



Hochschule Darmstadt
Fachbereich Mathematik und Naturwissenschaften
Studiengang Angewandte Mathematik (Bachelor)

Praktikumsbericht

**“Verbesserung und Weiterentwicklung der Abläufe im Mentorensystem
(geleitet von der Hochschuldidaktischen Arbeitsstelle) des Fachbereichs
Informatik der TU Darmstadt – Klassifizierung von Studierenden mit
schlechtem Prüfungserfolg im ersten Semester“**

Maxi Neubacher

Matrikelnummer:704726

maxi.neubacher@stud.h-da.de

über das abgeleistete Praktikum bei
Technische Universität Darmstadt
Fachbereich Informatik
Hochschulstraße 10
D-64289 Darmstadt

vom 12.09.2011 bis 23.01.2012

Betreuender Professor: Prof. Dr. Dietrich Baumgarten
Betreuer im Praktikumsbetrieb: Dipl. –Inform. Frederik Janssen

Inhalt

1	Informationen zum Praktikumsbetrieb	3
1.1	Technische Universität Darmstadt.....	3
1.2	Fachbereich Informatik	4
1.3	Hochschuldidaktische Arbeitsstelle	4
2	Das Mentorensystem	5
2.1	Ziele und Konzepte	5
2.2	Ablauf erstes Fachsemester	6
2.3	Ablauf zweites Fachsemester	7
3	Aufgabenstellung	8
4	Durchgeführte Aufgaben	9
4.1	Maschinelles Lernen	10
4.2	arff-Datei	11
4.3	Klassifizierung	13
4.4	Auswertung	15
5.	Ausblick	19
6	Anhang	20
7	Quellen	22

1 Informationen zum Praktikumsbetrieb

1.1 Technische Universität Darmstadt

Die Technische Universität Darmstadt (TUD) wurde 1877 gegründet. Auf circa 260.000 Quadratmeter Nutzfläche verteilen sich die 140 Gebäude über das gesamte Stadtgebiet, von dem Standort Lichtwiese, zum Botanischen Garten, zur Stadtmitte, über das Sportstadion, bis zum Flugplatz und Windkanal in Darmstadt-Griesheim. Die Universität besitzt 13 Fachbereiche und 4 Studienbereiche und bietet mit ihren 100 Studiengängen genügend Plätze und Angebote für die über 23.000 Studierenden. Es sind rund 2.200 wissenschaftliche Mitarbeiterinnen und Mitarbeiter, darunter 270 Professorinnen und Professoren, sowie 1.780 nicht wissenschaftliche Mitarbeiterinnen und Mitarbeiter, an der TUD beschäftigt. Außerdem bildet die Universität in mehr als 20 Berufen aus. Zurzeit beläuft sich die Zahl der Auszubildenden auf circa 180. Dass die Universität als international orientiert bezeichnet wird, begründet sich unter anderem durch die hohe Anzahl von 108 Partneruniversitäten, über die ganze Welt verteilt. Auch auf nationaler Ebene pflegt die TU langjährige Kooperationsbeziehungen zu etlichen Wirtschaftsunternehmen, wie zu Vertretern der Automobilindustrie (z.B. BMW, Audi, Opel...), der chemischen Industrie (z.B. Merck, BASF...) und vielen Weiteren.

Die TU Darmstadt ist rechtsfähige Körperschaft des öffentlichen Rechts und seit 2005, per Landesgesetz, eine mit weit reichender Autonomie ausgestattete Hochschule. Sie ist somit bei der Gestaltung von Forschung, Lehre und Transfer, sowie bei Fragen zum Personal, Bau und Grundstück eigenständig und eigenverantwortlich.

Intern wird die Technische Universität Darmstadt von einem Präsidium geleitet, zu dem der Präsident, die Vizepräsidenten und der Kanzler gehören.

Das selbsternannte Ziel und Leitspruch der Technischen Universität Darmstadt: „Wir wollen stets zu den drei besten Technischen Universitäten Deutschlands gehören“, wird durch Spitzenplätze in verschiedenen Rankings, wie beispielsweise zu Leistungen in Forschung und Studiengängen, immer wieder bestätigt. [1]

1.2 Fachbereich Informatik

Schon im Jahr 1964, als die Konzeption von Informatikstudiengängen noch am Anfang stand, wurde die Planung einer Einführung dieses Faches in Darmstadt, durch den damaligen Professor Robert Piloty, vorangetrieben. Nach ihm wurde das Fachbereichsgebäude im Herrngarten benannt, in dem der Fachbereich Informatik größtenteils untergebracht ist.

Zum Wintersemester 1971/72 konnten sich die ersten Studierenden für ein Informatikstudium an der Technischen Universität Darmstadt einschreiben. Zu diesem Zeitpunkt war nur eine Professur besetzt.

Der Fachbereich Informatik an der Technischen Universität Darmstadt zählt heute, laut Rankings und Vergleichen, zu einem der Top-Informatikbereiche deutschlandweit. Der Fachbereich bietet einen Bachelor of Science Informatik und sechs Masterstudiengänge an. Zudem ist er an mehreren interdisziplinären Studiengängen beteiligt. Am Fachbereich arbeiten circa 20 Professorinnen und Professoren. Jeder Professor leitet eine Arbeitsgruppe, die in Forschung und Lehre aktiv ist, viele davon an der Weltspitze. Regelmäßig werden Studierende, über Praktika, Seminare und Abschlussarbeiten, in die Forschung und Lehre der Arbeitsgruppen eingebunden. [2]

1.3 Hochschuldidaktische Arbeitsstelle

Die Hochschuldidaktische Arbeitsstelle (HDA) ist eine wissenschaftliche Einrichtung der Technischen Universität Darmstadt, die im Jahr 1985 zum ersten Mal offiziell in der Historie der Universität erwähnt wird. Sie unterstützt durch Forschung, Aus- und Fortbildung, Beratung, Evaluation und Studienreformprojekte die Weiterentwicklung von Lehre und Studium. Die Arbeitsbereiche und Angebote der HDA resultieren aus eigenen Untersuchungen und aktuellen Forschungsergebnissen sowie aus Anregungen, die durch Kooperationen mit Hochschulmitgliedern, Instituten, Fachbereichen und Gremien der TUD, entstehen. [3]

