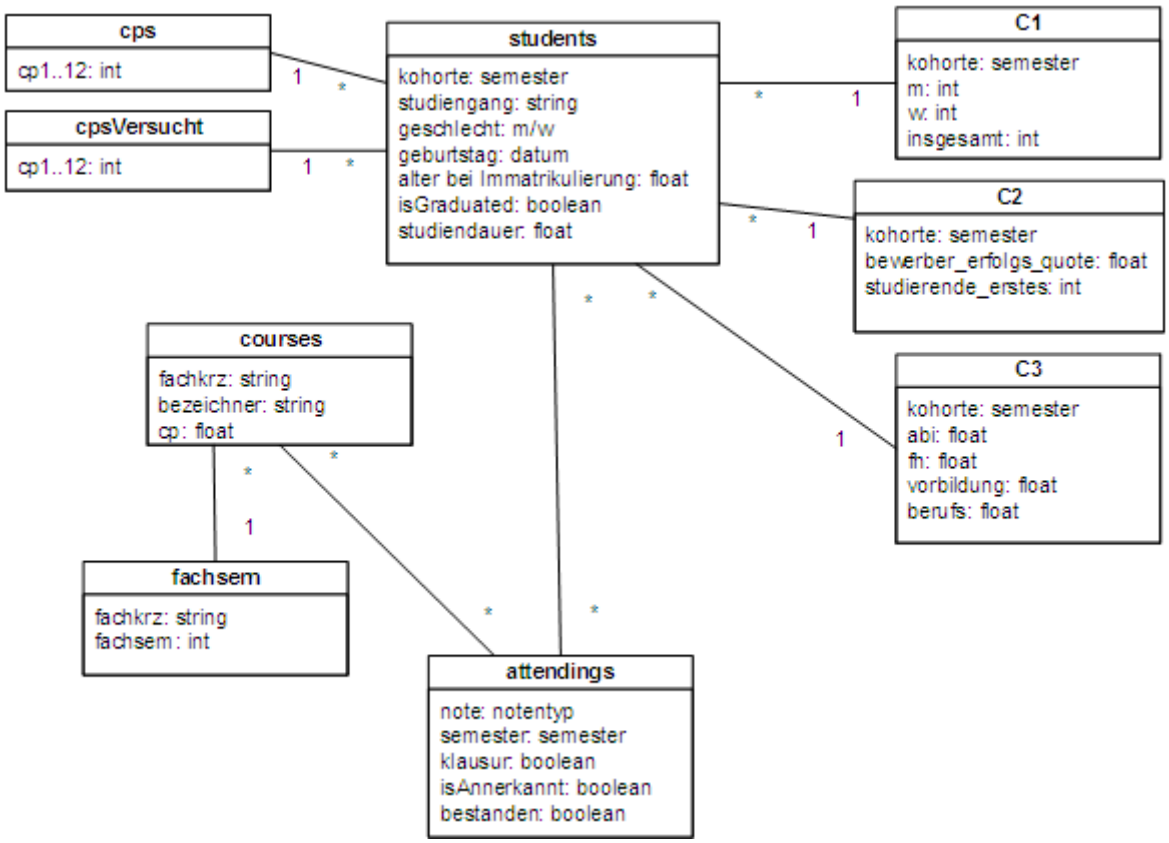


Abbildung 4.2.: Datenmodell (ERM ähnlich)

Nachfolgend wird auf die Teile des Modells spezifischer eingegangen und wesentliche Inhalte werden erläutert. Hierbei wird die Konvention verfolgt, eine sprechende Bezeichnung im Text zu verwenden und in Klammern nachführend den (*technischen Namen*) zu nennen. Synthetische IDs ohne eigenen fachlichen Mehrwert werden hier nicht gezeigt, aber in der Datenbank für Studenten und Kurse sowie die dazugehörigen Verknüpfungen gegeben.

Die Tabellen *cp*, *cps* und *fachsem* passen insofern nicht in die Datenbank, als dass die enthaltenen Daten auch direkt an den zugehörigen Tabellen geführt werden könnten. Das unelegante Design der Datenbank ist hier in der unterschiedlichen Herkunft der Daten begründet. Die genannten Tabellen kommen nicht aus dem Parser-Prozess, sondern sind Stammdaten, die in einem späteren Schritt in die Datenbank nachgeladen und daher technisch getrennt behandelt werden. Sofern die Daten nicht direkt aus dem Stisysdatensatz stammen, wird in Kapitel 4.3 beschrieben, wie sie berechnet wurden.

4.2.1. Studentendaten

Zentrales Element der Datenbank ist die Liste der Studenten in **students**, die neben demografischen Daten (*geschlecht, geburtstag, alter bei immatrikulierung*) über jeden Studenten studienspezifische Daten (*isGraduated, kohorte, studiengang, studiendauer*) führt. *isGraduated* ist ein boolscher Marker, der angibt, ob der Student das Studium erfolgreich abgeschlossen hat. Die Studiendauer entspricht dem Zeitraum[5] zwischen Immatrikulierung und letzter Prüfung im Datensatz.

Die Tabellen **cps** und **cpsVersucht** enthalten vor berechneten Daten dazu, wie viele *Credit Points* ein Student in den Semestern 1 bis 12[6] jeweils versucht und erreicht hat. Die Daten werden hier geführt, um spätere Auswertungsschritte möglichst frei von Berechnungslogik halten zu können.

4.2.2. Statistische Daten

Statistische Daten aus den offiziellen Berichten der HAW finden sich in den Tabellen **C1**, **C2** und **C3**, die Benennung der Tabellen entspricht dem üblichen Sprachgebrauch von der C*x*-Statistik. Die Tabellen haben jeweils die Kohorte (*kohorte*) als Schlüssel zum Studenten.

Die Tabelle **C1** enthält Daten aus der Übersicht zum „Zulassung- und Immatrikulationsverfahren“, dies sind die Menge von im Studiengang immatrikulierten Studenten männlichen und weiblichen Geschlechts (*m/w*) sowie die Gesamtsumme[7] (*insgesamt*).

In der **C2-**Tabelle werden Informationen aus der Statistik „Bewerbungen, Zulassungen, Neuimmatrikulationen 1. Fachsemester, Ablehnungen (NC)“ geführt. Vorhanden sind hier Daten über die Anzahl der im ersten Semester neu zugelassenen Studenten (*studierende_erstes*). Die Gesamtzahl an Zulassungen wäre diese Zahl addiert zur Zahl der Zulassungen in höheren Fachsemestern. Weiterhin enthält die Tabelle Daten über die Bewerbungserfolgsquote (*bewerber_erfolgs_quote*). Die Bewerbungserfolgsquote ist der prozentuale Anteil an formal zulässigen Bewerbern, denen ein Studienplatz angeboten wurde, eine Bewerbungserfolgsquote von 100% bedeutet also, dass jedem Bewerber ein Platz angeboten wurde. Daraus ergibt sich nicht zwingend, dass alle Bewerber diesen Platz auch angenommen haben. Üblicherweise entstehen hohe Quoten durch nachträgliche Zulassungen von der Warteliste.

Die „Schulische Vorbildung der Studierenden im 1. Fachsemester“ ist in Tabelle **C3** geführt. In Prozenten der neu zugelassenen Studenten im ersten Fachsemester werden hier die

[5] in Semestern
[6] 12 Semester ist die doppelte Regelstudienzeit
[7] üblicherweise: *summe(w) + summe(m)*

Hochschulzugangsberechtigungen nach Quote der Abiturienten (*abi*), derjenigen, die über eine Fachhochschulreife[8] und derjenigen, die über eine fachgebundene Hochschulreife[9] verfügen, geführt. Neben diesen primär schulischen Zugangsberechtigungen werden diejenigen, die über eine Eingangsprüfung nach Berufsausbildung und mind. dreijähriger Praxis oder eine Meisterschule (*vorbildung*) für einen Studienplatz zugelassen diejenigen, aufgeschlüsselt.

4.2.3. Kurs- und Prüfungsdaten

Zentrale Kursdaten sind in der Kurstabelle (**courses**) geführt. Hier wird jeder Kurs mit Fachkürzel (*fachkrz*), sprechendem Bezeichner (*bezeichner*) und den Credit Points, die durch den Kurs zu erreichen sind gespeichert (*cp*).

Die Fachsemester-Tabelle (**fachsem**) führt die den einzelnen Prüfungen in der Prüfungsordnung zugeschriebenen Fachsemester (*fachsem*), d.h. das Hochschulsemester, in dem ein Student in Regelstudienzeit diese Prüfung erbringen würde.

4.2.4. Views

Auf der Datenbank werden die Views **big** und **studentsKlausuren** geführt, die die normalisierten Daten wieder zusammenführen, um sie manuell (*big*) bzw. maschinell (*studentsKlausuren*) weiterzuverarbeiten. Im Folgenden werden diese Views kurz erläutert.

Big entspricht in seinem Design den Daten, wie sie aus Stisys importiert wurden, indem eine Zeile pro Prüfungsereignis vorhanden ist, die das Ergebnis der Prüfung und alle zu dem Studenten gehörigen Daten enthält. Diese Darstellung wurde für die manuelle Analyse von Studienverläufen verwendet, da sie eine übersichtliche Analyse der Daten erlaubt.

Der View **studentsKlausuren** stellt die für das Datamining in der Regel erforderliche, alle Daten beinhaltende Tabelle. Hier ist pro Student ein Datensatz vorhanden, der als Attribute die Prüfungsnoten[10], die erreichten und versuchten Credit Points sowie alle demografischen Daten über den Studenten enthält.

[8] typischerweise 12(G9) bzw. 11(G8) Jahre Gymnasium + 3-jährige IHK-Berufsausbildung
[9] typischerweise 12(G9) bzw. 11(G8) Jahre Gymnasium + 1 Jahr fachgerichtetes Praktikum
[10] sollten für den relevanten Zeitraum mehrere Noten zur Auswahl stehen, die beste

4.3. Datenverdichtung

Im Folgenden wird beschrieben, wie aus den flachen Datensätzen des Exports hierarchische Informationen über Studenten, Kurse und Prüfungsleistungen generiert werden. Hierzu wird semiformal ausgedrückt eine Transformation benötigt, durch die aus einer Menge von Prüfungsdatensätzen p ein Studentendatensatz s und die zu diesem gehörenden Prüfungen s_p erzeugt werden, wobei gilt: $s_p = p$. Hier werden in Kapitel 4.3.1 die in den Quelldatensätzen vorliegenden Felder beschrieben und in den folgenden Kapiteln die Regeln für die berechneten Attribute.

4.3.1. Quelldatensätze

Listing 4.3.1 zeigt beispielhaft 3 Quelldatensätze, in denen farblich die unterschiedlichen enthaltenen Datenarten hervorgehoben sind. Die einzelnen Felder mit ihrer Zuordnung sind darauffolgend beschrieben.

Ursprungsdaten

```
1 27,645,2006WS,B-TI,2012WS,6,n.e.,VSP,11277,Praktikum
    Verteilte Systeme,13,,16.11.79,M
2 28,646,2006WS,BCH,2006WS,5,12,MA1,10645,Mathematik 1,1,
    ,05.09.84,M
3 29,646,2006WS,BCH,2006WS,"2,5",13,PRP1,10646,Praktikum
    Programmieren 1,1, ,05.09.84,M
```

- Identifikatoren
  - Exportzeilennummer
  - Fortlaufende ID des Studenten (studId)
- Immatrikulationsdaten
  - Immatrikulationssemester
  - Derzeitiger Studiengang
- Prüfungsdaten
  - Prüfungssemester
  - ECTS Punkte
  - Notenpunkte (bzw. erf. n.e)

  - Fachkürzel
  - FachId
  - Fachbezeichnung
  - Hochschulsemester der Prüfung
  - Prüfung wurde an einer anderen Hochschule geleistet und anerkannt (isAnerkannt)
- Demografische Daten
  - Geburtsdatum
  - Geschlecht

Es liegt somit eine Menge von Prüfungsdatensätzen mit redundanten Daten zum Prüfling vor, die in eine strukturierte Form überführt werden sollen. Die Struktur der Zieldaten wird in 4.2 beschrieben. Die Transformationsregeln werden in den darauffolgenden Kapiteln beschrieben.

4.3.2. Erzeugung Studentendatensätze

Die zu einem Studierenden gehörenden Datensätze werden an der `studId` erkannt und diesem zugeordnet. Nach dieser Gruppierung der Prüfungsdatensätze wird ein Studentendatensatz erstellt, indem die studentenspezifischen Merkmale, die in der Gruppe gleich sind (Kohorte, Studiengang, Geschlecht, Geburtstag) mit einer eindeutigen `id` und berechneten Merkmalen (`Alter bei Immatrikulierung`, `Abschluss bestanden`, `Studiendauer`) kombiniert werden.

Die berechneten Werte ergeben sich wie folgt aus den Grundwerten:

Alter bei Studienbeginn : Erster Tag des Kohortensemesters $-$ Geburtstag

Abschluss bestanden :

Hat eine Bachelorarbeit mit Note > 4 abgeschlossen

Erbrachte ECTS Punkte $>= 180$

Studiendauer : Maximales im Datensatz vorhandenes Semester minus Semester des Studenten

4.3.3. Erzeugung Prüfungsdatensätze

Für die Prüfungsdatensätze ergeben sich die berechneten Attribute wie folgt:

isAnerkannt : isAnerkannt in den Originaldaten ist $=$ Ja

isKlausur : Note ist numerisch

Bestanden : isKlausur und Note > 4 oder Note ist 'erf'

pruefungsversuch : Der wievielte Versuch der Prüfung ist dieser[11]

CPvorPruefung : Wie viele CP hat der Student in Summe aller Semester vor diesem Prüfungsversuch erreicht

CPVersuchtVorPruefung : Wie viele CP hat der Student in Summe aller Semester vor diesem Prüfungsversuch versucht zu erreichen

4.3.4. Erzeugung Kursdatensätze

Die Menge der Kurse ergibt sich aus den eindeutigen Fachkürzeln in den Quelldaten. Die in den Quelldaten verwendeten FachIds sind aufgrund von Änderungen der Prüfungsordnungen und verschiedenen Eingabevarianten nicht eindeutig. Eine Berechnung von zusätzlichen Attributen findet für Kurse nicht statt.

4.4. Datenbereinigung

In ersten Versionen dieser Arbeit und während die initialen Analysen durchgeführt wurden, die zur Anwendung von Regressionsanalysen geführt haben, wurde auf einem Datenbestand gearbeitet, der aufgrund von Erfassungsmodalitäten und der Abfragetechnik nur teilweise in der Form, in der er ursprünglich an den Autor übergeben wurde, Verwendung finden konnte. Im Folgenden wird auf zwei unterschiedliche Arten eingegangen, auf die Rauschen in den verwendeten Datensatz gelangt ist, und wie dieses behoben wurde, dies sind zum einen die im Design von Stisys begründeten, zum anderen die in der Art der Datenerhebung entstehenden[12].

Die Auswertungen, die für die endgültige Version dieser Arbeit auf einem aktualisierten Datenbestand durchgeführt wurden, sind nur noch von den unter 4.4.1 genannten, aus dem

[11] Bei mehreren Versuchen in einem Semester werden diese anhand der Note sortiert.

[12] Unreinheiten bezieht sich hier nur auf die Verwendung der Daten für diese Arbeit, im Kontext von Stisys als Datenquelle sind die übermittelten Daten unbestritten korrekt.

Design von Stisys resultierenden Problemen betroffen. Die unter 4.4.2 behandelten Probleme mit der Datenerhebung konnten für die finale Version der Arbeit stisysseitig durch die Integration der hochschulweiten Immatrikulationssysteme gelöst werden, hierdurch wurden nur noch Studenten mit den korrekten Immatrikulationssemestern erfasst.