2 Das Mentorensystem

Im Wintersemester 2006/07 wurde das Mentorensystem im ersten Fachsemester vom damaligen Dekan des Fachbereichs Informatik, Prof. Dr. Karsten Weihe, in Zusammenarbeit mit der Fachschaft als Pilotsystem für Erstsemester im Bachelor of Science Informatik erstmals eingeführt. Im Wintersemester 2007/08 entstand daraus ein Kooperationsprojekt mit der Hochschuldidaktischen Arbeitsstelle (HDA). Seit dem Sommersemester 2010 wurde das System um eine weitere Maßnahme im zweiten Fachsemester erweitert. Im zweiten Semester werden die Studierende individuell beraten und geschult, welche im ersten Fachsemester eine zu geringe Anzahl an Prüfungen bestanden haben. Mit fachkundiger Begleitung durch die HDA wird das Programm für beide Fachsemester kontinuierlich verbessert und schrittweise auf andere Studiengänge der TU Darmstadt in fachspezifischer Ausprägung übertragen.

2.1 Ziele und Konzepte

Durch das Mentorensystem soll den Studierenden der Einstieg in das studentische Leben an der TU Darmstadt deutlich erleichtert werden. Die Studienmotivation soll verbessert werden, individuelle Orientierungshilfen sollen den Übergang von Schule zu Hochschule erleichtern sowie die soziale Integration um nicht zuletzt den Studien- und Prüfungserfolg der Studierenden zu verbessern. Umgekehrt können die Ausrichter des Studiengangs detaillierter und zeitnaher über mögliche Probleme und Nöte der Studierenden erfahren. Dies ermöglicht es zeitnah Abhilfe zu schaffen, wie zum Beispiel eine Hilfestellung zum richtigen Zeitmanagement zu liefern oder auf weitere Beratungsstellen aufmerksam zu machen.

Durch die enge Beziehung zwischen Studierenden und ihren studentischen Mentoren werden weit mehr früher Studienfachwechsler und Studienabbrecher erreicht und die Erkenntnisse über die Gründe sind dadurch umfassender als konventionelle Ansätze wie beispielsweise nachträgliche Befragungen.

2.2 Ablauf erstes Fachsemester

Vor Beginn der Vorlesungszeit werden einige auserwählte ältere Studierende in einer zweitägigen Schulung einführend darin ausgebildet, die Studierenden des ersten Fachsemesters zu spezifischen Themen rund um das Studium zu beraten. Jedem Studienanfänger wird einer dieser Mentoren fest zugeordnet. Mentor und der zu beratende Studierende, der Mentee, treffen sich während der Vorlesungszeit einmal wöchentlich für ca. 15 Minuten. Die Mentees sind durch die Prüfungsordnung zu diesen Treffen verpflichtet. Sie können in diesen Treffen alle Fragen und Probleme zum Studienalltag ansprechen; umgekehrt sprechen die Mentoren ihrerseits Woche für Woche typische Probleme im Studium an. Dabei orientieren sie sich an einem Leitfaden, der semesterweise auf Basis der neuen Erfahrungen verfeinert wird. Für den Beginn des Semesters beinhaltet der Leitfaden eine erste Orientierung an der Universität und am Fachbereich sowie generelle Probleme im Studienalltag. In der Mitte des Semesters werden (selbst-) organisatorische Probleme der Studierenden mit der Bewältigung der wöchentlichen Herausforderungen im Studium intensiv thematisiert. Ungefähr in der Mitte des Semesters simuliert der Mentor eine mündliche Prüfungssituation mit dem Mentee und gibt ihm hinterher detaillierte Rückmeldung. Lernstrategien und Prüfungsvorbereitung sind das Hauptthema gegen Ende des Semesters. Dazu gehört ein Termin zur individuellen Lernplanerstellung für den Prüfungszeitraum. Die Mentoren halten die Ergebnisse jedes Gesprächs und Vereinbarungen mit den Mentees in Gesprächsprotokollen fest. In wöchentlichen Treffen besprechen die Mentoren mit den Verantwortlichen für das Mentorensystem, welche Leitthemen in der kommenden Woche angesprochen werden sollen, und tauschen sich darüber aus, welche konkreten Probleme auftraten und wie Abhilfe geschaffen werden kann.

Neben diesen wöchentlichen Gesprächen zwischen Mentor und Mentee bietet das studentische Mentorensystem separate Termine zur Findung von Lern- und Arbeitsgruppen. In einem Abschlusstreffen in der letzten Vorlesungswoche mit allen Mentoren und Mentees wird den Studierenden noch eine Orientierung für das nächste Semester mitgegeben.

2.3 Ablauf zweites Fachsemester

Ausgangspunkt für das Ausdehnen des Mentorensystems in das zweite Fachsemester ist die Erkenntnis, dass durch frühzeitiges Reagieren auf Schwierigkeiten in der ersten Phase des Studiums ernsthafte Probleme im späteren Studium verhindert werden können. Dafür ist die frühe Identifikation und individuelle Betreuung kritischer Einzelfälle entscheidend. Die Betreuung richtet sich dann allerdings gezielt nur noch auf diejenigen Studierenden, die im ersten Fachsemester geringen Prüfungserfolg hatten. In einem intensiven Beratungsgespräch unter vier Augen mit einem Mitarbeiter der Fachstudienberatung werden die individuellen Probleme genau erläutert. Auf dieser Basis wird mit dem Studierenden die Teilnahme an denjenigen Angeboten vereinbart, die auf sein spezifisches Problemprofil zugeschnitten sind. Dazu gehören Workshops der HDA zu Lerntechniken, Zeitmanagement, Selbstorganisation und effektiver Prüfungsvorbereitung. Nach der Teilnahme an einem der Workshops können weitere Beratungsgespräche mit den Workshopleitern in Anspruch genommen werden.

Organisiert und geleitet wird das Programm im Fachbereich Informatik seit dem Wintersemester 2008/09 von Dipl.-Psych. Sabine General, einer wissenschaftlichen Mitarbeiterin der HDA, die die studentischen Mentoren ausbildet und anleitet, als Ansprechpartner für Studierende im Zusammenhang mit studienrelevanten Fragen dient, die Weiterentwicklung und Neukonzeption der Maßnahmen und zusätzlicher Angebote im Rahmen des Konzepts durchführt.