4.4.1. Design der Datenquelle

Stisys weist einige für diese Arbeit problematische Designentscheidungen auf, die sich über den Datenimport ausgewirkt haben. Dies sind im Wesentlichen die drei in der folgenden Tabelle genannten:

1. Keine kanonischen Fachsemester
2. Feingranulare IDs
3. Fachkürzel resultierend aus Prüfungsordnungen

Keine kanonischen Fachsemester: In der Regel kann jede Prüfung über die Prüfungsordnung einem Fachsemester zugeordnet werden, in dem sie stattfinden sollte. Für die Kurse mit Wahlfreiheit der Studierenden ist diese Zuordnung nicht möglich, da sie in unterschiedlichen Fachsemestern stattfinden. Dies betrifft für den hier betrachteten Studiengang die Klassen der gemeinschaftswissenschaftlichen Fächer (GWs) und der Wahlpflichtfächer (WPs). Aus beiden Klassen legen die Studenten mehrere Prüfungen in unterschiedlichen Semestern ab. Welche konkreten Prüfungen dies sind, können die Studierenden aus einem vorhandenen Angebot größtenteils frei wählen. Hierdurch entsteht eine implizite Zuordnung einer konkreten Prüfung zu einem konkretem Fachsemester für jeden Studenten. Beispielsweise müsste der erste WP in Fachsemester 4 belegt werden. Am konkreten Kurs nehmen dann sowohl 4. Semester, für die dies der erste WP ist, als auch 5. und 6. Semester, für die dies der zweite bzw. dritte WP ist, teil. Die hierdurch auftretenden Probleme wurden gelöst, indem das betreffende Fachsemester jeweils dynamisch ermittelt wird.

Feingranulare IDs: In der Vorbereitung dieser Arbeit war ursprünglich geplant, die durch Stisys gelieferten IDs der Veranstaltungen zu nutzen, um diese zu identifizieren. Die Analyse der tatsächlich gelieferten Daten hat gezeigt, dass es wechselnde IDs für die gleiche Veranstaltung in unterschiedlichen Semestern gibt. Hier wurde die eindeutige Identifikation der Veranstaltungen über die Fachkürzel durchgeführt. Dies hat nur in einem einzigen Fall zu manueller Nacharbeit geführt[13].

[13] Kombivorlesung SE2/SY/PL der PO2008

Fachkürzel resultierend aus Prüfungsordnungen: Ursprünglich wurde angenommen, dass die Fachkürzel gleicher Prüfungen über Prüfungsordnungswechsel gleich bleiben würden. Dies war nicht in allen relevanten Fällen festzustellen. Insbesondere betroffen waren die Elektrotechnischen Vorlesungen der ersten Semester. Hier wurden künstliche Identifikatoren eingeführt, um eine einfache technische Verarbeitung sicherzustellen. Siehe hierzu auch Kapitel 3.2.3, in dem die Stetigkeit der Daten über Prüfungsordnungen hinweg behandelt wird.

4.4.2. Art Datenerhebung

In der ersten Hälfte der Entwicklung dieser Arbeit wurde mit einem älteren Datensatz gearbeitet, der aufgrund der Erhebungsmethodik mit einer deutlichen Unschärfe versehen war. Im Folgenden werden kurz die Ursachen dieser Unschärfe und die Heuristiken, die zur Behebung der Unschärfe verwendet wurden, erläutert. In der finalen Version dieser Arbeit lag ein Datensatz vor, der aufgrund der Anreicherung mit externen Daten nur genau die gewünschten Studenten beinhaltet hat (siehe 3.2.2.1).

Initial wurde ein Abzug von *Stisys* durchgeführt, der alle Prüfungsereignisse vom Wintersemester 2004 bis zum Wintersemester 2012 beinhaltet hat[14]. Es wurden also nicht Studenten erhoben, sondern Prüfungsereignisse. Beinahe zwangsläufig finden sich unter diesen Ereignissen auch solche, die Studenten betreffen, die nicht im Untersuchungszeitraum immatrikuliert wurden. Dies stellte ein Problem dar, da die Analyse auf am Immatrikulationsdatum hängenden Daten aufgesetzt ist[15]. Die im Folgenden beschriebenen zwei Heuristiken wurden exemplarisch durch Recherche in Stisys validiert.

1. Studiengang
2. Nur Prüfungen höherer Fachsemester im ersten Hochschulsemester

Es wurde geprüft, ob die durch Heuristiken ausgefilterten Studenten nicht im Untersuchungszeitraum immatrikuliert wurden. Die Validierung war erfolgreich und hatte keine falsch erkannten Studenten gezeigt. Eine Validierung des gesamten Datensatzes war aufgrund des hierfür notwendigen Aufwandes bei einzelnen Mitarbeitern der Hochschule nicht durchführbar.

Es wurden alle Studierenden aus dem Datensatz entfernt, die im Studiengang *DPL*[16] eingeschrieben sind.

[14] 32604 Datensätze
[15] z.B. Studiendauer und Prüfungsordnung
[16] Diplomstudiengang nur vor 2004 angeboten

Aus dem gesamten Datensatz wurden solche Studenten entfernt, die in ihrem ersten Hochschulsemester Prüfungen aus höheren Fachsemestern abgelegt haben[17] sowie nie Prüfungen aus dem ersten Fachsemester abgelegt haben. Diese Studenten werden hier ausgefiltert, da es sich wahrscheinlich um solche handelt, die in einem Semester vor dem Wintersemester 2004 immatrikuliert wurden und dann eine Studienpause eingelegt haben.

Durch diese Maßnahmen wurden etwa ca. 180 Studentendatensätze von der Verarbeitung ausgeschlossen. Zu diesem Zeitpunkt war absehbar, dass ein bereinigter Datensatz zur Verfügung stehen wird, was zur Einstellung weiterer heuristischer Bemühungen geführt hat.

4.5. Ruby on Rails Webapplikation

Im Rahmen dieser Arbeit wurde in Ruby on Rails eine Webapplikation implementiert, die ursprünglich der Ergebnisvisualisierung dienen sollte (vgl. Steenbuck, 2015b). Durch Verschiebungen des Fokus zwischen vorbereitenden Projekten und dieser Arbeit wurde diese Oberfläche nur noch zur Visualisierung von Studentendatensätzen und zur Erstellung und Verbindung von Semesterverlaufsdaten mit Studienprognosen verwendet. Die entsprechenden Seiten der Applikation werden hier kurz dargestellt. Diese sind zum einen die von Prof. Dr. Meisel entwickelte erweiterte Kreth-Hörnstein[18]-Darstellung und die Ansicht einzelner Studentendaten.

4.5.1. Studentenvergleich

Abbildung 4.3 zeigt den Verlauf einer Kohorte aufgeschlüsselt nach den Credit Points, die Studenten nach jedem Semester erreicht haben. Die Tabelle ist sortiert nach den CPs, die die Studenten im ersten Semester erreicht haben. Diese Darstellung entspricht funktional der fakultätsintern als erweiterte Kreth-Hörnstein-Analyse bezeichneten Analyse.

[17] an der HAW geschrieben, nicht anerkannt

[18] zur Kreth-Hörnstein-Analyse siehe https://www.haw-hamburg.de/en/qualitaet-in-der-lehre/lehrelotsen/hochschulweite-projekte/studienerfolgsmessung.html

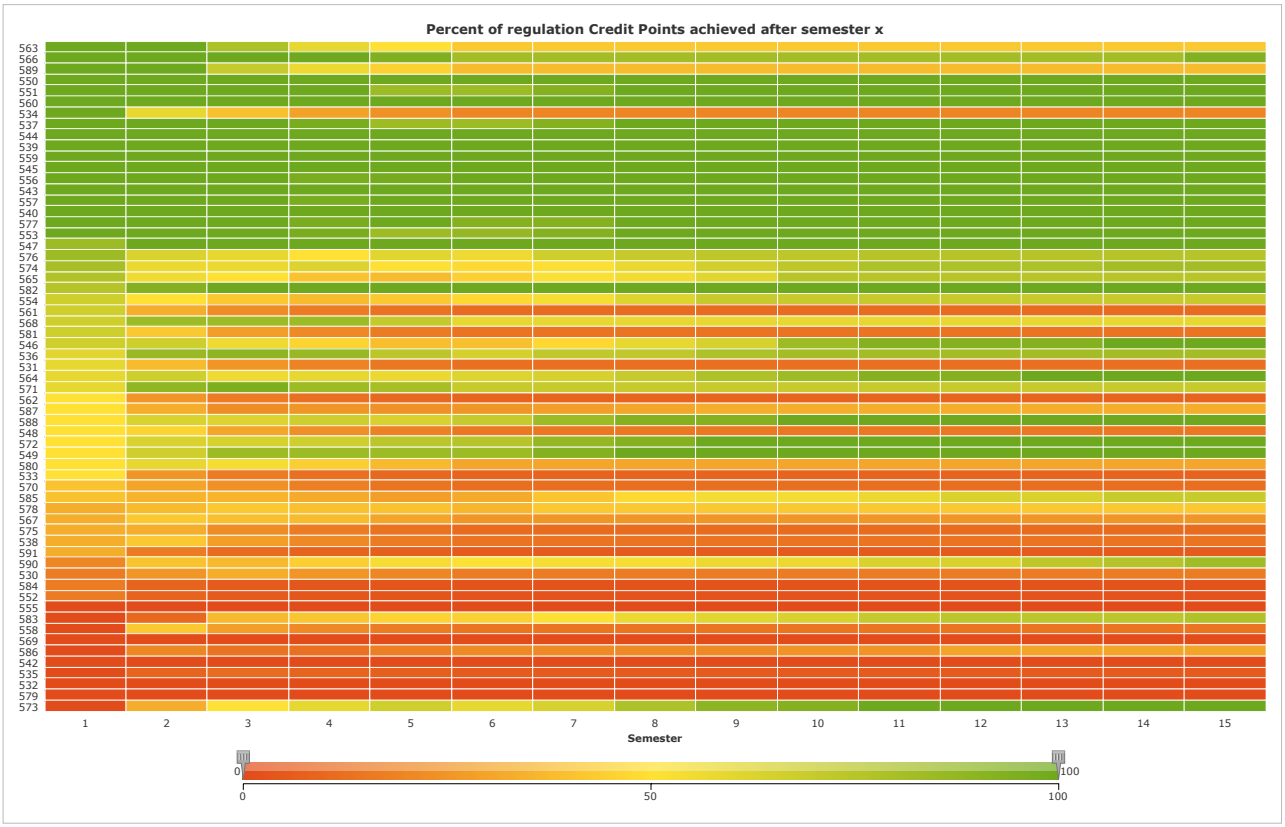


Abbildung 4.3.: Erweiterte Kreth-Hörnstein-Analyse

Gezeigt wird ein Semester. Auf der y-Achse sind die einzelnen Studenten aufgetragen. Auf der x-Achse die Hochschulsemester 1 bis 15[19]. Die einzelnen Zellen sind entsprechend der prozentual von den möglichen Credit Points erreichten, die der auf der y-Achse stehende Student im auf der x-Achse stehendem Hochschulsemester erreicht hat eingefärbt. Rot entspricht 0 % erreichter Credit Points und Grün 100 % erreichter Credit Points (bezogen auf die in diesem Fachsemester geforderten).

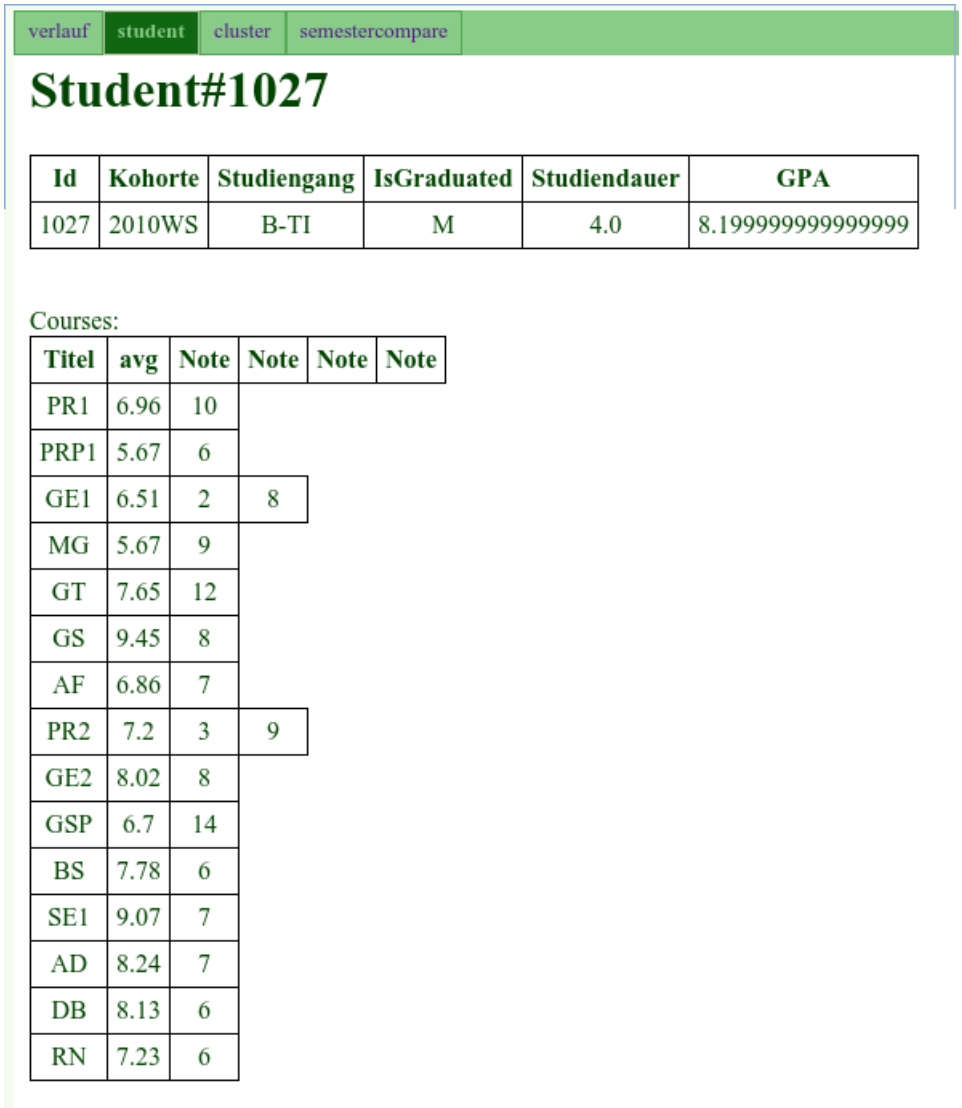
In der hier gezeigten Darstellung ist auf der y-Achse als Label die ID der Studierenden aufgetragen. Hier kann auch die prozentuale prognostizierte Wahrscheinlichkeit, das der jeweilige Student einen Abschluss erreicht, aufgetragen werden.

Die Auswahl eines Studierenden in der Grafik führt zur in 4.5.2 gezeigten Einzelansicht für diesen Studierenden.

[19] nach Hochschulsemester 15 ist nur noch sehr wenig Bewegung in den Daten

4.5.2. Studentenansicht

Die Studentenansicht dient der Darstellung aller Daten über einen Studenten in einer aufbereiteten Form, die für Menschen einfacher verständlich ist als die direkte Interaktion mit der Datenbank. Die hier im Beispiel gezeigten Daten entsprechen keinem realen Studenten, sondern wurden für die Veröffentlichung generiert.