Das Konzept und die gewonnenen Erkenntnisse sind mehrfach auch außerhalb der TU Darmstadt präsentiert worden, insbesondere in einem von Frau General und Herrn Prof. Dr. Weihe ausgerichteten Workshop auf dem Fakultätentag Informatik 2009. Zudem hat Herr Prof. Dr. Weihe Vorträge an der TU Dortmund und der RWTH Aachen über das Mentorensystem gehalten. In der Informatik in Aachen ist das System nunmehr übernommen worden. Besonders erwähnenswert ist, dass die Fachschaft Magdeburg sich unter dem Namen UniMentor von diesem Konzept (mit Modifikationen) hat inspirieren lassen und für die erfolgreiche Durchführung den Preis für ein Best-Practice Projekt des Fakultätentags Informatik erhalten hat. [4]

3 Aufgabenstellung

Das Mentorensystem im Ablauf des zweiten Semesters richtet sich an die Studierenden die im ersten Semester maximal eine Prüfung bestanden haben. Da die Anzahl derjenigen erfahrungsgemäß zu hoch ist um alle davon einzeln zu beraten, muss diese Gruppe unterteilt werden. Der eine Teil der Studierenden wird zu einem verpflichtenden Vortrag, über mögliche Fehlerquellen des Prüfungsmisserfolges und deren Vermeidung, eingeladen. Jedem dieser Teilnehmer wird anschließend die Möglichkeit zu einem weiteren Beratungsgespräch gegeben. Der andere Teil der Studierenden, die so genannten „Härtefälle“, werden zu einem verpflichtenden Einzelgespräch mit der Fachstudienberatung eingeladen. Mein Aufgabenfeld bestand darin, unter bestimmten Kriterien, potentielle „Härtefälle“ zu erkennen, bei denen ein individuelles Gespräch von Vorteil ist.

Da die Erstsemesterzahlen im Studiengang Bachelor Informatik jährlich steigen (Vom SS 2010 bis zum SS 2011 stieg die Anfängerzahl auf 14% und die Erstsemesterzahl vom WS 10/11 zum WS 11/12 erhöhte sich auf knapp 16%), wird der Aufwand für die Studienberatung den Prüfungserfolg jedes einzelnen Studierenden zu verfolgen, um Einzelentscheidungen darüber zu treffen, ob ein „Härtefall“ vorliegt oder nicht, immer schwieriger und zeitintensiver. Darüber hinaus gestaltet sich das Auswahlverfahren, gerade durch die steigende Anzahl an Studierenden im ersten Fachsemester, als problematisch, da die gegebene Kapazität der Studienberatung für persönliche Gespräche nur etwa 30 Beratungstermine zulässt. Um diesen Aufwand der Datenanalyse zu verringern, ist geplant, die Abläufe so zu verändern, dass weitestgehend automatisiert eine Liste der zu beratenden Studierenden erzeugt wird.

4 Durchgeführte Aufgaben

Nachdem die Studierenden das erste Fachsemester absolviert und Prüfungen geschrieben haben sowie die Klausurergebnisse bekannt sind, sind diejenigen, die maximal eine Prüfung bestanden haben, verpflichtet, mittels dem Onlinesystem Moodle ^[5], einen Fragebogen (der im Abschnitt 6 angehängt ist), über das Semester, auszufüllen. Der Fragebogen beinhaltet außer allgemeinen Angaben zur Wohnsituation und der vorher besuchten Schule auch Fragen zu den aktuell absolvierten Prüfungen, dem Studienverhalten, zu subjektiven Gründen des Prüfungsmisserfolgs sowie zur momentanen privaten Situation. Die Daten der ausgefüllten Fragebögen bilden daher die Grundlage aller weiteren Betrachtungen.

Angedacht war ursprünglich, neben dem Fragebogen, weitere Daten wie erreichte Klausurpunkte, Zwischenklausurergebnisse, Hausübungs- und Bonuspunkte mit in die Bewertung einfließen zu lassen. Hierzu wäre ein System auf Basis einer SQL-Datenbank und PHP-Skripten eingerichtet worden. Nach Gesprächen mit meinem Betreuer und den Verantwortlichen des Mentorensystems hat sich die Frage ergeben, ob es nicht möglich wäre eine Einteilung der Studierenden ausschließlich mit den Daten aus den Fragebögen vorzunehmen. Dadurch wäre man unabhängig von eventuellen Informationen, welche durch die Veranstalter bereit gestellt werden müssten. Zudem werden die Fragebögen durch die Verantwortlichen des Mentorensystems selbst erstellt und könnten in Hinsicht auf eventuell erforderliche Informationen selbstständig und zeitnah verändert werden.

Es wurden speziell nur die Daten der ausgefüllten Fragebögen der Studierenden des Wintersemesters 2010/11 und des Sommersemesters 2011 benutzt, da hier zu jedem der Studierenden, neben dem jeweiligen Fragebogen, bereits eine weitere Angabe über den Status „Härtefall“ oder nicht, vorliegt. Diese Angaben wurden von der Studienberatung noch händisch erstellt, indem jeder Fragebogen der Studierenden sorgfältig überprüft und aufgrund von Erfahrungswerten dann darüber entschieden wurde in welche Kategorie der Studierende eingestuft wird. Was die Studienberatung, mittels Erfahrung, bisher gemacht hat, soll nun der Computer,

mittels Regeln, übernehmen. Mit dem frei verfügbarem Softwaretool „Waikato Environment for Knowledge Analysis“, kurz WEKA, sollen nun geeignete Regeln gefunden werden, nach welchen Kriterien die Studierenden als „Härtefall“ gelten und somit zu einem Fachgespräch mit der Studienberatung eingeladen werden. WEKA wurde an der University of Waikato (Neuseeland) entwickelt, ist in Java geschrieben und stellt verschiedene Techniken aus dem Bereich Maschinelles Lernen bereit. [6]

4.1 Maschinelles Lernen

Maschinelles Lernen ist ein Forschungsgebiet der Informatik und befasst sich mit Computerprogrammen, die durch Erfahrung lernen können. Ein Programm lernt also anhand von Beispielen Gesetzmäßigkeiten zu erkennen.

Definition Maschine Learning:

„ A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.“ (Mitchell 1997, Informatiker und Professor an der Carnegie Mellon University in Pittsburgh, Pennsylvania, USA) [7]

An unserem konkreten Beispiel veranschaulicht, lernt das Computerprogramm, von den Daten der ausgefüllten Fragebögen (experience E), „Härtefälle“ zu erkennen (task T), wenn der Anteil korrekt klassifizierter „Härtefälle“ (performance measure P) sich mit der Erfahrung verbessert.

Maschinelles Lernen wird in vielen Gebieten angewendet, wie zum Beispiel bei der Sprach- und Handschrifterkennung, bei Vorhersagen für die Zukunft (z.B. Wettervorhersage), bei der Steuerung von Robotern sowie bei Diagnosen in der Medizin. Ein weiteres Beispiel ist das Betrachten vergangener Transaktionen, um Finanzinstitutionen das Kreditrisiko der Kunden vorherzusagen. Auch der Warenhandel analysiert erhobene Verkaufsdaten um auf das Kaufverhalten der Kunden zu schließen.

4.2 arff-Datei

Die Daten der ausgefüllten Fragebögen aller Studierender des WS 2010/11 und SS 2011 werden im Attribute-Relation File Format (arff) verarbeitet um anschließend in WEKA eingebunden zu werden.

Beobachtete Beispiele werden in Instanzen (Instances) beschrieben. Jeder ausgefüllte Fragebogen eines Studierenden stellt damit genau eine Instanz dar. Da wir von 123 Studenten jeweils einen Fragebogen zur Verfügung haben, gibt es in dieser Datei 123 Instanzen. Der im Anhang beschriebene Fragebogen besteht aus 52 Fragen, von denen wir aber wissentlich einige für die Datei weggelassen haben da sie für die Fragestellung unerheblich sind, wie zum Beispiel die Tätigkeit vor Beginn des Studiums. Weiterhin wurden einige Fragen, wie beispielsweise die Selbsteinschätzung zur Prüfungsvorbereitung, nicht mit in Betracht gezogen, da das Wissen über das Prüfungsergebnis die Antwort darauf zu sehr beeinflussen würde. Jede Instanz hat daher in unserem Fall 45 Attribute (Attributes). Attribute sind die Eigenschaften einer Instanz und stehen hier für die Antworten der Fragen aus dem Fragebogen. Da die komplette arff-Datei zu umfangreich ist, beschreibe ich diese anhand von einem kurzen Ausschnitt davon mit nur 5 Attributen und 2 Instanzen:

```
@relation fragebogen
```

```
@attribute prüfungen_bestanden numeric
```

```
@attribute gdi1_übung_besucht {fast_immer, häufig, unregelmäßig, selten, k.A.}
```

```
@attribute fgdi1_übung_besucht {fast_immer, häufig, unregelmäßig, selten, k.A.}
```

```
@attribute private_gründe {ja, nein}
```

```
@attribute beratung {ja, nein}
```

```
@data
```

```
0,selten,selten,ja,ja
```

```
1,häufig,?,nein,nein
```

WEKA erkennt vier verschiedene Arten von Attributtypen (Type):

- string = beliebige Zeichenfolgen
- date = Datumsangaben
- numeric = numerische Attribute
- nominal = nominale Attribute

Fragen mit gegebenen Antwortmöglichkeiten (in geschweifter Klammer), wie die Häufigkeit der Übungsbesuche, wurden nominal kodiert. Die Antwort die angekreuzt wurde wird in die Instanz mit aufgenommen. Nominale Attribute die nur zwei Werte zur Auswahl haben nennt man binär. Die Frage nach Gründen einer grundsätzlichen Veränderung der privaten Situation kann, durch ein leeres Textfeld im Fragebogen, beantwortet werden oder nicht und wird demnach als ja oder nein in der Instanz beschrieben. Fragen die als Antwort einen Zahlenwert mit natürlicher Ordnung zulassen, wie die Anzahl der bestandenen Prüfungen, wurden als numerisch deklariert. In diesem Fall bestand der gesuchte Wert aus der Addition angekreuzter Felder.

Die Menge aller Instanzen, die mit einer gemeinsamen Menge von Attributen beschrieben werden bilden einen Datensatz (Relation). Die Klasse (class) ist das Attribut einer Instanz, auf die, aufgrund der vorhandenen Informationen, geschlossen werden soll und ist in WEKA das zuletzt aufgeführte Attribut. Unsere Klasse ist das Attribut „beratung“, welche sich in `beratung=ja` (Einladung zum Einzelgespräch) und `beratung=nein` (Einladung zum Vortrag) unterteilt. Von den 123 Studierenden, deren Daten zur Weiterverarbeitung vorliegen, wurden 35, noch händisch, der Klasse `beratung=ja` zugeteilt und 88 zu der Klasse `beratung=nein`.

Eine arff-Datei kann jeweils nur eine Relation enthalten. Die Zeilen, die mit einem @ (leitet ein Schlüsselwort ein) beginnen, nennt man den Kopf (header) der Datei. Nach dem Kopf schließt der Körper (body) an. Eine arff-Datei beginnt immer mit „@relation Relationenname“, dann folgen die Deklarationen der Attribute, wobei jedes Attribut mit einer neuen Zeile beginnt, mit „@attribute Attributname Attributtyp“. Bei nominalen Attributen, werden die möglichen nominalen Werte in geschweifte Klammern gesetzt. der Header endet mit dem Schlüsselwort „@data“. In einer neuen Zeile beginnend folgt der Body, die eigentlichen Daten, wobei jede Instanz mit ihren

Attributen, durch Kommas getrennt, eine Zeile ausfüllt. Die Reihenfolge der Attributwerte muß mit der Reihenfolge in der Deklaration der Attribute übereinstimmen. Leerzeichen werden durch Unterstriche ersetzt und fehlende Werte mit einem Fragezeichen.