Id	Kohorte	Studiengang	IsGraduated	Studiendauer	GPA
1027	2010WS	B-TI	M	4.0	8.199999999999999

Titel	avg	Note	Note	Note	Note
PR1	6.96	10			
PRP1	5.67	6			
GE1	6.51	2	8		
MG	5.67	9			
GT	7.65	12			
GS	9.45	8			
AF	6.86	7			
PR2	7.2	3	9		
GE2	8.02	8			
GSP	6.7	14			
BS	7.78	6			
SE1	9.07	7			
AD	8.24	7			
DB	8.13	6			
RN	7.23	6			

Abbildung 4.4.: Einzelansicht eines Studenten

Zu sehen sind die Daten des Studenten mit der ID 1027. Beginnend mit den Stammdaten in der Kopfzeile (ID, Kohorte, Studiengang, Graduationsstatus, Studiendauer und Notendurch-

schnitt). Darauffolgend die vom Studenten besuchten Kurse mit den Daten Fachkürzel (Titel), Durchschnittsnote der Prüfung in dem Semester, in dem der Studierende die Prüfung bestanden hat, oder im Semester des letzten Versuches (falls die Prüfung noch nicht bestanden wurde). Und maximal vier Noten für die Prüfung (entsprechend drei schriftlichen und einem mündlichen Versuch). Der Hovertext über den Noten ist das Semester der Prüfung.

5. Evaluation

In der Evaluation wird auf die Ergebnisse des Dataminings eingegangen und diese bewertet. Grob kann dieses Kapitel in induktive und deskriptive Bereiche unterteilt werden. Die ersten beiden Sektionen greifen diese im Fazit der Analyse (Kapitel 3.5) wieder auf und beantworten diese. Dies sind zum einen die Frage nach der Prognosefähigkeit von Daten des ersten Studienjahres (Studienerfolgsvorhersage (Abschlüsse) in 5.1), zum anderen die Frage, ob eine Prognose auch für die Studiendauer von erfolglosen Studenten möglich ist (Studienerfolgsvorhersage, langes erfolgloses Studieren in 5.2) und zuletzt die Frage, in welcher Art retrospektiv analysierte Daten dem einzelnen Studenten Aufschluss über seinen individuellen Studienfortschritt geben können.

Der deskriptive Teil ist zusammengefasst im Kapitel Deskriptive Ergebnisse (5.3) und beinhaltet primär beschreibende Statistiken. Einleitend wird auf die Fehlversuchsquoten unterschiedlicher Modulprüfungen eingegangen (Kapitel 5.3.2), um dann auf die durchschnittlich in einem Modul erreichten Notenpunke einzugehen (Kapitel 5.3.3). Es folgt eine Betrachtung der von der Hochschule erhobenen Statistiken zur Anzahl der Bewerber (5.3.5) und zur Art der Hochschulzugangsberechtigung (Kapitel 5.3.4). Abschließend wird darauf eingegangen, welche Prüfungen am häufigsten nicht im für sie vorgesehenem Semester geschrieben werden (Kapitel 5.3.6).

Schlussendlich werden die Ergebnisse zusammenfassend noch einmal dargestellt (Fazit: 5.5).

5.1. Studienerfolgsvorhersage (Abschlüsse)

Im Rahmen dieser Arbeit wurde untersucht, inwiefern eine Prognose des Studienerfolges aus den Daten des ersten Studienjahres[1] möglich ist. Stellt man den Studienerfolg als Abschluss/kein Abschluss dar, kann diese Prognose als binäres Klassifikationsproblem

[1] Die Beschränkung auf Daten des ersten Studienjahres spiegelt die in diesem ersten Studienjahr durch die Prüfungsordnung zwingend vorgeschriebene Studienfachberatung wieder.

(vgl. Russell und Norvig, 2010, S. 696) betrachtet und als solches gelöst werden. Folgende Methoden wurden zur Prognoseerstellung eingesetzt: SVM, Naive Bayes, Entscheidungsbäume, k-nächste Nachbarn, ein mit Auto MLP trainiertes neuronales Netz sowie logistische Regression (siehe zu diesen Methoden auch das Kapitel 2.2 (Statistische Verfahren)).

5.1.1. Datenauswahl

Die unterschiedlichen Methoden wurden jeweils mit dem gleichen Datenbestand getestet. Hierzu wurden zuerst alle Datensätze, die zuvor oder im Wintersemester 2010 immatrikulierten Studenten gehören, ausgewählt und durch eine geschichtete Zufallsstichprobe[2] in zwei gleich große Hälften geteilt. Mit der ersten Hälfte wurden die Klassifikationsmodelle trainiert, die zweite wurde zur Performance Analyse verwendet. Tabelle 5.1 zeigt die Ergebnisse dieser Analyse. Durch die Wahl des Wintersemesters 2010 wird ein Abstand von 10 Semestern zum Ende der vorhandenen Notendaten eingehalten, was 166% der Regelstudienzeit entspricht und Verfälschungen durch Studenten, die noch studieren, vermeidet.

5.1.2. Attributsauswahl

Für das Training und die Prognose wurden bei allen Verfahren die gleichen Attribute verwendet. Dies sind die Klausurnoten der Kurse Programmieren 1, Programmieren 2 und Grundlagen Technische Informatik, die Note der Laborprüfung, die zu Programmieren 1 gehört, sowie die Gesamtanzahl der vom Studenten erreichten CPs im ersten und zweiten Semester. Diese Auswahl beruht initial auf Veröffentlichungen, die den Prognosecharakter technisch/-mathematischer Fächer zeigen (Golding und Donaldson, 2006; Konvalina u. a., 1983) und den persönlichen Erfahrungen des Autors. Während des Hyperparameter-Tunings wurden systematisch auch andere Attribute als Prediktoren getestet. Dies hat nicht zu einer Verbesserung der Prognosemodelle geführt. Im Datensatz fehlende Noten (also solche, bei denen noch kein 1. Versuch erfolgt ist) wurden vor dem Training auf 0 erreichte Leistungspunkte gesetzt, um Methoden nutzen zu können, die nicht auf Datensätze mit fehlenden Attributen angewendet werden können. Die korrektesten Prognosen wurden mit der Kombination Programmieren 1, Praktikum Programmieren 1, Programmieren 2 und gesamte CPs erstes und zweites Semester erreicht.

[2] geschichtet auf dem zu prognostizierendem binären Attribut: Abschluss/kein Abschluss

5.1.3. Ergebnisse

Tabelle (5.1) zeigt die verwendeten Methoden mit der jeweils erreichten Prognosegenauigkeit. Bei allen Verfahren außer Entscheidungsbäume, Lineare Regression und KNN handelt es sich um nicht deterministische Verfahren, sodass hier eine gewisse Abhängigkeit von zufälligen Startbedingungen gegeben ist. Eine nähere Beschreibung der einzelnen Verfahren findet sich in Kapitel 2.2.

Methode	Genauigkeit
KNN	86,53%
Linear Regression	83,84%
SVM	83,16%
Bayes (Kernel)	81,48%
Neuronal Net (Auto MLP)	80,81%
Entscheidungsbaum	80,47%
Logistic Regression	80,13%

Tabelle 5.1.: Prognose Studienerfolg, Abschluss

Zu sehen ist, dass die Ergebnisse durchweg zwischen 80% und 87% sind und relativ nahe beieinander liegen (+/- 7%). Diese Ergebnisse lagen vor dem Hyperparameter-Tuning zwischen 2% (k-nearest neighbors) und 15% (Neuronal Net) niedriger. Sie sind unter Modifikation der Lern-bzw. Validierungsmenge (durch Änderung der Startpunkte für den verwendeten Pseudozufallsgenerator) von Datensätzen relativ konstant (+/- 2%). Tabelle 5.2 zeigt die Prognose, die durch ein k-nächstes-Nachbar-Modell erstellt wurde, im Detail.

	true false	true true	class precision
pred. false	151	17	89,88%
pred. true	23	106	82,17%
class recall	86,76%	86,18%	

Tabelle 5.2.: Prognose Studienerfolg k-nearest neighbors, Details

Zu sehen sind jeweils die Anzahl der Prognosen und zu welchem Prozentsatz sie korrekt waren sowie wie viel Prozent einer Klasse (true=Abschluss, false=kein Abschluss) korrekt erkannt wurden.

Das Ergebnis für jeden einzelnen Studenten ist eine Klassifikation (Abschluss/kein Abschluss) und der Konfidenzwert dieser Aussage (prozentual). Die hier jeweils falsch klassifizierten Datensätze wurden manuell analysiert um Rückschlüsse auf den Grund von Fehl-

klassifikationen zu erlauben. Hierbei konnten 2 wesentliche Gruppen von Klassifikationsfehlern identifiziert werden. Zum einen solche Datensätze, bei denen die Prognosekonfidenz niedrig war ($< 70\%$), die also an den Grenzen der Klasse lagen. Zum anderen solche Datensätze mit hoher Konfidenz ($> 90\%$), bei denen offensichtlich ein zur Hochschule externes Ereignis zum Erfolg/Misserfolg geführt hat. Hierdurch wird hier auch die höhere Präzision der Klassifikation 'Kein Abschluss' erklärt. Es gibt mehr hochschulexterne Ereignisse, die einen erfolgsversprechenden Studenten davon abhalten, einen Abschluss zu erhalten als umgekehrt. So sind z.B. der Wechsel des Studienortes oder Fachs ein Grund, der zu keinem Abschluss führen kann. Umgekehrt können zum Beispiel plötzliche Krankheit oder soziale Verpflichtungen einen eingeschriebenen Studenten im ersten Studienjahr davon abhalten, Leistungen zu erbringen und hierdurch zu einer Fehlklassifikation führen. Da die Zeit zwischen Bewerbung und Studienbeginn kurz ($<$4 Monate) und das Durchschnittsalter der Studenten niedrig (23 Jahre) ist, ist die Prävalenz in den vorhandenen Daten niedrig[3].

5.1.4. Plausibilitätskontrolle

Zur Plausibilitätskontrolle der erstellten Modelle und um eine eventuelle Überanpassung zu erkennen, wurden atypische Semester generiert und mit den erstellten Modellen bewertet. Hier wurden keine gravierenden Probleme aufgedeckt. Die von den Modellen gelieferten Wahrscheinlichkeiten entsprachen den Erwartungen. Als atypische Semester wurden solche generiert, die in der Verteilung der Notenpunkte sehr stark von den retrospektiv bekannten abweichen. Beispielsweise also viele Punkte in Programmieren 2 und sehr wenige in Programmieren 1 oder wenige Punkte in Programmieren 1 und 2 dafür aber viele im Praktikum Programmieren 1.

5.2. Studienerfolgsvorhersage, langes erfolgloses Studieren

Es wurde eine Gruppe von Studierenden untersucht, die sich durch ihr langes erfolgloses Studium definiert, und ob diese an den Daten des ersten Studienjahres erkannt werden kann. Motivierendes Ziel ist an dieser Stelle, positiv zu intervenieren und durch Angebote der Studienförderung (z.B. Tutorien, Beratung) auf die Studierenden einzuwirken und sie so entweder zu einem erfolgreichen Abschluss zu führen oder alternativ ein Überdenken der Studienmotivation für den konkreten Studiengang zu erreichen.

[3] weniger als 10 im betrachteten Zeitraum

5.2.1. Zielgruppendefinition

Als Definition eines langen erfolglosen Studiums wurde für diese Arbeit eine Studiendauer von mehr als 6 Semestern und ein ohne Abschluss beendetes Studium gewählt. 6 Semester entspricht der Regelstudienzeit für den Studiengang Technische Informatik. Die Mediandauer eines erfolglosen Studiums beträgt 4 Semester und der Durchschnitt 6 Semester.

5.2.2. Quantität

Im Datensatz trifft die in 5.2.1 gezeigte Definition auf 102 Studierende, die vor/im Wintersemester 2009 immatrikuliert wurden, zu. Das Ende des Untersuchungszeitraumes wurde gewählt, da so mindestens 11 Semester Abstand zum letzten erfassten Semester 2015 bestehen, was $Q_{0,75}$ dem oberen Quantil der Studiendauer erfolgreicher Studenten entspricht und diese von der Modellbildung ausschließen soll.

5.2.3. Vorgehen

Initial wurden exploratorisch die schon in 5.1 verwendeten Verfahren und Attribute eingesetzt, um ein Prognosemodell zu entwickeln. Hier konnten im Gegensatz zur Prognose der Abschlüsse nur Prognosegenauigkeiten zwischen 30 und 40 Prozent erreicht werden. Im Folgenden wurden weitere Verfahren und Attribute evaluiert.

5.2.3.1. Erfolgswahrscheinlichkeit

Evaluiert wurde, ob es eine Partition der Daten unter Nutzung der in 5.1 berechneten Wahrscheinlichkeit eines Abschlusses gibt (also etwa alle Studenten mit Erfolgswahrscheinlichkeit zwischen 30% und 60%), auf der eine höhere Genauigkeit als 30% erreicht werden kann. Eine solche wurde nicht gefunden, ein Ergebnis, das in der Reflexion nicht überraschend ist, da die Attribute der Erfolgswahrscheinlichkeit schon evaluiert wurden. Und so also keine neue Information, sondern nur eine weitere Abstraktion eingeführt wurde.

5.2.3.2. Verhältnis versuchte/erfolgreiche Credit Points

Aufbauend auf der Annahme, dass erfolglose Langzeitstudierende ihren Fortschritt bzw. ihre eigene Leistungsfähigkeit nicht korrekt einschätzen, wurde das Verhältnis von versuchten zu

erlangten CPs als Attribut in der Modellbildung verwendet. Hierbei hat sich gezeigt, dass dieser Quotient keine bessere Einschätzung der Studiendauer erlaubt als die bereits evaluierten technischen Kurse.

5.2.3.3. Auflockerung der Regeln

Die Analyse-Ergebnisse verbessern sich, wenn die Analyse nach einem höheren Hochschulsemester durchgeführt wird, also etwas nach dem vierten Hochschulsemester und nicht nach dem zweiten. Auch dann wird jedoch keine Genauigkeit erreicht, die in einem für die Prognose sinnvollen Semester besser als 40% ist.

5.2.4. Ergebnis

In Ermangelung eines zuverlässigen Prognosemodells mit Daten aus dem ersten Studienjahr wurde dann untersucht, wie die Studienentwicklung allgemein voranschreitet. Hierzu wurden die Studenten nach dem Prozentsatz der erreichten Credit Points pro Semester in 3 Gruppen eingeteilt. Diese Gruppen umfassen nicht alle Studierenden, da die Abgrenzungen aufgrund der Credit Points nicht genau getroffen werden können. Diese Ungenauigkeit wird durch die Puffer zwischen den Gruppen abgebildet.

Bezeichnung	CPs %	CPs abs. %	Studierende %	echte Langzeitstudenten %
Vollzeitstudenten	$>$70%	$>$21	30,87%	2,75%
Teilzeitstudenten	40%-60%	$>$12 CPs $<$ 18	20,47%	29,36%
pot. Langzeitstudenten	$<=$20%	$<=$6	21,76%	25,69%

Tabelle 5.3.: Gruppeneinteilung Studenten nach erreichten CPs pro Semester

Gezeigt ist hier die Klassifizierung der Studenten, die Credit Points die prozentual und absolut pro Semester die Grenzen der Klassen darstellen, und der prozentuale Anteil der Studenten in der vorhandenen Datenbasis, die den einzelnen Gruppen zugeordnet werden können, sowie der Prozentsatz der echten Langzeitstudenten, der jeweils in einer Gruppe vorhanden ist. Insgesamt können so ca. 73% der Studenten einer Gruppe zugeordnet werden. Die Treffergenauigkeit für Langzeitstudenten ist hier nicht höher als bei den komplexeren Modellen.

5.2.5. Bewertung

Hier wird hypothetisiert, dass sich erfolglose Langzeitstudenten im erfassten Notenbild nicht deutlich von den „nur“ erfolglosen Studierenden unterscheiden. Mögliche Erklärungen sind, dass die Langzeitstudierenden ihren Studienfortschritt falsch einschätzen oder aus anderen Gründen[4] am erfolglosen Studium festhalten. Zur Zeit liegen keine Erkenntnisse vor, die die Beantwortung dieser Frage zulassen. Im Ausblick dieser Arbeit (Kapitel 6.2.2) wird dieses Thema noch einmal aufgegriffen, da die Erhebung qualitativer Daten zur Beantwortung der aufgeworfenen Fragen notwendig erscheint.

5.3. Deskriptive Ergebnisse

Der deskriptive Teil der Evaluation beschreibt einzelne Statistiken, die sich aus der Datenbasis ergeben. Zu Beginn wird eine später im Kapitel genutzte Metrik (Creditpoints pro Student einer Kohorte) eingeführt (Kapitel 5.3.1). Fachlich einleitend wird auf die Fehlversuchsquoten unterschiedlicher Modulprüfungen eingegangen (Kapitel 5.3.2), um dann auf die durchschnittlich in einem Modul erreichten Notenpunke einzugehen (Kapitel 5.3.3). Es folgt eine Betrachtung der von der Hochschule erhobenen Statistiken zur Anzahl der Bewerber (5.3.5) und zur Art der Hochschulzugansberechtigung (Kapitel 5.3.4). Abschließend wird darauf eingegangen, welche Prüfungen am häufigsten nicht im für sie vorgesehenen Semester geschrieben werden (Kapitel 5.3.6).

5.3.1. Metrik: Credit Points pro Student einer Kohorte

Im Folgenden wird an einigen Stelle eine Metrik verwendet, die die kumulierten durchschnittlichen Credit Points pro Student unterschiedlicher Kohorten vergleicht. Diese wird hier erläutert. Die Metrik ist definiert für eine Kohorte X und ein Hochschulsemester Y als

$$\frac{\text{Kumulative Summe CP durch } X \text{ erreicht bis Hochschulsemester } Y}{\text{Anzahl Studenten in } X} \tag{5.1}$$

Beispielsweise hat die Kohorte $2005WS$ mit 61 Studierenden im 3. Hochschulsemester $2652,5$ Credit Points erbracht. Für die Metrik ergibt sich also ein Wert von $43,48 = \frac{2652,5}{61}$.

Ein Semester ist also umso erläutert. umso höher die Metrik für dieses Semester ist.

[4] z.B. Selbstbild als Informatiker, familiärer/beruflicher Druck

5.3.2. Fehlversuchsquoten

Die Fehlversuchsquoten[5] im Erstversuch aller benoteten Prüfungen der Studierenden, die bis und inklusive Wintersemester 2010 immatrikuliert wurden (entspricht der in den Abschnitten 5.1 und 5.2; Studienerfolg betrachteten Gruppe), wurden untersucht, um Auffälligkeiten exploratorisch herauszuarbeiten. Die Prüfungen beider Prüfungsordnungen (2004/2008) werden hier in einer Tabelle geführt. Neben der exploratorischen Analyse wurden gezielt die Prüfungen des ersten Studienjahres untersucht, die Programmieren als Schwerpunkt haben (PR1, PRP1, PR2), dies entspricht der Gruppe von Prüfungen, die in einem linearen Regressionsmodell am stärksten mit dem Studienerfolg gemessen am Abschluss korrelieren. Die vollständige Liste der Fehlversuchsquoten findet sich im Anhang (A.1.2). Im Folgenden werden auffällige Effekte besprochen.

Generell zeigen die ermittelten Daten einen Trend zu höheren Fehlversuchsquoten bei Prüfungen in niedrigeren Fachsemestern. Dieser Trend wird hier darauf zurückgeführt, dass Prüfungen der unteren (<3) Fachsemester relativ häufiger von Studenten, die schlussendlich keinen Studienerfolg haben, besucht werden als solche höherer Fachsemester. Umgangsprachlich wird mit dem Praktikum Programmieren 1 (Fehlversuchsquote 52,3%) “herausgeprüft“ und nicht mit der Bachelorarbeit (Durchfallquote: 5,6%). Dies entspricht auch der in verschiedenen Gremien geäußerten Zielsetzung der Fakultät, den Studienerfolg in Form des Abschlusses primär in der ersten Hälfte des Studiums (Semester eins, zwei und drei) zu bestimmen.

5.3.2.1. Gruppen

Tabelle 5.4 teilt die Klausuren in drei Gruppen geordnet nach der Fehlversuchsquote ein. Die gewählte Aufteilung teilt die Prüfungsleistungen grob in die drei Gruppen Schwierig/Normal/-Leicht.

Gruppe	Fehlversuchsquote
1	$>=$25%
2	$>=$10%
3	$<$10%

Tabelle 5.4.: Fehlversuchsquoten, Gruppen

In Gruppe 1 finden sich die Prüfungen, an der mehr als ein Viertel der Studierenden im Erstversuch scheitert, dies sind Klausuren der ersten drei Semester (erste Studienhälfte). Es

[5] Ein Fehlversuch ist die Teilnahme an einer Prüfung, die mit der Note *nicht ausreichend*, entspricht weniger als 5 Notenpunkten bewertet wurde.

handelt sich um technisch-mathematische[6] Module. Gruppe 2 besteht aus den Prüfungen, an denen 10% und mehr, aber weniger als 25% der Studierenden im Erstversuch scheitern. Der Großteil aller Prüfungen befindet sich hier. In Gruppe 3 sind die Prüfungen, durch die wenige Studenten durchfallen. Sie setzt sich primär zusammen aus Betriebswirtschaftsmodulen, Wahlpflichtkursen (GW und WP) und der Bachelorarbeit.

5.3.2.2. Auffälligkeiten bei einzelnen Ergebnissen

Auffällig ist, dass Digitaltechnik 2 sich in Gruppe 3 befindet, ein technisch/mathematisch orientiertes Modul der Prüfungsordnung 2004, das in seiner Ausprägung als Digitaltechnik in der Prüfungsordnung 2008 eine Durchfallquote von 24% aufweist. Eine Theorie zur höheren Fehlversuchsquote ist, dass mit der Reduzierung der Credit Points für Digitaltechnik (PO2004 Digitaltechnik 1 und 2 insgesamt 10 Credit Points, PO2008 Digitaltechnik 6 Credit Points) nicht analog die Prüfungsinhalte angepasst wurden. Diese Hypothese könnte in einer späteren Arbeit durch das Vergleichen der jeweiligen Prüfungsinhalte validiert werden.

Die Prüfung aus der zweiten Studienhälfte mit der höchsten Fehlversuchsquote ist die Prüfung zum Modul Software Engineering und Anwendungen aus der Prüfungsordnung 2008[7]. Dies bestätigen Berichte von sowohl Professoren als auch Studenten, die diese Modulprüfung als mit hohen Anforderungen verbunden bezeichnen bzw. wahrnehmen. Ein Grund dafür ist, dass das mit 10 Credit Points bewertete Modul die Inhalte aus drei Vorlesungen (Software Engineering, Prozesslenkung und System-/Echtzeitprogrammierung) verbindet. Die Inhalte aller drei Vorlesungen werden in einem zum Modul gehörenden Praktikum verbunden und in Form einer mündlichen Prüfungsleistung kontrolliert. Vergleicht man das Modul mit den drei Vorgängermodulen aus der Prüfungsordnung 2004, fällt auf, dass die kombinierte Prüfung höhere Fehlversuchsquoten (22,9% zu zwischen 9% und 12% für die Vorläufer) hat, aber die Durchschnittsnote marginal besser ist (11,2 im Gegensatz zu 10,7).

5.3.2.3. Stetigkeit der Quoten

Die Quoten, zu denen Prüfungen bestanden bzw. nicht bestanden werden, weisen zum Teil erhebliche Varianzen für unterschiedliche Semester/Kohorten auf, Tabelle 5.5 zeigt beispielhaft die Varianz für das Praktikum Programmieren 1.

[6] Mathe, Programmieren, Digitaltechnik und zugehörige Laborprüfungen

[7] in der Tabelle mit *Software Engineering II/Prozesslenkung/System- und Echtz* bezeichnet

Kohorte	Fachkrz.	Fehlversuchsquote
2010WS	PRP1	43.18%
2010SS	PRP1	66,00%
2009WS	PRP1	59,61%
2009SS	PRP1	54,76%
2008WS	PRP1	46,66%
2008SS	PRP1	55,00%
2007WS	PRP1	46,93%
2007SS	PRP1	51,35%
2006WS	PRP1	57,14%
2006SS	PRP1	55,17%
2005WS	PRP1	52,72%
2005SS	PRP1	55,10%
2004WS	PRP1	35,00%

Tabelle 5.5.: Fehlversuchsquote Praktikum Programmieren 1 nach Kohorten

Gezeigt wird die Kohorte, das Fachkürzel (hier immer PRP1 für Praktikum Programmieren 1) und die Quote, zu der der Erstversuch von Studenten aus der betreffenden Kohorte nicht bestanden wurde. Diese Quote bewegt sich hier zwischen 35% im Wintersemester 2004 und 66% im Sommersemester 2010, Median 55% und Durchschnitt 52%. Die Varianz und Verteilung ist ähnlich, wenn nicht nach Kohorten, sondern nach dem Semester der Prüfung gruppiert wird. Auf Grundlage der vorhandenen Daten kann hier nicht abschließend geklärt werden, was die Ursache der Varianz ist. Hypothetisch kommen neben unterschiedlich begabten Studenten unter anderem auch unterschiedliche Lehrende als Quelle der Varianz in Frage.

5.3.2.4. Ausblick

Fragen, die sich aus dem hier behandelten Themenkomplex der Erfolgsquoten ergeben, sind der Abgleich der statistischen Schwierigkeit einer Prüfung in Form der Quote von nicht erfolgreichen Prüfungsversuchen mit der durch die Studierenden wahrgenommenen Schwierigkeit einer Prüfung. Eine weitere offene Frage ist die Abhängigkeit der Erfolgswahrscheinlichkeit vom Lehrenden, durch Anreicherung der Daten mit den aus archivierten Stundenplänen ersichtlichen Lehrenden kann diese beantwortet werden.

5.3.3. Durchschnittliche Prüfungsnoten

Tabelle A.1 in Anhang A.1.1 zeigt die durchschnittlichen Noten aller Modulprüfungen, die von Studenten abgelegt wurden, die vor und im Wintersemester 2010 immatrikuliert wurden und bestanden worden sind (Notenpunkte > 4). Gezeigt wird der Name des Moduls, das Fachsemester des Moduls und die Durchschnittsnote. Nicht gezeigt werden Wahlpflichtkurse (GW und WP), da diese keinem Semester zugeordnet werden können. Im Folgenden wird im Allgemeinen auf die Durchschnittsnoten eingegangen und danach der Unterschied zwischen mündlichen und schriftlichen Prüfungen am Beispiel zweier Module erläutert.

5.3.3.1. Allgemeines

Allgemein zeigt die Tabelle, dass die Prüfungsnoten in den höheren Semestern im Durchschnitt besser sind als in den unteren. Ein Effekt, der analog auch schon für die Fehlversuchsquoten (5.3.2) festgestellt wurde. Dieser Effekt wird hier auch (analog zu 5.3.2) darauf zurückgeführt, dass an den Prüfungen aus höheren Fachsemestern mehr erfolgreiche Studenten teilnehmen als an Prüfungen aus niedrigeren Semestern. Tabelle 5.6 zeigt die 5 Prüfungen mit dem besten Notendurchschnitt.

Prüfung	Fachsem.	Note
Bachelorarbeit	6	12,4708
Software Engineering 2	4	11,2429
SE2/PL/SY	4	11,0598
Betriebswirtschaft	5	10,8599
Betriebswirtschaftslehre 2	4	10,7431

Tabelle 5.6.: Durchschnittliche Noten aller Modulprüfungen, Top 5

Gezeigt werden der Name der Modulprüfung, ihr Fachsemester und die durchschnittliche Note aller bestandenen Prüfungen. Zwei der fünf gezeigten Prüfungen sind mündliche Leistungskontrollen (Software Engineering 2 (PO2004) und SE2/PL/SY (PO2008)), zwei sind nicht technisch-mathematische Module (Betriebswirtschaftslehre 2 (PO2004) und Betriebswirtschaft (PO2008)), die durchschnittlich besten Prüfungsergebnisse werden in der Bachelorarbeit erreicht.

5.3.3.2. Mündliche Prüfungen

Anhand der Prüfung für die Modul Automaten und Formale Sprachen (AF) und Betriebssysteme (BS) wurde betrachtet, ob mündliche Prüfungen bessere Noten erbringen als Klausu-

ren (ein Effekt, der von Professoren berichtet wird). AF und BS stellen insofern eine günstige Möglichkeit dar, diesen Effekt zu analysieren, weil mit dem Wechsel der Prüfungsordnung von 2004 auf 2008 aus einer mündlichen Prüfung eine Klausur(AF) bzw. aus einer Klausur eine mündliche Prüfung (BS) wurde. Die Inhalte haben sich jeweils laut Modulhandbuch nicht wesentlich geändert. Tabelle 5.7 zeigt die Durchschnittsnoten von AF und BS.

Modul	Mündliche Prüfung	Klausur
AF	9,95	9,23
BS	10,05	10,37

Tabelle 5.7.: Vergleich mündliche/schriftliche Prüfungen

Gezeigt ist das Modul und jeweils die durchschnittliche Note für mündliche bzw. schriftliche Prüfungen. Während AF einen deutlichen Unterschied im Prüfungserfolg hin zu einer besseren Note für mündliche Prüfungen zeigt, ist dieser Effekt für BS umgekehrt und sehr viel schwächer ausgeprägt. Ein Grund für diesen Unterschied ist aus den vorhandenen Daten nicht zu ermitteln. Eine Hypothese zur Erklärung ist, dass weniger unterschiedliche Lehrende BS prüfen als AF.

5.3.4. Auswertung der C2-Statistik (Bewerbungserfolgsquote)

Aufbauend auf der C2-Statistik der Hochschule konnte ein signifikanter ($p = 0.00179$) negativer Zusammenhang zwischen dem Studienerfolg gemessen in CP[8] sowie dem Studienerfolg gemessen an der Abschlussquote eines Semester und der Bewerbungserfolgsquote gezeigt werden. Im Folgenden werden die C2-Statistik sowie diese Ergebnisse erläutert.

5.3.4.1. C2- (Bewerbungserfolgsquote) Statistik

Die C2-Statistik beschreibt Bewerbungen, Zulassungen, Immatrikulationen 1. Fachsemester und Ablehnungen (NC) pro Studiengang und Semester[9]. Aus diesen Daten können die Bewerbungserfolgsquote (Zulassungen/Bewerbungen) sowie die Annahmequote (Immatrikulationen/Zulassungen) berechnet werden (siehe auch Kapitel 3.2.2.4 für mehr Details zur C2-Statistik).

Anekdotisch berichtet und intuitiv angenommen wird ein Zusammenhang zwischen niedriger Bewerbungserfolgsquote und höherem Studienerfolg. Begründet wird dies durch eine

[8] gemessen an der Metrik CP pro Studierendem, siehe 5.3.1
[9] Die Statistik wird seit Sommersemester 2007 erhoben

größere Selektivität in der Aufnahme von Studierenden, die zu einer höheren Qualität derselben führt[10].

5.3.4.2. Ergebnis

Grafik 5.1 zeigt die Verteilung der durchschnittlichen CPs pro Student in den Semestern, inklusive der Regressionsgeraden im Verhältnis zur Bewerbungserfolgsquote, für die Daten vorliegen. Grafik 5.2 zeigt die Bewerbungserfolgsquote im Vergleich zum prozentualen Anteil der Studierenden einer Kohorte, die schließlich einen Abschluss erreicht haben.