Fehlende Werte sind zwangsläufig dadurch entstanden, da sich die verwendeten Daten aus einem Wintersemester und einem Sommersemester zusammensetzen. Während die Veranstaltung „Grundlagen der Informatik“ (GDI), für Erstsemestler, zu jedem Semesterbeginn angeboten wird, unterscheiden sich die weiteren angebotenen Veranstaltungen vom Wintersemester zum Sommersemester. Wenn die Studierenden zum Wintersemester ihr Studium beginnen, können sie neben GDI noch die Veranstaltung „Technische Grundlagen der Informatik“ (TGDI) sowie „Mathematik“ (mathe) besuchen und darüber dann Angaben machen, die in die Instanz mit einfließen. Da in der arff-Datei, als Attribute, alle möglichen Veranstaltungen aufgelistet sind und, wie oben schon erwähnt, die Reihenfolge dieser in den Instanzen einbehalten werden muss, wird hier bei den Fragen zu den Veranstaltungen des Sommersemesters ein Fragezeichen gesetzt.

4.3 Klassifizierung

Die Umsetzung von Maschinellern Lernen kann durch verschiedene Algorithmen erfolgen, wobei man zwischen überwachtem Lernen und unüberwachtem Lernen unterscheidet. Bei unüberwachtem Lernen gibt es zwar eine Menge von Beispielen, die aber keine Zuordnung zu einer Klasse besitzen. Hier gibt es die Möglichkeit einer Clusteranalyse, um Instanzen, die sich ähneln, in Gruppen einzuteilen. Ein weiterer Bereich des unüberwachten Lernens stellt die Assoziationsanalyse dar, die Abhängigkeiten zwischen den Attributen ermittelt. Im Gegensatz zum unüberwachten Lernen gibt es beim überwachten Lernen auch eine Menge an Beispielen, deren Zuordnung zu einer Klasse jedoch bekannt ist. Man spricht hier von einer Klassifizierung, wenn bei der Menge von Beispielen jede Instanz einer Klasse zugeordnet ist und neue, noch unbekannte Instanzen anhand dieser Beispiele, klassifiziert werden. Da (in unserem Fall) die Beispieldaten schon zu ihrer Klasse zugeordnet sind und die Aufgabe darin besteht, nachfolgende Daten in ihre korrekte Klasse einordnen zu können, verwenden wir die Klassifizierung.

Dabei wird ein ausgewählter Lernalgorithmus, zunächst mit den gegebenen Daten trainiert, einer so genannten Trainingsmenge, und anschließend, für eine Genauigkeitsabschätzung, mit einer Testmenge getestet. Eine bereits klassifizierte Instanzmenge muss demnach in eine Trainings- und eine Testmenge aufgeteilt werden. Die Trainingsmenge besteht aus einer Anzahl von Instanzen des Datensatzes, für welche bereits die zugehörige Klasse bekannt ist. Mit Hilfe dieser Daten wird ein ausgewähltes Klassifikationsverfahren „trainiert“. Während dieser Phase des Lernens wird ein Modell konstruiert, welches den Zusammenhang zwischen den Attributwerten und ihrer Klassenzugehörigkeit beschreibt. Mit der Testmenge wird nun dieses Modell getestet, in wie weit es Vorhersagen zutreffend liefert. Von den Testdaten müssen natürlich auch die jeweiligen Klassen bekannt sein, um sie mit den vorhergesagten vergleichen zu können. Die Aufteilung der Menge kann durch zwei Arten erfolgen. Zum einen kann man den Anteil der Trainings- und Testmenge prozentual eingeben (percentage split), wobei man für die Trainingsmenge, zum Beispiel den in WEKA standardmäßig vor eingestellten Wert, 66% und für die Testmenge den Rest nimmt. Zum anderen kann auch eine Kreuz-Validierung (cross-validation) durchgeführt werden. Dabei wird die Datenmenge in x -gleichgroße Teile (folds) zerlegt. Das $(x-1)$ -fache der Teilmenge wird als Trainingsmenge genutzt und der Rest als Testmenge. Hier wird oftmals der Wert $x=10$ verwendet.

Es gibt mehrere verschiedenartige Klassifikationsmodelle. Um eine Zuordnung zu ermöglichen wird aus der Menge der Trainingsdaten eine Entscheidungsfunktion oder Regeln erlernt. Einige der möglichen Modelle:

- Support Vector Machines (SVM): Die Instanzen werden, mittels ihrer Attribute, in einem Raum dargestellt, in dem ähnliche Instanzen beieinander liegen. Es wird eine Ebene konstruiert, wobei die eine Seite die Klasse „ja“ und die andere „nein“ ist. Das zu klassifizierende Objekt wird mit den (Support-) Vektoren, die an der Grenze der Klassen liegen, verglichen und demnach in ihre Klasse eingeordnet.
- Entscheidungsbaum: ist Graph mit einer Baumstruktur. Bei jedem Knoten wird ein bestimmtes Attribut einer zu klassifizierenden Instanz auf die Erfüllung einer Bedingung getestet. Dementsprechend verläuft der Pfad bis hin zum Blatt des Baumes, welches die Klassifikation angibt.

- Regeln: Eine Regel drückt einen kausalen Zusammenhang in einem Datensatz aus. Sie besteht aus einer Menge an Bedingungen, die erfüllt werden müssen und der zugehörigen Klasse, wenn die Bedingungen zutreffen. Mehrere Regeln bilden eine Entscheidungsliste, bei der die Reihenfolge der Regeln wichtig ist. Sie funktioniert wie ein Sieb. Eine zu klassifizierende Instanz durchläuft die Entscheidungsliste von oben nach unten. Wenn sie von einer Regel abgedeckt wird, wird sie aus der Instanzmenge entnommen. Wenn keine der Regeln auf eine Instanz zutrifft, tritt die zuletzt aufgeführte Standard-Regel in Kraft, die alle bisher nicht abgedeckten Instanzen einer Klasse zuordnet.

Im Allgemeinen sind Regeln recht einfach zu verstehen, da sie logische Bedingungen mit sich tragen, nur jeweils die Zuordnung zu einer Klasse betrachten und somit weniger komplex sind wie beispielsweise Entscheidungsbäume.

Im Weiteren werde ich nur noch auf die Klassifizierer eingehen, die anhand von Regeln lernen. Um das beste Modell zu finden, welches den Datensatz am besten beschreibt, müssen verschiedene Regel-Lerner ausprobiert werden. WEKA stellt hiervon einige zur Verfügung, wie z.B. JRip, Ridor oder Part.