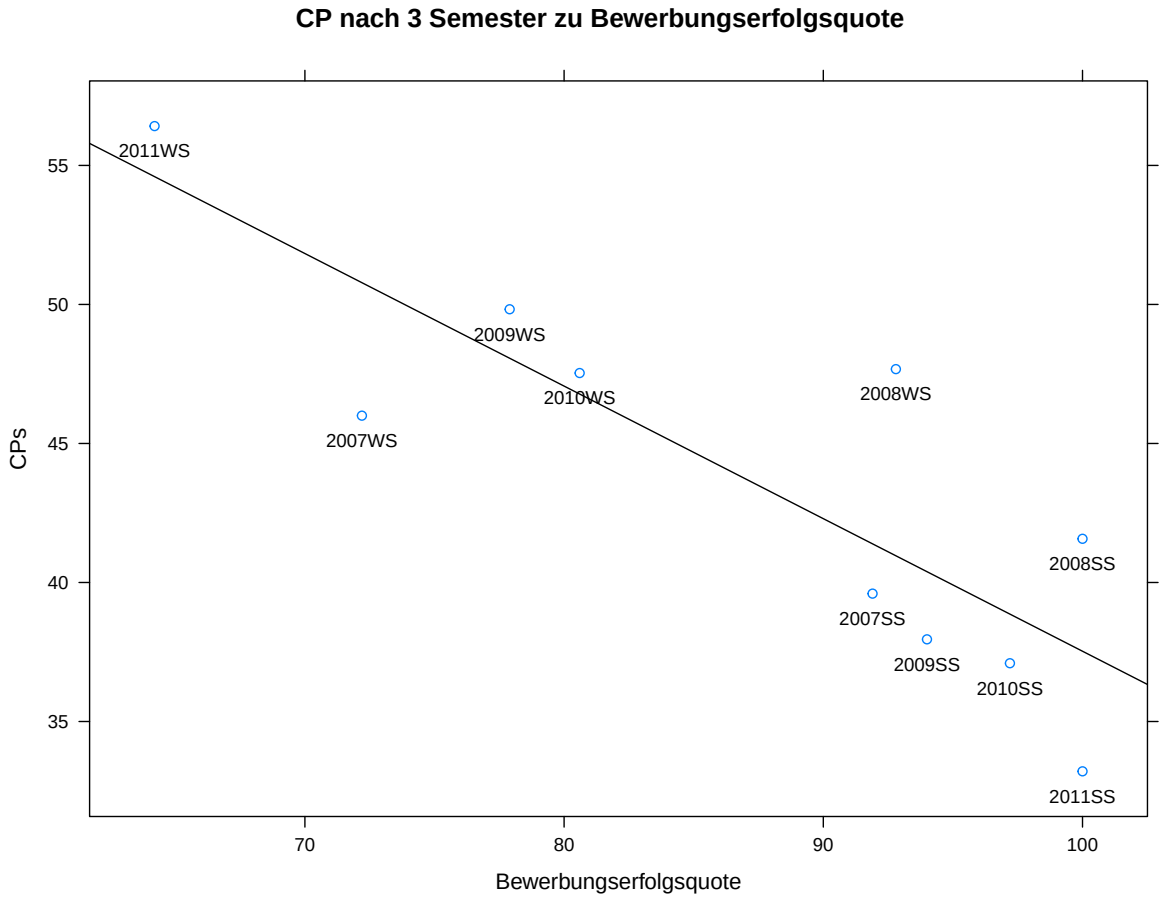


Abbildung 5.1.: Bewerbungserfolgsquote zu durchschnittlichen CPs pro Student nach 3 Semestern

[10] Selektionskriterium: Note der Hochschulzugangsberechtigung

Die grafische Darstellung (5.1) zeigt auf der y-Achse die durchschnittliche pro Student erreichten CPs nach 3 Semestern und auf x-Achse die Bewerbungserfolgsquote. Gut erkennbar ist zum einen ein deutlicher Unterschied zwischen Sommer und Wintersemester in der Bewerbungserfolgsquote, die in Wintersemestern regelmäßig 10-20 Prozentpunkte niedriger [11] ist als in Sommersemestern. Sowie hiermit zusammenhängend mit den durchschnittlich pro Student erreichten CPs nach 3 Semestern[12].

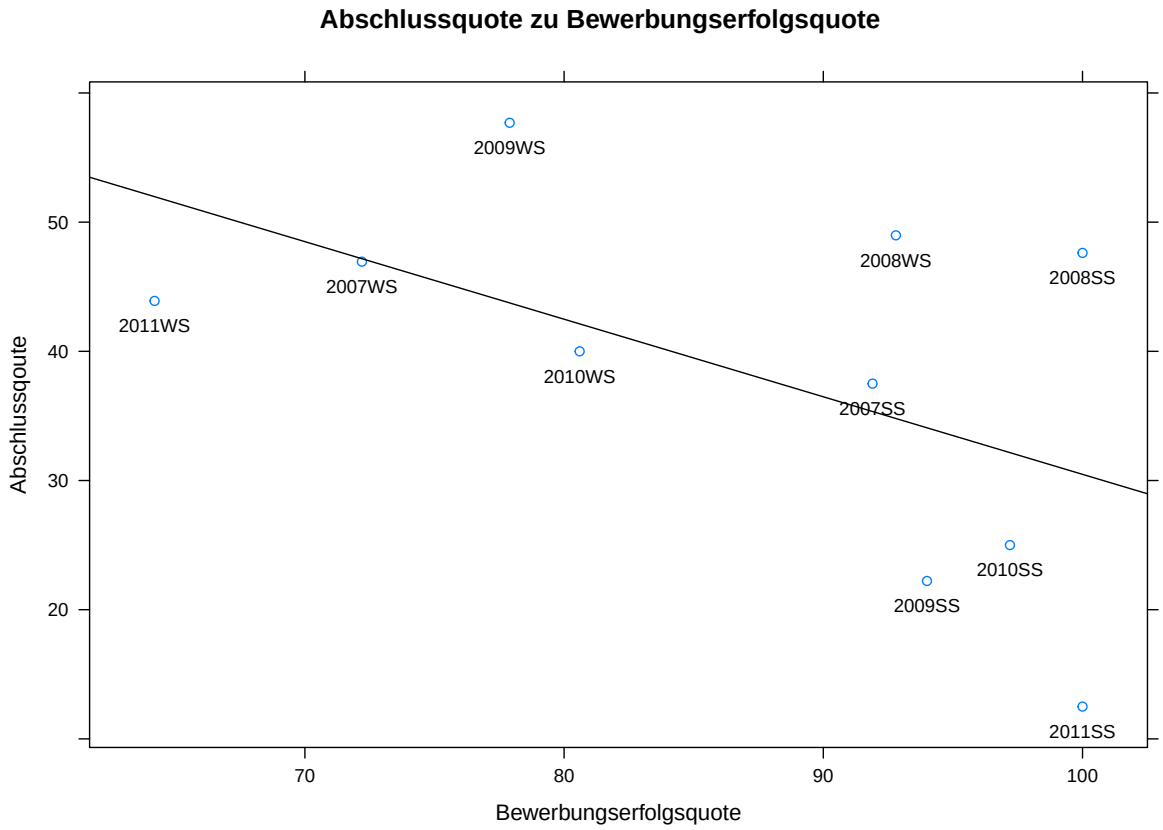


Abbildung 5.2.: Bewerbungserfolgsquote zur Abschlussquote in Prozent

Grafik 5.2 zeigt den Vergleich von Abschlussquote (y-Achse) eines Semesters mit Bewerbungserfolgsquote (x-Achse). Der oben zu erkennende Zusammenhang zwischen Bewerbungserfolgsquote und Studienerfolg ist hier nicht mehr so deutlich ausgeprägt und hat bei $p = 0.11$ auch nicht mehr die gleiche Signifikanz.

Zusammenfassend liegt ein Zusammenhang der Bewerbungserfolgsquote mit dem in

[11] d.h. weniger der zulässigen Bewerbungen wahren erfolgreich
[12] statistisch relevant bei $p = 0.00179$

Deutschland im August endenden Schuljahr[13] gedanklich nahe. Eine direkte Auswirkung auf den Studienerfolg gemessen an den CPs pro Student nach 3 Semestern besteht ($p = 0.00179$) lässt sich für den Studienerfolg gemessen an der Abschlussquote jedoch nicht mehr in der gleichen Deutlichkeit nachweisen ($p = 0.11$).

5.3.5. Auswertung der C3-Statistik (Art der Hochschulzugangsberechtigung)

Die C3-Statistik beschreibt die Vorbildung der neu immatrikulierten Studenten aufgeschlüsselt nach den entsprechenden Absätzen des Hamburger Hochschulgesetzes (siehe auch Kapitel 3.2.2.5 für mehr Details zur C3-Statistik). Hier konnte kein signifikanter Zusammenhang ($p = 0.5088$) zwischen dem Studienerfolg[14] eines Semesters und der Abiturientenquote gezeigt werden, Grafik 5.3 zeigt diese Verteilung.

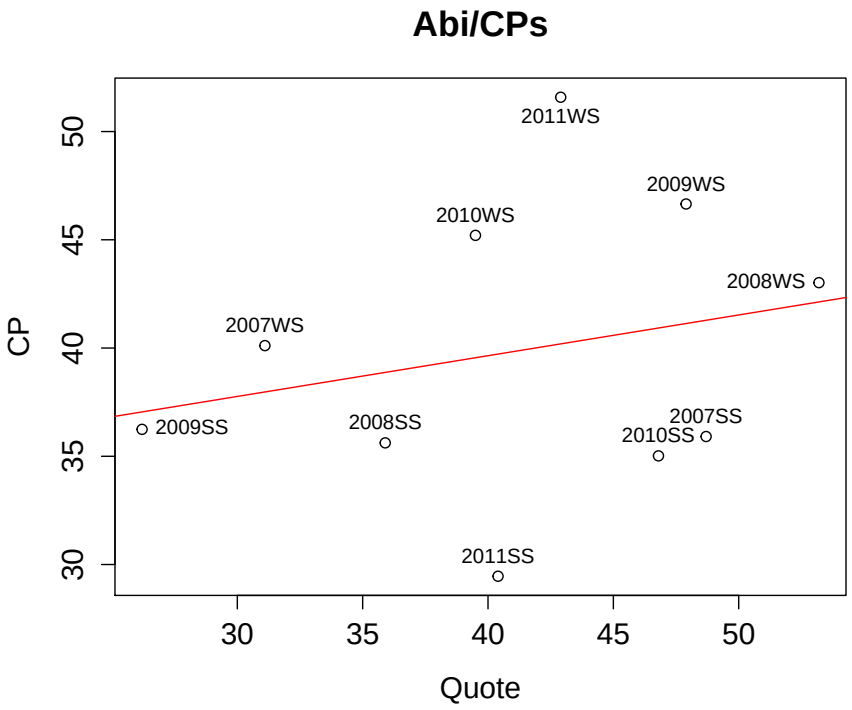


Abbildung 5.3.: Abiturientenquote zu durchschnittlichen CPs pro Student nach 3 Semestern

[13] und den dadurch mit dem Studium beginnenden Abiturienten
[14] gemessen an der Metrik CPs pro Studierendem, siehe 5.3.1

5.3.6. Aufgeschobene Prüfungen

Im Rahmen der Arbeit wurde untersucht, welche Prüfungen überdurchschnittlich oft von den Studierenden aufgeschoben wurden, also aus der gewählten Perspektive nicht zum durch die Prüfungsordnung vorgegebenen Zeitpunkt absolviert wurden. Hierzu wurden verschiedene Metriken definiert, um aufgeschobene Prüfungen zu erkennen. Es wird jeweils nur der erste Prüfungsversuch betrachtet.

- Hochschulsemester
- Credit Points
- Aufgegliedert nach Semestern

Im Folgenden werden die einzelnen Metriken beschrieben, ihre Ergebnisse gezeigt und abschließend eine Wertung getroffen.

5.3.6.1. Hochschulsemester

Definition

Eine intuitive Metrik, um zu erkennen, ob eine Prüfung aufgeschoben wurde, ist das *Fachsemester* der Prüfung mit dem *Hochschulsemester* des Studenten zu vergleichen. Wenn das Hochschulsemester $>$ dem Fachsemester ist, wurde die Prüfung nicht rechtzeitig abgelegt, also aufgeschoben.

Ergebnisse

Die Tabelle 5.8 zeigt die fünf am häufigsten aufgeschobenen Prüfungen. Tabelle 5.9 zeigt die gleiche Ansicht, beschränkt diese Ansicht auf die Prüfungen der ersten drei Semester und Tabelle 5.10 auf die Prüfungen des ersten Semesters.

Prüfung	Fachsem.	% aufgeschoben
MA3	3	82,76%
BA	6	80,29%
VS	5	57,52%
SE2PL	2	57,14%
AF	5	50,71%

Tabelle 5.8.: Aufgeschobene Prüfungen, nach Hochschulsemestern

Prüfung	Fachsem.	% aufgeschoben
MA	3	82,76%
AF	2	50,71%
SE1	3	47,57%
BS	3	46,80%
AD	3	46,02%

Tabelle 5.9.: Aufgeschobene Prüfungen der ersten 3 Semester, nach Hochschulsemestern

Prüfung	Fachsem.	% aufgeschoben
PR1	1	24,34%
MG	1	14,65%
GT	1	12,57%
GE1	1	12,34%
MA1	1	12,25%

Tabelle 5.10.: Aufgeschobene Prüfungen des ersten Semesters, nach Hochschulsemestern

Zu sehen ist jeweils das Fachkürzel des Moduls, sein Fachsemester und der prozentuale Anteil von Prüfungen, die aufgeschoben wurden.

Bewertung

Die so erfassten Daten berücksichtigen nicht, dass ein signifikanter Anteil der Studierenden neben dem Studium arbeitet (40%-50%). Diese Erwerbstätigkeit füllt, wenn vorhanden, auch regelmäßig einen Tag oder mehr ($>$ 8 Stunden) aus (vgl. Meisel, 2005; EQA, 2009; Meisel, 2014). Diese einfache Metrik spiegelt im Zusammenspiel mit der unterschiedlich geplanten Studiendauer der Studierenden auch nicht zwingend die Wahrnehmung dieser wider[15]. Die Metrik spiegelt auch nicht wider, ob eine Prüfung geschoben wird, weil sie vom Studenten als schwierig empfunden wird oder aus anderen Gründen. Ein anderer Grund für das Aufschieben von Prüfungen ist, dass der Student in einem vorherigen Semester eine Prüfung nicht bestanden hat und erst diese nachholt. In Tabelle 5.10 ist deutlich zu sehen, dass im ersten Semester Programmieren 1 die am häufigsten geschobene Prüfung ist, dies wirkt sich zwangsläufig auf die folgenden Semester aus.

[15] Der Studierende, der für sich beschlossen hat, in Teilzeit zu studieren, wird es nicht als Verschieben empfinden, eine Prüfung des 2. Fachsemesters in seinem 4. Hochschulsemester abzulegen.

5.3.6.2. Credit Points

Die Metrik nach Credit Points (im Folgenden CP) versucht, einige der oben genannten Kritikpunkte zu beheben.

Definition

Eine Prüfung gilt als verschoben, wenn, bevor sie geschrieben wurde, mehr als Fachsemester[16] $*$ 30 CP erreicht wurden. Ein Beispiel: Eine Prüfung für das zweite Semester gilt als verschoben, wenn zum Prüfungszeitpunkt mehr als 60 CP erbracht waren. Hier treten also einige der oben genannten Probleme nicht mehr auf, da die individuelle Geschwindigkeit beachtet wird.

Ergebnisse

Prüfung	Fachsem.	% aufgeschoben
MA3	3	26,21%
MA2	2	11,76%
AF	2	6,71%
PR2	2	6,04%
AA	1	6,02%

Tabelle 5.11.: Aufgeschobene Prüfungen, nach Credit Points

Hier wird nur eine Tabelle verschoben, da im Unterschied zu 5.3.6.1 die Top 5 verschobenen Prüfungen alle in der ersten Studienhälfte liegen. Die Metrik erlaubt nur eine Modellierung für die ersten fünf Semester und misst bis zu einem gewissen Grad, wie genau an der Prüfungsordnung entlang studiert wird. Also ob die Prüfungen in deutlich anderer Reihenfolge absolviert werden, als in der Prüfungsordnung vorgesehen. Da nur solche Klausuren als verschoben betrachtet werden, die nach dem Zeitpunkt erbracht werden, zu dem sie laut Prüfungsordnung erbracht werden sollten[17].

5.3.6.3. Geschobene Prüfungen nach Semestern

Es wurde im Rahmen dieser Arbeit auch untersucht, wie die Verteilung von geschobenen Prüfungen auf die einzelnen Semester ist.

[16] der Prüfung
[17] gerechnet in CP

Definition

Wie viel Prozent der Prüfungen wurden pro Semester geschoben. Also in welchem Fachsemester wurden wie viel Prozent der Erstversuche verschoben absolviert. Tabelle 5.12 zeigt diese Daten für die Hochschulsemestermetrik und Tabelle 5.13 für die Credit-Points-Metrik.

Ergebnisse

Semester	% aufgeschoben
1	15,62%
2	31,01%
3	45,78%
4	48,10%
5	59,80%
6	58,72%

Tabelle 5.12.: Geschobene Prüfungen pro Semester, Hochschulsemestermetrik

Semester	% aufgeschoben
1	1,77%
2	3,37%
3	5,19%
4	6,02%
5	1,11%

Tabelle 5.13.: Geschobene Prüfungen pro Semester, Credit-Points-Metrik

5.3.6.4. Wertung

Nach beiden Metriken lässt sich vermuten, dass im ersten Semester mehr über die Möglichkeiten des Verschiebens aufgeklärt werden sollte. Der Sprung, insbesondere in der CP-Metrik in den verschobenen Prüfungen vom ersten ins zweite Semester lässt vermuten, dass Studenten die Klausuren des ersten Semesters zur Selbsteinschätzung nutzen.

5.4. Studienfortschrittsmessung für einzelne Studenten

Unter Studienfortschrittsmessung wird in dieser Arbeit verstanden, wie weit ein Studierender in seinem Studium vorangekommen ist. Zur Selbsteinschätzung dieses Fortschrittes stehen den Studierenden im Moment nur wenige Möglichkeiten zur Verfügung. Im Folgenden wird kurz auf die Probleme mit den zur Verfügung stehenden Möglichkeiten eingegangen, um dann Lösungsansätze aufzuzeigen, die die in dieser Arbeit erhobenen statistischen Daten einbeziehen.

5.4.1. Problemstellung

Die Studenten des hier betrachteten Studiengangs können den Fortschritt ihres Studiums nur anhand der erreichten Credit Points bzw. bestandenen Prüfungen messen. Diese Größen sind Generalisierungen, dahingehend dass fast alle Module als gleichwertig für den Studienerfolg angesehen werden.

5.4.2. Lösungsansätze

Aus den in dieser Arbeit genutzten Methoden und den daraus entstandenen Daten und Erkenntnissen stellen sich verschiedene Lösungsansätze dar. Hier wird zum einen eine Fokussierung auf die Schwierigkeit einzelner Prüfungen und zum anderen ein direktes Nutzen der erstellten Prognosedaten angesprochen.

5.4.2.1. Schwierigkeit der Prüfung

Die Kapitel 5.3.2 und 5.3.3 betrachten jeweils Aspekte der einzelnen Modulprüfungen, die aus statistischer Sicht für eine Einschätzung der Schwierigkeit einzelner Module genutzt werden können, nämlich die Fehlversuchsquote und die durchschnittlich erreichte Anzahl an Notenpunkten pro Prüfung. Diese Daten einzubeziehen, bietet mehrere Möglichkeiten, den Studenten tiefere Einblicke in ihren Studienverlauf zu geben.

So können ähnlich wie in 4.5.2 prototypisch gezeigt die Durchschnittsnoten einer Prüfung zusammen mit dem Ergebnis des einzelnen Studenten gezeigt werden. Diese Ansicht kann durch historische Daten vervollständigt werden, die zeigen, wie eine Prüfung im Schnitt der letzten Semester benotet wird. Hierdurch wird dem Studenten für jede Prüfung gezeigt, wo er sich im Vergleich zu den anderen Teilnehmern an der Prüfung und zum historischen Schnitt befindet.

Durch eine Semesterdarstellung, die die Fehlversuchsquoten inkludiert (etwa durch eine Einfärbung oder Größe), können den Studierenden Einblicke dahingehend gegeben werden, welche Prüfungen sich in der Vergangenheit als schwierig dargestellt haben. Hierdurch wird den Studierenden ermöglicht, die Auswahl der Module, die sie hören, an den Schwierigkeitsgrad der selbigen anzupassen. Das Darstellen der tatsächlichen Schwierigkeit einzelner Prüfungen würde auch der Theorie entsprechen, nach der erfolglose Langzeitstudierende ihren eigenen Fortschritt nicht korrekt einschätzen und diesem entgegenwirken.

5.4.2.2. Prognostizierte Daten

Die in 5.1 genutzten Verfahren zu Erstellung von Studienerfolgsprognosen liefern zum größten Teil auch Wahrscheinlichkeitswerte. Diese können für jeden Studierenden angezeigt werden und würden diesem auf Grundlage des verwendeten statistischen Modells eine Einschätzung seiner Erfolgswahrscheinlichkeit liefern. Da die Grundlagen dieser Einschätzung und ihre statistische Anwendung nicht zwingend intuitiv erkennbar sind, ist hier jedoch fragwürdig, ob ohne Beratung die korrekten Schlüsse gezogen werden.

5.4.3. Fazit

Hier wird aufgrund der persönlichen Erfahrung des Autors angenommen, dass mehr Wissen zu einer besseren Studienplanung genutzt werden kann und auch genutzt werden wird. Aufgezeigt wurden Möglichkeiten, verschiedene statistische Auswertungen der Notendatenbank und aus diesen erstellte Prognosen direkt an die Studierenden weiterzugeben. Eine Überprüfung dahingehend, ob die gewünschten Effekte der Effizienzsteigerung im Studium eintreten, kann endgültig nur retrospektiv etwa nach testweiser Einführung in einer Kohorte stattfinden.

5.5. Fazit

Abschließend werden die Erkenntnisse der Evaluierung hier noch einmal zusammengefasst und mit bereits bekannten Datenpunkten verbunden. Es wurde gezeigt, dass eine Studienerfolgsprognose aus den Daten des ersten Semesters mit höherer Zuverlässigkeit möglich ist, die besten Prognoseergebnisse werden mit technischen Kursen erzielt. Verbleibende Unsicherheiten in der Prognose beruhen wahrscheinlich auf nicht erfassten externen Faktoren. Im Gegensatz hierzu scheint eine Prognose von Studenten, die lange erfolglos studieren, aus den vorhandenen Daten nicht möglich. Sowohl für die Fehlversuchsquoten als auch für

die durchschnittlichen Noten wurde ein Trend zu besseren Ergebnissen in höheren Semestern festgestellt.

Schon in einer früheren Arbeit wurde gezeigt, dass Alter und Geschlecht der Studierenden keinen Einfluss auf den Studienerfolg haben (Steenbuck, 2014). Für die Kohorte konnte diese Aussage hier auch für die Art der Hochschulzugangsberechtigung gezeigt werden, die keinen Einfluss auf den Studienerfolg hat. Im Gegensatz dazu hat die Bewerbungserfolgsquote (also die Quote von Bewerbern, denen ein Studienplatz zugesagt wurde) Einfluss auf den Erfolg der Kohorte. An dieser Stelle wird daraus abgeleitet, dass die als Auswahlkriterium verwendete Note der Hochschulzugangsberechtigung ihren Zweck erfüllt.

Für den Studienverlauf konnte gezeigt werden, dass sich dieser zwar bei einem Großteil der Studenten absolut nach hinten verschiebt (viele Studenten erreichen erst nach 7 oder 8 Semestern einen Abschluss), aber die Reihenfolge, in der Module gehört werden, mit einer gewissen Unschärfe (ca. 1 Semester) dicht an der in der Prüfungsordnung definierten Reihenfolge bleibt.

Teil III.

Schluss

6. Fazit und Ausblick

Abschließend wird zuerst ein Fazit der wesentlichen Punkte gezogen und dann ein Ausblick auf mögliche weitere Arbeiten gegeben. Das Fazit gliedert sich hier in einen technischen Teil (Kapitel 6.1.1) und einen statistischen Teil (Kapitel 6.1.2). Der Ausblick ist dreigeteilt in einen Ausblick auf mögliche Generalisierungen der hier gewonnenen Erkenntnisse (Kapitel 6.2.1), eine Erweiterung der Datenbasis um qualitative Daten (Kapitel 6.2.2) und eine Verbreiterung der vorhandenen quantitativen Daten (6.2.3).

6.1. Fazit

Hier wird das Fazit dieser Arbeit in zwei Teilen gezogen. Als erstes wird der technische Teil des Designs und die Implementierung betrachtet und als zweites die statistischen Ergebnisse, die gezeigt werden konnten.

6.1.1. Design/Implementierung

Der gewählte technische Ansatz, die Daten durch einen Parser aus dem Stisysexport auszulesen, hat sich als erfolgreich und robust auch gegenüber Änderungen im Format bei später erfolgten Exports gezeigt. Die gewählten Werkzeuge *R* und *RapidMiner* haben die benötigte Bandbreite an statistischen Verfahren jeweils abgedeckt. Ein ständiger Wechsel zwischen den Werkzeugen, wie er während der Durchführung dieser Arbeit vorgenommen wurde, ist nicht zwingend erforderlich und wurde hier nur verwendet, um unterschiedliche Implementierungen zu testen und vereinzelt die Vorteile jeweils eines Werkzeuges (z.B. Visualisierung von Regressionen in R) ausnutzen zu können. Die untersuchten Methoden sollten sich von technischer Seite mit moderatem Aufwand auf die Produktionsumgebungen der Hochschule umsetzen lassen. Der Aufwand für die einzelnen Phasen des KDD-Prozesses war für diese Arbeit relativ gleich verteilt, was auch an der bereits geleisteten Vorarbeit ((Steenbuck, 2014), (Steenbuck, 2015a), (Steenbuck, 2015b)) liegt. Wird diese mit einbezogen, fließt erwartungsgemäß mehr Zeit in die vorbereitenden Phasen des KDD ein.

6.1.2. Statistische Ergebnisse

In diese Arbeit wurden Methoden gezeigt und an einem Studiengang des Departments Informatik der Hochschule für Angewandte Wissenschaften Hamburg validiert, die aus den Studienverlaufsdaten[1] des ersten Studienjahres eine Prognose des späteren Studienerfolgs (definiert als, "hat einen Abschluss des Studienganges Technische Informatik erworben") erlauben (siehe Kapitel 5.1). Neben diesen Prognosen wurde gezeigt, dass eine präzise Vorhersage der Studiendauer insbesondere der Studiendauer von nicht erfolgreich Studierenden aus diesen für diese Arbeit vorliegenden Daten nicht möglich ist (siehe Kapitel 5.2). Erfolglose Langzeitstudierende unterscheiden sich hinsichtlich des in Noten und Credit Points gemessen Studienfortschrittes nicht von erfolglosen Studierenden, die das Studium der Technischen Informatik innerhalb der ersten 4 Semester beenden. Hier wird hypothetisiert, dass Gründe für ein längeres Festhalten an einem erfolglosen Studium außerhalb der quantitativ erfassten Umwelt dieser Arbeit liegen. Eine weitere, auch qualitative Untersuchung der Gründe scheint notwendig.

Es konnte auch das Funktionieren der Zulassungskriterien zum Studium für den Studiengang Technische Informatik gezeigt werden. Insgesamt zeigt sich bei einer geringeren Bewerberzahl, also einer sinkenden Möglichkeit der Hochschule, eine Auswahl der Studierenden vorzunehmen, auch ein geringer ausfallender Studienerfolg, gemessen an der Abschlussquote. Gemessen an den durchschnittlichen Credit Points pro Semester ist dieser Unterschied noch ausgeprägter (siehe Kapitel 5.3.4).

Mit den hier gezeigten Methoden lassen sich auch individuelle Studienprognosen erstellen, die einem Studenten ein konkretes Bild davon geben, wo er sich im Blick auf den angestrebten Abschluss und im Vergleich zu anderen Studenten, die in der Vergangenheit ähnliche Leistungen hatten, befindet.

6.2. Ausblick

Im Folgenden wird ein Ausblick auf drei unterschiedliche Bereiche gegeben. Dies sind die Generalisierbarkeit der in dieser Arbeit gewonnenen Erkenntnisse, die Frage der Erweiterung der hier genutzten quantitativen Daten um qualitative Datenquellen und zuletzt die Frage nach einer breiten Datenbasis, um Bildungverläufe außerhalb des untersuchten Studienganges verfolgen zu können.

[1] Noten, Credit Points

6.2.1. Generalisierung

Eine wichtige Frage an dieser Stelle ist die Generalisierbarkeit der Erkenntnisse dieser Arbeit auf andere Studiengänge und Hochschulen. Hier ist als notwendige Voraussetzung das **studienbegleitende Prüfungswesen** zu nennen, durch das die genutzten Daten erst entstehen. Mit Bachelor- und Masterstudiengängen setzt sich diese Art der Leistungskontrolle in größerem Maße durch. Im Gegensatz hierzu sind beispielsweise Diplomstudiengänge zu sehen, die deutlich weniger bis keine studienbegleitenden Leistungskontrollen haben und diese vielmehr nur zum Vordiplom und Diplom durchführen. Sinnvoll erscheint die Frage danach, inwiefern ein ausgeprägtes **Studiengangsprofil** Einfluss auf die Prognosequalität hat. Umso enger die Inhalte des Untersuchungszeitraumes und genauer der als Prediktoren gewählten Leistungen mit den Inhalten des Studium zusammenhängen, desto wahrscheinlicher scheint es, dass eine aussagekräftige Prognose gebildet werden kann. Schlussendlich ist die **Studienmotivation** der Studierenden zu betrachten, d.h. ist ein Abschluss für die Studierenden notwendig oder nicht. Die Informatik bildet hier insofern einen Sonderfall, als dass die Einkommensunterschiede zwischen Bachelor- und Masterabsolventen gering sind und eine Schranke beim Arbeitsmarktzugang kaum gegeben ist. Ein deutlich anderes Bild bieten hier zum Beispiel Lehramtsstudiengänge, deren erfolgreicher Abschluss in der Regel eine notwendige Voraussetzung für den Zugang zum verbeamteten Lehrerberuf darstellt. Wenn ein Abschluss wie beim Lehramt quasi zwingend notwendig ist, werden wahrscheinlich weniger den Noten nach erfolgreiche Studenten ihr Studium abbrechen als in dem hier betrachteten Informatik-Studiengang, bei dem für talentierte Studenten auch ohne Abschluss immer die Option auf eine gut bezahlte Stelle in der Wirtschaft besteht.