4.4 Auswertung

Verwendet man in WEKA JRip, mit einer 10-Fold-Kreuz-Validierung, erhält man zunächst nachfolgende Entscheidungsliste als Ausgangsbasis:

```
(prüfungen_bestanden <= 0) and (prüfungen_mitgeschrieben <= 0) => beratung = ja (28.0/12.0)
=> beratung = nein (95.0/19.0)
```

Wenn eine Instanz die Bedingung der ersten Regel erfüllt, wird sie der Klasse beratung=ja zugeordnet, ansonsten tritt die nächste Regel in Kraft, was in diesem Fall, sofort die Einordnung in die Klasse beratung=nein zur Folge hat. Die Werte in Klammern besagen, dass während der Trainingsphase 28 der 123 Instanzen mit der ersten Regel abgedeckt wurden, wovon bei 12 der 28 Instanzen, zwar die Bedingungen erfüllt wurde, sie aber tatsächlich in die andere Klasse gehören.

Nachdem der Klassifizierer die Regeln gelernt hat, werden diese, anhand der Testmenge, getestet. Die Auswertung dieses Testverfahrens beschreibt WEKA zum einen in der Klassifikationsrate (correctly classified instances), die den prozentualen Wert der Anzahl korrekt klassifizierter Instanzen, gemessen an der Gesamtmenge der getesteten Instanzen, angibt. Eine zu testende Instanz durchläuft die Entscheidungsliste, wird demnach in eine Klasse eingeteilt und wird schließlich mit ihrer tatsächlich, zugeordneten Klasse verglichen. Wenn die vorhergesagte Klasse der richtig, bekannten Klasse entspricht, ist diese Instanz korrekt klassifiziert. Die Fehlerrate (incorrectly classified instances) drückt den Anteil falsch klassifizierter Instanzen an der Gesamttestmenge aus, die Instanzen deren vorhergesagten Klassen mit den tatsächlichen nicht übereinstimmen. Das Klassifikationsverfahren wird demnach als „gut“ empfunden, wenn es soviel wie möglich korrekt klassifizierte Instanzen mit sich trägt. Zum anderen zeigt WEKA bei jeder Ausführung einer Klassifizierung die jeweilige Konfusionsmatrix (confusion matrix) des Testergebnisses auf. Die Konfusionsmatrix gibt genau Aufschluss darüber, wie viele der Instanzen korrekt und nicht korrekt klassifiziert wurden und das für jede Klasse.

Die Klassifikationsrate ergibt, in unserem Fall, bei der Verwendung von JRip und einer 10-fachen-Kreuzvalidierung, 62,6016%. 77 der 123 Instanzen wurden korrekt klassifiziert. Einen detaillierten Überblick darüber liefert die nachfolgende Konfusionsmatrix:

	positiv vorhergesagt	negativ vorhergesagt
tatsächlich positiv	7 (TP)	28 (FN)
tatsächlich negativ	18 (FP)	70 (TN)

Tabelle 4.1: JRip – Konfusionsmatrix

7 der 123 Instanzen wurden korrekterweise der Klasse `beratung=ja` zugeordnet und 70 Instanzen richtigerweise der Klasse `beratung=nein`. 18 Instanzen wurden in die Klasse `beratung=ja` eingeteilt, obwohl sie tatsächlich in die andere Gruppe gehören und 28 Instanzen wurden irrtümlicherweise der Klasse `beratung=nein` zugewiesen.

Hier spielt es nicht nur eine Rolle, dass die Zahlen der „wahren“ Werte möglichst groß sind, wichtig ist auch, dass der False-Negative-Wert so klein wie möglich ist. Wenn Studenten fälschlicherweise in die Klasse `beratung=ja` geraten (False-Positive) und deshalb zu einem Fachgespräch eingeladen werden ist das weniger problematisch wie das „Nicht-Einladen“ für einen Studenten, der die Beratung eigentlich nötig hat (False-Negative).

Die oben genannte Regelmenge hat den Vorteil, dass sie sehr allgemein gehalten ist und dadurch eine hohe Abdeckung der Beispiele mit sich bringt. Der Vorgang mit dem Ziel, die Regelmenge möglichst allgemein zu halten, geschieht durch bestimmte Abbruchbedingungen, nach denen die Regelmenge oder einzelne Regeln „beschnitten“ werden, das so genannte „pruning“. Pruning wird oftmals eingesetzt um zu vermeiden, dass sich der Algorithmus zu sehr an die Trainingsmenge anpasst (Overfitting). Wenn es, zum Beispiel, für jede Instanz der Trainingsmenge genau eine Regel gibt, ist das eine exakte Beschreibung der Daten. Wenn jetzt aber ein neues Beispiel klassifiziert werden soll, das es so nicht in der Trainingsmenge gibt, versagt der Klassifizierer.

Der Nachteil beim Pruning ist allerdings, dass die Regeln, eben durch die starke Verallgemeinerung, nicht sehr genau sind und im unserem Fall nur anhand von zwei Attributen klassifizieren. JRip ermöglicht beide Optionen, bei der Modellierung einer Regelmenge: mit Pruning oder ohne Pruning. Letzteres liefert dieses komplexere Regelmodell:

```
(prüfungen_bestanden <= 0) and (prüfungen_angemeldet <= 1) and (studienaufwand_selbststudium = 5-10_h) => beratung=ja (5.0/0.0)
```

```
(prüfungen_bestanden <= 0) and (stundenprotag_computer = 1/2-1_h) and (mathe1_vorlesung_besucht = fast_immer) => beratung=ja (4.0/0.0)
```

```
(prüfungen_bestanden <= 0) and (prüfungen_zugelassen <= 3) and (lernstoff_zu_viel = trifft_weniger_zu) => beratung=ja (5.0/0.0)
```

```
(prüfungen_bestanden <= 0) and (private_gründe = ja) and (lerngruppe = nein) and (gdi1_übung_besucht = unregelmäßig) => beratung=ja (3.0/0.0)
```

```
(prüfungen_bestanden <= 0) and (mathenote = 12-10) and (studienaufwand_selbststudium = 16-20_h) => beratung=ja (3.0/0.0)
```

```
(prüfungen_bestanden <= 0) and (studienaufwand_lehrveranstaltung = weniger_als_5_h) and (lerngruppe = nein) => beratung=ja (3.0/0.0)
```

```
(zu_nervös_in_klausur = trifft_voll_zu) and (abiturnote = 2.6-3.0) and (gdi1_hausübung_gemacht = fast_immer) => beratung=ja (3.0/0.0)
```