6.2.2. Qualitative Daten

Eine wesentliche Verbesserung im Verständnis der Hochschule zu speziell ihren erfolglosen Mitgliedern würde durch die Einführung strukturierter und effizienter Maßnahmen zur Erforschung der Gründe des Studienendes erreicht werden können. Eine solche Befragung findet im Moment nur per E-Mail mit Teilnahmequoten im einstelligen Prozentbereich statt, außerdem ist anzunehmen, dass durch Effekte der Selbstselektion keine repräsentative Stichprobe an den Befragungen teilnimmt. Hier könnten die im Rahmen dieser Arbeit erarbeiteten Algorithmen eine unterstützende Rolle bei der Auswahl zu befragender Studierender spielen. Denkbar wären zum Beispiel Interviews mit erfolgreichen und erfolglosen Studierenden, für die quantitative Methoden mit hoher Wahrscheinlichkeit das jeweils gegenteilige Ergebnis prognostiziert haben. Hier könnte die Frage beantwortet werden, was Faktoren des erfolgreichen bzw. vergeblichen Studierens sind.

6.2.3. Verbreiterung der Datenbasis

Diese Arbeit hat die Anwendbarkeit statistischer Methoden auf einen Studiengang untersucht. Weitere Fragestellungen beziehen sich auf die Bildungsabschlüsse von Studenten, die nicht erfolgreich in diesem Studiengang studiert haben. Offen ist, ob diese etwa überwiegend an andere Hochschulen gewechselt sind oder eine Industrie- und Handelskammer anerkannte duale Ausbildung aufnehmen. Und unterscheiden sich die Bildungserfolge im weiteren Leben nach den Teilerfolgen im untersuchten Studiengang? Hier stellt sich letztendlich die Frage nach Bildungsverläufen in der bundesweiten Ausbildungs- und Hochschullandschaft und speziell nach den Verknüpfungs- und Berührungspunkten dieser beiden Bildungsumgebungen.

Literaturverzeichnis

[cheStudierbarkeit] : *Studierbarkeit CHE Definition*. – URL http://www.che-ranking.de/methodenwiki/index.php/Studierbarkeit. – Zugriffsdatum: 2016-03-14

[Beierle und Kern-Isberner 2008] BEIERLE, Christoph ; KERN-ISBERNER, Gabriele: *Methoden wissensbasierter Systeme*. 4. Edition. Wiesbaden : Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, 2008. – ISBN 978-3-8348-0504-1

[Borrego u. a. 2005] BORREGO, M.J. ; PADILLA, M.A. ; OHLAND, M.W. ; ANDERSON, T.J.: Graduation Rates, Grade-Point Average, and Changes of Major of Female and Minority Students Entering Engineering. In: *Proceedings Frontiers in Education 35th Annual Conference*, Ieee, 2005, S. T3D–1–T3D–6. – URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1611931. – ISBN 0-7803-9077-6

[Cleve und Lämmel 2014] CLEVE, J ; LÄMMEL, U: *Data Mining*. De Gruyter Oldenbourg, 2014 (De Gruyter Studium). – URL https://books.google.de/books?id=4i2nngEACAAJ. – ISBN 9783486713916

[Conway und White 2012] CONWAY, Drew ; WHITE, John M.: *Machine learning for hackers*. Sebastopol, CA : O'Reilly Media, 2012. – URL http://www.amazon.com/Machine-Learning-Hackers-Drew-Conway/dp/1449303714. – ISBN 9781449303716 1449303714

[Cramer 2002] CRAMER, Jan: The Origins of Logistic Regression / Tinbergen Institute. URL http://econpapers.repec.org/RePEc:tin:wpaper:20020119, 2002 (02-119/4). – Tinbergen Institute Discussion Papers

[Elkan 2010] ELKAN, Charles: *Predictive analytics and data mining*. URL http://cseweb.ucsd.edu/users/elkan/255/dm.pdf, 2010. – 1–164 S. – ISBN 9780128014608

[EQA 2009] EQA: *Analyse der Studienbedingungen im Sommersemester 2009*. 2009

[Fayyad u. a. 1996] FAYYAD, Usama M. ; PIATETSKY-SHAPIRO, Gregory ; SMYTH, Padhraic ; FAYYAD, Usama M. (Hrsg.) ; PIATETSKY-SHAPIRO, Gregory (Hrsg.) ; SMYTH, Padhraic (Hrsg.) ; UTHURUSAMY, Ramasamy (Hrsg.): *Advances in Knowledge Discovery*

and Data Mining. Menlo Park, CA, USA : American Association for Artificial Intelligence, 1996. – 1–34 S. – URL http://dl.acm.org/citation.cfm?id=257938.257942. – ISBN 0-262-56097-6

[Gabriel u. a. 2009] GABRIEL, R ; GLUCHOWSKI, P ; PASTWA, A: *Data Warehouse & Data Mining*. W3L-Verlag, 2009. – URL https://books.google.de/books?id=hm9tKfg21L8C. – ISBN 9783937137667

[Golding und Donaldson 2006] GOLDING, Paul ; DONALDSON, Opal: Predicting Academic Performance. In: *Proceedings. Frontiers in Education. 36th Annual Conference*, Ieee, 2006, S. 21–26. – URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4117161. – ISBN 1-4244-0256-5

[Hunt u. a. 1966] HUNT, Earl B. ; MARIN, Janet ; STONE, Philip J.: *Experiments in induction*. Academic Press, 1966. – URL https://books.google.de/books?id=sQoLAAAAMAAJ

[Konvalina u. a. 1983] KONVALINA, John ; WILEMAN, Stanley A. ; STEPHENS, Larry J.: Math Proficiency: A Key to Success for Computer Science Students. In: *Commun. ACM* 26 (1983), may, Nr. 5, S. 377–382. – URL http://doi.acm.org/10.1145/69586.358140. – ISSN 0001-0782

[Kultusministerkonferenz] KULTUSMINISTERKONFERENZ: Ländergemeinsame Strukturvorgaben für die Akkreditierung von Bachelor- und Masterstudiengängen. Beschluss vom 10.10.2003 i.d.F vom 04.02.2010. Kultusministerkonferenz

[Land und Fischer 2012] LAND, Sebastian ; FISCHER, Simon: RapidMiner in academic use. (2012)

[Meisel 2005] MEISEL, Andreas: *Workload-Erfassung 2005*. 2005

[Meisel 2014] MEISEL, Andreas: *TI-Studierendenbefragung Basiszahlen*. 2014

[Quinlan 1979] QUINLAN, JR: Discovering rules by induction from large collections of examples. In: MICHIE, D (Hrsg.): *Expert Systems in the Micro-Electronic Age*. Edinburgh : Edinburgh University Press, 1979

[Rapid-i 2010] RAPID-I: RapidMiner 5.0 GUI Guide. (2010)

[RapidMiner] RAPIDMINER: *RapidMiner*. – URL https://rapidminer.com/the-core-of-rapidminer-is-open-source/. – Zugriffsdatum: 2016-02-19

[Romero und Ventura 2010] ROMERO, C ; VENTURA, S: Educational data mining: a review of the state of the art. In: *IEEE Transaction on Systems, man, and cybernetics—Part C: Applications and Reviews* 40 (2010), Nr. 6, S. 601–618. – URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5524021

[RStudio] RSTUDIO: *RStudio*. – URL https://www.rstudio.com/

[Runkler 2015] RUNKLER, T A.: *Data Mining: Modelle und Algorithmen intelligenter Datenanalyse*. Springer Fachmedien Wiesbaden, 2015 (Computational Intelligence). – URL https://books.google.de/books?id=AZuoCgAAQBAJ. – ISBN 9783834821713

[Russell und Norvig 2010] RUSSELL, Stuart J. ; NORVIG, Peter ; EDUCATION, Pearson (Hrsg.): *Artificial Intelligence - A Modern Approach*. 3. 2010. – ISBN 978-0-13-207148-2

[Stackoverflow] STACKOVERFLOW: *What is the relation between the number of Support Vectors and training data and classifiers performance?*. – URL http://stackoverflow.com/questions/9480605/what-is-the-relation-between-the-number-of-support-vectors-and-train – Zugriffsdatum: 2016-02-22

[(Statistikamt Nord) 2015] (STATISTIKAMT NORD): Statistisches Jahrbuch Hamburg / Nord, Statistikamt. Hamuburg, 2015. – Forschungsbericht. – 145–152 S

[Steenbuck 2014] STEENBUCK, Oliver: Statistische Prognose von Studienerfolgen an der Hochschule für Angewandte Wissenschaften Hamburg / University of Applied Science Hamburg. Hamburg, 2014. – Forschungsbericht. – 1–4 S

[Steenbuck 2015a] STEENBUCK, Oliver: Statistische Prognosen zukünftigen Studienerfolges / University of Applied Science Hamburg. Hamburg, 2015. – Forschungsbericht. – 1–17 S

[Steenbuck 2015b] STEENBUCK, Oliver: Statistische Prognosen zukünftigen Studienerfolges, Visualisierung und Ausblick / University of Applied Science Hamburg. Hamburg, 2015. – Forschungsbericht. – 1–17 S

[The R Foundation] THE R FOUNDATION: *The R Foundation*. – URL https://www.r-project.org/

[Thomas Breuel 2010] THOMAS BREUEL, Faisal S.: AutoMLP: Simple, Effective, Fully Automated Learning Rate and Size Adjustment, Online, 2010

[Tsang u. a. 2005] TSANG, Ivor W. ; KWOK, James T. ; CHEUNG, Pak-Ming: Core Vector Machines: Fast SVM Training on Very Large Data Sets. In: *Journal of Machine Learning Research* 6 (2005), S. 363–392. – URL http://eprints.pascal-network.org/archive/00003072/. – ISBN 1532-4435

[Wu u. a. 2008] WU, Xindong ; KUMAR, Vipin ; ROSS, Quinlan J. ; GHOSH, Joydeep ; YANG, Qiang ; MOTODA, Hiroshi ; MCLACHLAN, Geoffrey J. ; NG, Angus ; LIU, Bing ; YU, Philip S. ; ZHOU, Zhi H. ; STEINBACH, Michael ; HAND, David J. ; STEINBERG, Dan: *Top 10 algorithms in data mining*. 2008. – 1–37 S. – ISBN 1011500701

A. Anhang

A.1. Tabellen

A.1.1. Durchschnittliche Noten

Prüfung	Fachsem.	Note
Bachelorarbeit	6	12,4708
Software Engineering 2	4	11,2429
SE2/PL/SY	4	11,0598
Betriebswirtschaft	5	10,8599
Betriebswirtschaftslehre 2	4	10,7431
System- und Echtzeitprogrammierung	4	10,5401
Prozesslenkung	6	10,4815
Digitaltechnik 2	3	10,2368
Grundlagen systemnahes Programmieren	2	10,1742
Betriebssysteme	3	10,0164
Rechnerstrukturen	4	9,8951
Betriebswirtschaftslehre 1	3	9,8470
Mathematik 3	3	9,8440
Software Engineering 1	3	9,7896
Datenbanken	2	9,6697
Verteilte Systeme	5	9,5957
Numerik und Stochastik	4	9,5214
Algorithmen und Datenstrukturen	3	9,3977
Digitaltechnik	3	9,3938
Computer Engineering	5	9,3895
Automatentheorie und formale Sprachen	2	9,3292
Grundkurs technische Informatik	1	9,1232
Programmieren 1	1	9,0744
Digitaltechnik 1	2	9,0248
Programmieren 2	2	9,0062
Grundlagen Elektrotechnik 2	2	8,9858

Mathematik 2	2	8,7226
Grundlagen Elektrotechnik 1	1	8,6596
Grundlagen Rechnernetze	4	8,6438
Mathematik 1	1	8,3596
Analysis und Lineare Algebra	3	8,1518
Mathematische Grundlagen	1	7,8647

Tabelle A.1.: Durchschnittliche Noten aller Modulprüfungen

A.1.2. Fehlversuchsquoten

Prüfung	Fachsem.	Fehlversuchsquote
Praktikum Programmieren 1	1	52,3810 %
Mathematische Grundlagen	1	39,4737 %
Grundlagen Elektrotechnik 1	1	32,9897 %
Analysis und Lineare Algebra	3	31,6832 %
Mathematik 1	1	30,8824 %
Praktikum Grundlagen Systemnahes Programmieren	3	28,2110 %
Programmieren 1	1	26,6811 %
Mathematik 2	2	26,4706 %
Programmieren 2	2	25,5376 %
Digitaltechnik	3	24,8756 %
Software Engineering II/Prozesslenkung/System- und Echtz	4	22,9050 %
Numerik und Stochastik	4	20,5556 %
Grundlagen der Informatik	1	19,7917 %
Computer Engineering	5	19,1558 %
Grundlagen Rechnernetze	4	18,6120 %
Automatentheorie und formale Sprachen	2	16,6667 %
Grundlagen der Elektrotechnik 2	2	16,6259 %
Digitaltechnik 1	2	15,4762 %
Software Engineering I	3	14,8997 %
Verteilte Systeme	5	13,1313 %
Datenbanken	2	12,3832 %
Algorithmen und Datenstrukturen	3	11,5385 %
Mathematik 3	3	11,0345 %
Prozesslenkung	6	10,7914 %
System- und Echtzeitprogrammierung	4	10,7914 %
Betriebssysteme	3	10,4790 %
Grundlagen Systemnahes Programmieren	2	10,0478 %

Digitaltechnik 2	3	9,6154 %
Software Engineering 2	4	8,3333 %
Betriebswirtschaft	5	6,1798 %
Rechnerstrukturen	4	6,1644 %
Betriebswirtschaftslehre 1	3	5,9783 %
Bachelorarbeit	6	5,6000 %
Betriebswirtschaftslehre 2	4	4,1379 %
Wahlpflichtmodul Sammler	NULL	2,3778 %
GW Sammler	NULL	1,5267 %

Tabelle A.2.: Durchfallquoten im Erstversuch, Kohorte $<=$ 2010WS

A.2. Listings

A.2.1. Parser

Listing A.1: Antlr-Grammatik

```
grammar CSV;
options {
    // antlr will generate java lexer and parser
    language = Java;
    // generated parser should create abstract syntax tree
    output = AST;
}
   @lexer::header {
       package pj1.parser.generated;
   }


@parser::header {
 package pj1.parser.generated;

 //add imports
 import pj1.parser.ParserException;
 import org.apache.log4j.Logger;
}

@parser::members{
```

```
  static Logger logger = Logger.getLogger("Parser");

    public void displayRecognitionError(String[] tokenNames,
                                        RecognitionException
    e) {
        String hdr = getErrorHeader(e);
        String msg = getErrorMessage(e, tokenNames);
        logger.fatal(hdr);
        logger.fatal(msg);
        throw new ParserException(hdr + " : "  + msg, e);
    }
}
  //{System.out.println("exam number: " + $uniqueID.text);}

 data   :        exam+;

exam : uniqueID! ','! studentID^ ','! immaSemester ','!
   status ','!  examSemester ','!
('"'?)! cps ('"'?)!  ','! (note|unbenotet) ','! fachkuerzel
   ','!  fachId ','! fachBezeichn ('.'?)! ','!  fachSemester
    ','! annerkant ','!
 geburtstag ','! geschlecht  ;


uniqueID        :       INT ;
studentID       :       INT ;
immaSemester
        :       SEMESTER;
status  :       ('B-TI'|
                'BCH'|
                'MST'|
                'zwex'|
                'M-INF'|
                'weg'|
                'DPL'|
                'B-AI'|
                'TI'|
                'B-ECS'|
                'MES-M'|
                'E'|
```

```
                'B-MT');
examSemester
        :       SEMESTER;
cps     :        (INT^ (',' INT)?);
note    :        INT;
unbenotet
        :       ('erf'|'n.e.');
fachkuerzel
        :       IDENTIFIER;
fachId  :       INT;
fachBezeichn
        :       IDENTIFIER;
fachSemester
        :       INT;
annerkant       :       (' '|'Ja');

geburtstag      :       (INT^ '.'! INT '.'! INT);

geschlecht      :       ('M'|'W');

INT :  DIGIT+ ;
SEMESTER:       DIGIT+('WS'|'SS');
IDENTIFIER  :   (LETTER | DIGIT | OTHERCHARS) ( LETTER |
   DIGIT | OTHERCHARS)*;
WS: (' '|'\n'|'\r')+   {$channel=HIDDEN;} ; // ignore
   whitespace

fragment LETTER  : ALPHA;
fragment DIGIT   : '0'..'9';
fragment ALPHA   : ('a'..'z'|'A'..'Z');
fragment OTHERCHARS  :   ('ä'|'ö'|'ü'|'Ä'|'Ö'|'Ü'|'-'|'/'|'
   '|'ß');
```

A.2.2. Datenbankschema

Listing A.2: Database Creation SQL

```
CREATE database student_data charset utf8;
```

```
USE student_data;

CREATE TABLE students
      (
      id INT PRIMARY KEY NOT NULL,
      kohorte VARCHAR(255) NOT NULL,
      studiengang VARCHAR(255) NOT NULL,
      geschlecht VARCHAR(255) NOT NULL,
      birthday DATE NOT NULL,
      ageAtImma FLOAT NOT NULL,
      isGraduated INT NOT NULL,
      studienDauer FLOAT,
   gpa FLOAT,
   tutorialMarker INT
      );


CREATE TABLE cps
   (
      student_id INT NOT NULL,
      cp1 FLOAT NOT NULL,
      cp2 FLOAT NOT NULL,
      cp3 FLOAT NOT NULL,
      cp4 FLOAT NOT NULL,
      cp5 FLOAT NOT NULL,
      cp6 FLOAT NOT NULL,
      FOREIGN KEY (student_id) REFERENCES students(id)
   );

CREATE TABLE C2
   (
      kohorte VARCHAR(255) NOT NULL,
      bewerber_erfolgs_quote FLOAT NOT NULL
   );

CREATE TABLE C3
   (
      kohorte VARCHAR(255) NOT NULL,
      abi FLOAT NOT NULL,
```

```
      fh FLOAT NOT NULL,
      vorbildung FLOAT NOT NULL,
      berufs FLOAT NOT NULL
   );


CREATE TABLE courses
      (
      id INT PRIMARY KEY NOT NULL,
      cp FLOAT NOT NULL,
      fachkrz VARCHAR(255) NOT NULL,
      bezeichner VARCHAR(255) NOT NULL
      );

CREATE TABLE fachsem
      (
      fachkrz VARCHAR(255) PRIMARY KEY NOT NULL,
      fachsem INT NOT NULL
   );

CREATE TABLE attendings
      (
      student_id INT NOT NULL,
      course_id INT NOT NULL,
      note INT NOT NULL,
      semester VARCHAR(255) NOT NULL,
      klausur bool NOT NULL,
      isAnnerkant bool NOT NULL,
      bestanden bool NOT NULL,
   hochschulsemester INT,
      pruefungsversuch INT,
   cpVorPruefung INT,
      CPVersuchtVorPruefung INT,
      FOREIGN KEY (student_id) REFERENCES students(id),
      FOREIGN KEY (course_id) REFERENCES courses(id)
      );

CREATE INDEX student_id_idx on attendings (student_id) USING BTREE;
CREATE INDEX semester_idx on attendings (semester) USING BTREE;
CREATE INDEX course_id_idx on attendings (course_id) USING BTREE;
```

A.2.3. Dataload Skript

Listing A.3: Skript zum Datenladen

```
#/bin/sh

#insertStudents and insert exams must be generated beforehand
#structure
echo "deleting database"
mysql < ./delete_database.sql
echo "creating database"
mysql < ./create_database.sql
echo "loading stored procedures and functions"
mysql student_data < proc/cpVersuchtVorHochschulsem.sql
mysql student_data < proc/cpErreichtVorHochschulsem.sql
mysql student_data < proc/gpa.sql
mysql student_data < proc/cpsInSemStud.sql
mysql student_data < proc/cps.sql
mysql student_data < proc/cpsVersucht.sql
mysql student_data < proc/studiendauer.sql
mysql student_data < proc/maxCps.sql
mysql student_data < proc/triedFirstSem.sql
mysql student_data < proc/plausibilityControl.sql
echo "creating views"
mysql < ./create_views.sql

#data
echo "inserting student data"
mysql < ./insertStudents.sql
echo "inserting veranstaltungen"
mysql < ./insertVeranstaltungen.sql
echo "inserting relations"
mysql < ./insertRelation.sql
echo "inserting fachsem table"
mysql < ./insertFachsem.sql
echo "inserting c1 table"
mysql < ./insertC1.sql
echo "inserting c2 table"
mysql < ./insertC2.sql
echo "inserting c3 table"
mysql < ./insertC3.sql
```

```
#berechnete Daten
echo "creating bayes graduation predictions"
mysql student_data < ./bayes_predictions.sql
echo "creating hochschulsem data"
ruby addHochschulsemester.rb > ./insertHochhschulsem.sql
echo "inserting hochschulsem data"
mysql < ./insertHochhschulsem.sql
echo "creating pruefungsversuch data"
ruby addPrüfungsversuch.rb > insertPruefungsversuch.sql
echo "inserting pruefungsversuch data"
mysql < ./insertPruefungsversuch.sql
echo "creating cp before data"
mysql student_data -e 'update attendings
set cpVorPruefung =
cpErreichtVorHochschulsem(student_id, hochschulsemester);'
echo "creating cp tried before data"
mysql student_data -e 'update attendings
set CPVersuchtVorPruefung =
cpVersuchtVorHochschulsem(student_id, hochschulsemester);'
echo "creating studiendauer data"
mysql student_data -e 'update students
set studienDauer = studiendauer(id);'
echo "creating gpa data"
mysql student_data -e 'update students set gpa = gpa(id);'
echo "creating cps erreicht data"
mysql student_data -e 'CALL cps();'
echo "creating cps versucht data"
mysql student_data -e 'CALL cpsVersucht();'
echo "Normalizing WP and GW"
mysql < ./normalizeWPGW.sql
echo "creating statistik tables"
mysql student_data < proc/studentsKlausuren.sql
```

A.3. Abbildungen

A.3.1. RapidMiner

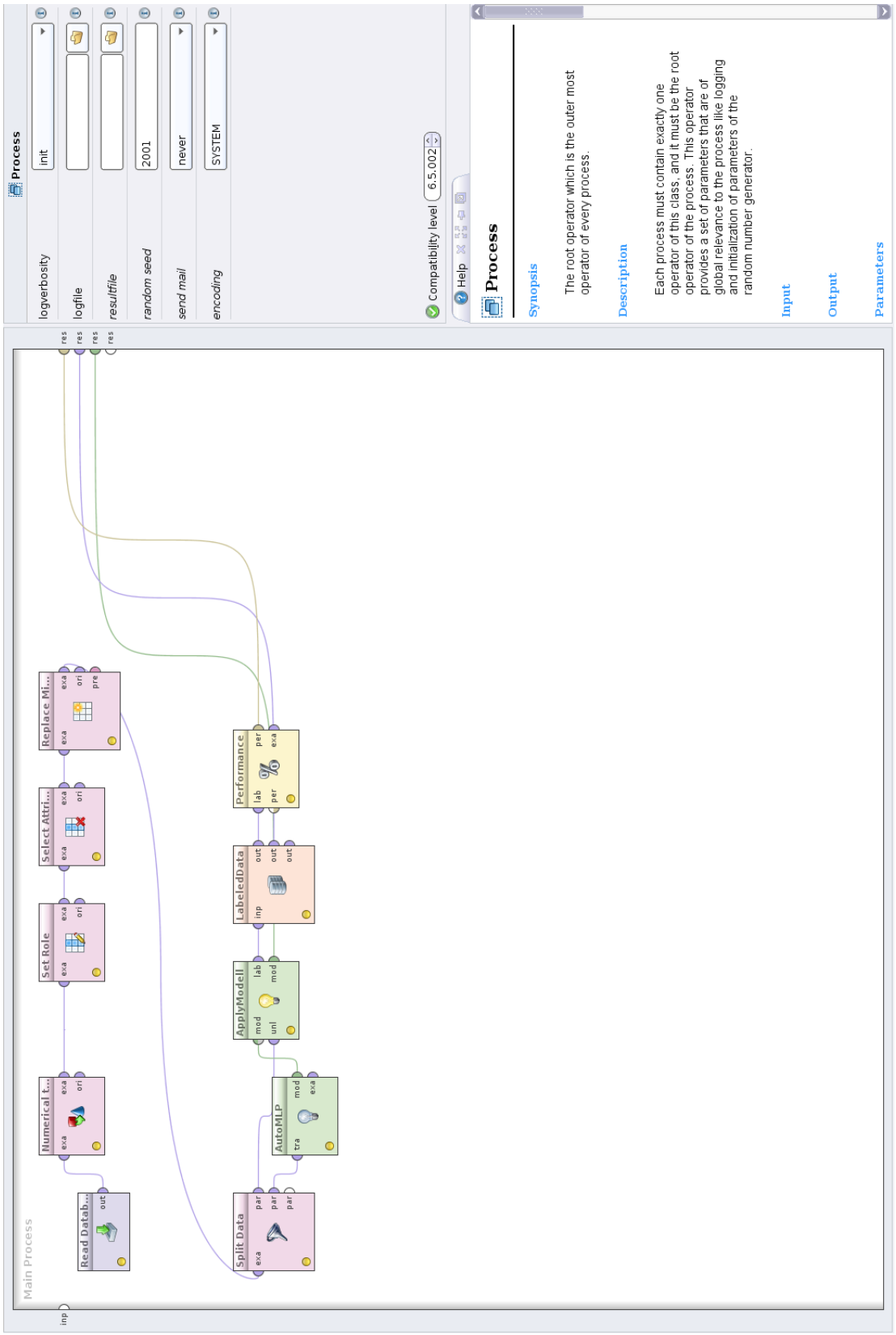


Abbildung A.1.: Beispiel; RapidMiner Flowdesign, groß

A.3.2. RStudio

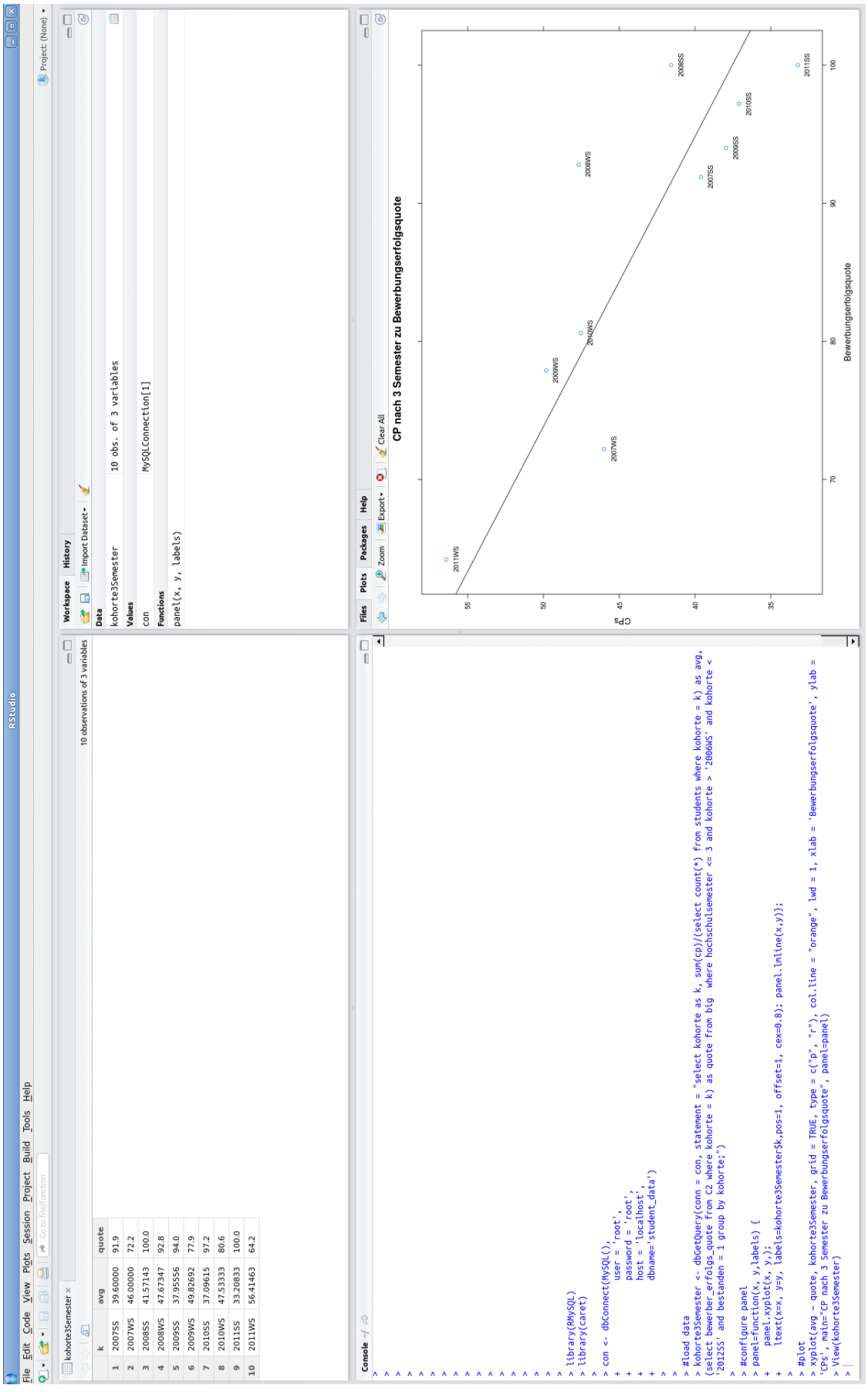


Abbildung A.2.: Beispiel; RStudio, groß

Glossar

CHE
: CHE Hochschulranking ist eine Organisation, die seit 2003 regelmäßig in Zusammenarbeit mit der Zeitung *Die Zeit* ein umfassendes Hochschulranking primär deutscher Hochschulen basierend auf Befragungen von Studierenden und Hochschullehrern veröffentlicht.

Credit Points
: Die Arbeitsbelastung (work load) eines Studierenden für einen Kurs wird in Leistungspunkten gemessen. Ein Leistungspunkt entspricht einer Arbeitsbelastung von 25 bis max. 30 Stunden. Ein Semester beinhaltet üblicherweise 30, ein Studienjahr 60 Leistungspunkte. Die Leistungspunkte werden in den Prüfungsordnungen der Hochschulen synonym als Credit Points, ECTS Punkte oder Leistungspunkte bezeichnet (Kultusministerkonferenz).

Data Frame
: Eine zusammengesetzte Datenstruktur, ähnlich einem *struct* in C.

Fachsemester
: Das Fachsemester einer Prüfung entspricht dem Semester, in dem es laut Prüfungsordnung abgelegt werden sollte.

Hochschulsemester
: Die Anzahl Semester, die ein Studierender im Studiengang TI zu einem bestimmten Zeitpunkt verbracht hat. Bsp. Studierender Immatrikulation SS2005 befindet sich im SS2006 im Hochschulsemester 4.

Logit
: Der Logarithmus einer Wahrscheinlichkeit.

Stisys
: Das Studierendeninformationssystem, das unter anderem im Studiengang Technische Informatik an der Hochschule für Angewandte Wissenschaften Hamburg verwendet

wird. Es enthält alle Prüfungsereignisse, die im Studiengang stattgefunden haben bzw. diesem durch Anerkennung externer Leistungen zugeordnet wurden.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, . April 2016	
Ort, Datum	Unterschrift