(prüfungen_mitgeschrieben <= 0) and (studienaufwand_lehrveranstaltung = 16-20_h) and (stundenprotag_fern = 1/2-1_h) => beratung=ja (2.0/0.0)

(prüfungen_bestanden <= 0) and (zu_wenig_zeit_investiert = trifft_kaum_zu) and (stundenprowoche_freunde = 5-10_h) => beratung=ja (2.0/0.0)

(stundenprowoche_job = 5-10_h) and (abiturnote = 2.1-2.5) and (mathenote = 12-10) => beratung=ja (2.0/0.0)

(tgdi_hausübung_gemacht = k.A.) and (mathe1_hausübung_gemacht = häufig) and (prüfungen_entschuldigt >= 1) => beratung=ja (2.0/0.0)

=> beratung=nein (89.0/1.0)

Schon die Werte in Klammern deuten darauf hin, dass die Regeln die Trainingsmenge sehr gut beschreiben, da es keine Fehlklassifikation gibt. Dies trägt auch dazu bei, dass sich die Klassifikationsrate, bei diesem Modell, auf 67,4797%, im Vergleich zur Regelmenge mit Pruning, erhöht hat. Durch die Konfusionsmatrix ist weiterhin zu erkennen, dass die Anzahl der richtig klassifizierten, positiven Instanzen von 7 auf 13 gestiegen ist und gleichermaßen die Anzahl, derjenigen die tatsächlich zur Klasse beratung=ja gehören, aber zur Klasse beratung=nein falsch klassifiziert wurden, von 28 auf 22, gesunken ist.

Dieses Regelwerk scheint offenkundig für diesen Datensatz die bessere Option zu sein.

Bei meinen Betrachtungen habe ich noch weitere Regel-Lerner, unter anderem Part und Ridor, betrachtet. Durch das vielseitige Ausprobieren geeigneter Optionen zur Optimierung der jeweiligen Regelmengen dieser Klassifizierer, ergaben sich auch hier noch weitere interessante Modelle, die in Betracht gezogen werden können. Der nachfolgenden Tabelle ist neben dem jeweiligen Klassifizierer noch die Anzahl der erlernten Regeln und der dazugehörigen Klassifikationsrate zu entnehmen. Bei der Modellierung der Regelmengen wurde jeweils eine 10-Fold-Kreuzvalidierung durchgeführt.

Klassifizierer	Anzahl der Regeln	Klassifikationsrate
JRip mit Pruning	2	62,6016%
JRip ohne Pruning	12	67,4797%
Part mit Pruning	11	68,2927%
Part ohne Pruning	24	57,7236%
Ridor	5	67,4797%

Tabelle 4.1: Vergleich betrachteter Klassifizierer

Auch wenn die Klassifizierer Part und Ridor eine mindestens ebenso hohe Klassifikationsrate aufweisen wie JRip, gab es bei diesen, im Gegensatz, aber einige Fehlklassifikationen bei der Trainingsphase.

5. Ausblick

Die Durchführung des Mentorensystems im Fachbereich Informatik an der TU Darmstadt und speziell die Ausdehnung in das zweite Fachsemester, mit der Möglichkeit einen Teil der Studierenden, die im ersten Fachsemester einen geringen Prüfungserfolg hatten, zu einem Einzelgespräch einzuladen, ist noch relativ neu. Diese Information spiegelt sich in der noch geringen Anzahl an verfügbaren Daten wieder. Für die Aufgabe einer geeigneten Regelfindung standen lediglich die Daten der Studierenden von den letzten zwei Semestern zur Verfügung. Die weitere Aufgabe wird sein, die Regeln auf die neu erhobenen Daten der nächsten Semester anzuwenden um deren Genauigkeit zu testen. Ein weiterer Schritt wird darin bestehen, die neuen Daten in den vorhandenen Datensatz mit einzupflegen um zu vergleichen ob sich Regeländerungen ergeben oder die gefundenen Regeln bestehen bleiben. Dadurch wird sich zeigen, ob es möglich ist, anhand von diesen Regeln, bestimmte Typen zu erkennen um dadurch gezielt Verbesserungsmaßnahmen, in Bezug auf den Prüfungserfolg, ergreifen zu können. Eine weitere Überlegung, zur Entwicklung dieser Klassifizierung, ist das Hinzufügen neuer Attribute in den Datensatz, wie zum Beispiel die erreichte Punkteanzahl bei Klausuren, Zwischenklausurergebnisse oder Hausübungspunkte, was eine neue Anforderung darstellt.

6 Anhang

Fragebogen 1. Semester

Weiter

(*)Elemente sind erforderlich

1. Allgemeine Angaben

1 Name:*

2 Vorname:*

3 Matrikelnummer:*

Wohnsituation

4 Mein Anfahrtsweg zur Uni dauert einfach:*

5 Ich wohne während des Semesters:*

Nicht ausgewählt bei meinen Eltern im eigenen Haushalt (WG, allein, etc.)

Schule

6 In welchem Bereich liegt Ihre Abiturnote?*

7 In welchem Bereich liegt Ihre letzte Mathematiknote?*

Tätigkeiten vor Beginn des Studiums

8 Was haben Sie vor Beginn des Informatikstudiums gemacht? (Mehrfachnennungen möglich)*

Bund Zivildienst freiwilliges soziales Jahr Reisen Ausbildung/Lehre Jobben anderes Studium nichts Bestimmtes
 Studienkolleg k.A.

2. Allgemeine Angaben zur Prüfungsanmeldung

9 Zu welchen Prüfungen waren Sie zugelassen?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

10 Für welche Prüfungen waren Sie angemeldet?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

11 Welche Prüfungen haben Sie geschrieben?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

12 Von welchen Prüfungen haben Sie sich fristgerecht abgemeldet?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

13 Bei welchen Prüfungen waren Sie kurzfristig entschuldigt (z.B. Attest)?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

14 Welche Prüfungen haben Sie bestanden?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

15 Zu welchen Klausuren sind Sie in die Klausureinsicht gegangen?*

GdI 1 FGdI 1 FGdI 2 HCS sonstige keine

3. Angaben zur Prüfungsvorbereitung

Wie würden Sie Ihre Prüfungsvorbereitung selbst einschätzen? (bitte entsprechende Aussage ankreuzen)

16 GdI 1: Wie würden Sie Ihre Prüfungsvorbereitung selbst einschätzen?*

Nicht ausgewählt sehr gut gut weniger gut schlecht k.A.

17 GdI 1: Wie häufig haben Sie die Übungen besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

18 GdI 1: Wie häufig haben Sie die Hausübungen gemacht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

19 GdI 1: Haben Sie die Vorlesung regelmäßig besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

20 FGdI: Wie würden Sie Ihre Prüfungsvorbereitung selbst einschätzen?*

Nicht ausgewählt sehr gut gut weniger gut schlecht k.A.

21 FGdI: Wie häufig haben Sie die Übungen besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

22 FGdI: Wie häufig haben Sie die Hausübungen gemacht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

23 FGdI: Haben Sie die Vorlesung regelmäßig besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

24 HCS: Wie würden Sie Ihre Prüfungsvorbereitung selbst einschätzen?*

Nicht ausgewählt sehr gut gut weniger gut schlecht k.A.

25 HCS: Wie häufig haben Sie die Übungen besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

26 HCS: Wie häufig haben Sie die Hausübungen gemacht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

27 HCS: Haben Sie die Vorlesung regelmäßig besucht?*

Nicht ausgewählt fast immer häufig unregelmäßig selten k.A.

4. Angaben zum Studienverhalten

Wie hoch schätzen Sie Ihren Studienaufwand ein? (Angabe durchschnittlicher Stunden pro Woche)

28 Für Lehrveranstaltungsbesuche (Übungen, Vorlesungen, Sprechstunden):*

29 Selbststudium:*

30 Haben Sie eine Lerngruppe?*

Nicht ausgewählt ja nein

31 Lernen Sie bevorzugt in Gruppen oder allein?*

Nicht ausgewählt eher in Lerngruppen

	<input type="radio"/> eher allein
32	Verwenden Sie folgende Hilfsmittel zur Prüfungsvorbereitung? [*] <input type="checkbox"/> Karteikarten <input type="checkbox"/> Lerntagebuch <input type="checkbox"/> Lernplan <input type="checkbox"/> Lesemethoden (z.B. SQ3R) <input type="checkbox"/> empfohlene Lehrbücher <input type="checkbox"/> selbstrecherchierte Lehrbücher
33	Gehen Sie neben dem Studium einer Erwerbstätigkeit (Jobben) nach? [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> ja <input type="radio"/> nein
34	Wenn ja, wie viele Stunden pro Woche?
35	Wie viel Zeit verbringen Sie mit Ihren Hobbies (pro Woche)? [*]
36	Wie viel Zeit verbringen Sie mit Ihren Freunden und Ihrer Familie (pro Woche)? [*]
37	Wie viele Stunden verbringen Sie durchschnittlich pro Tag mit Computer spielen? [*]
38	Wie viele Stunden verbringen Sie durchschnittlich pro Tag mit der Pflege von sozialen Onlinenetzwerken? [*]
39	Wie viele Stunden sehen Sie durchschnittlich pro Tag fern (Serien, Filme,...)? [*]
5. Angaben zu Gründen des Prüfungsmisserfolgs	
Laut Ihrem Studienplan des BSc Informatik sollten Sie 4 Prüfungen im ersten Semester erfolgreich abgelegt haben (GdI I, FGdI 1 + 2, HCS). Wenn dies auf Sie nicht zutrifft, welche Gründe sehen Sie dafür? (Bitte für alle Gründe die entsprechende Ausprägung auf der Skala ankreuzen.)	
40	Der Lernstoff war zu schwer. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
41	Der Lernstoff war zu viel. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
42	Der Lernstoff war anders als erwartet. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
43	Ich habe zu wenig Zeit investiert. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
44	Ich habe mir zu viel vorgenommen. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
45	Ich hatte keinen Lernplan. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
46	Klausurinhalte waren anders als erwartet. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
47	Ich war zu nervös in der Klausur. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
48	Die Lehrveranstaltung hat nicht auf die Klausur vorbereitet. [*] <input type="radio"/> Nicht ausgewählt <input type="radio"/> trifft voll zu <input type="radio"/> trifft weniger zu <input type="radio"/> trifft kaum zu <input type="radio"/> trifft nicht zu
49	Spezielle Gründe für Prüfungsmisserfolge bei GdI 1: [*] <div style="border: 1px solid black; height: 40px; width: 100%;"></div>
50	Spezielle Gründe für Prüfungsmisserfolge bei FGdI: [*] <div style="border: 1px solid black; height: 40px; width: 100%;"></div>
51	Spezielle Gründe für Prüfungsmisserfolge bei HCS: [*] <div style="border: 1px solid black; height: 40px; width: 100%;"></div>
6. Private Situation	
Die Übergangsphase Schule-Hochschule bedeutet häufig auch eine deutliche private Veränderung. Wenn dies der Fall ist, würden wir uns freuen, wenn Sie freiwillig Angaben dazu machen würden.	
52	Private Gründe: <div style="border: 1px solid black; height: 40px; width: 100%;"></div>
<input type="button" value="Weiter"/>	

7 Quellen

- [1] <http://www.tu-darmstadt.de> (Daten und Fakten aus 2010 bis 2011)
- [2] <http://www.informatik.tu-darmstadt.de/>
- [3] <http://www.hda.tu-darmstadt.de>
- [4] <http://www.ft-informatik.de/>
- [5] <https://moodle.tu-darmstadt.de/>
- [6] http://de.wikipedia.org/wiki/Waikato_Environment_for_Knowledge_Analysis
- [7] http://en.wikipedia.org/wiki/Tom_M._Mitchell
<http://www.ke.tu-darmstadt.de/lehre/archiv/ws1011/mldm/introduction.pdf>
http://de.wikipedia.org/wiki/Maschinelles_Lernen
http://en.wikipedia.org/wiki/Machine_learning
<http://www.cs.waikato.ac.nz/ml/weka/arff.html>