



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Statistische Prognosen zukünftigen Studienerfolges

Oliver Steenbuck

Projekt 1: Statistische Prognosen zukünftigen
Studienerfolges

Inhaltsverzeichnis

Tabellenverzeichnis	2
Abbildungsverzeichnis	3
Listings	3
1. Einleitung	3
2. StISys	3
2.1. Probleme mit dem Export aus StISys	4
3. Data Mining und Wissensfindung in Datenbanken	4
3.1. Hintergrundwissen und Zielsetzung	5
3.2. Datenauswahl	6
3.3. Datenbereinigung	6
3.4. Modellfunktionalität	6
3.5. Verfahrensauswahl	7
3.6. Data Mining	8
4. Technischer Aufbau	9
4.1. Datenbank	10
4.2. Datenverdichtung	10
4.2.1. Quelldatensätze	10
4.2.2. Erzeugung Studentendatensätze	12
4.2.3. Erzeugung Prüfungsdatsätze	12
4.2.4. Erzeugung Kursdatensätze	12
5. Ausblick	13
Literatur	13
A. Listings	14
A.1. Parser	14
A.2. Datenbankschema	16

Tabellenverzeichnis

1. Regression Programmieren 1 und Durchschnittsnote	8
---	---

Abbildungsverzeichnis

1. Alter bei Immatrikulierung zu Durchschnittsnote (Regressionsgerade in Rot)	8
2. Technische Architektur/Datenfluss	9
3. Fachliches Datenmodell	10

Listings

1. Modell Fit in R	7
2. Antlr Grammatik	14
3. Database Creation SQL	16

1. Einleitung

In diesem Projektbericht werden Methoden und Verfahren beschrieben, um Studienverlaufs- und Notendaten von Studierenden der Hochschule für Angewandte Wissenschaften Hamburg (HAW) auszuwerten, sowie erste Ergebnisse präsentiert. Langfristiges Ziel ist statistische Aussagen über zukünftige Studienerfolge ableiten zu können. Grundlage dieser Untersuchung sind ca. 32600 Prüfungsereignisse, die von 1038 Studierenden des Studienganges Technische Informatik von 2004 bis 2012 erbracht wurden. Die Daten stammen aus dem Studierenden Informations System¹, das an der HAW Hamburg verwendet wird.

Der Projektbericht ist wie folgt aufgebaut, es wird zuerst ein Überblick über den Prozess des Knowledge Discovery in Databases (KDD) gegeben und dann der technische Aufbau im Rahmen des Projektes beschrieben. Abschließend findet ein Ausblick auf weitere geplante Arbeiten in diesem Umfeld statt.

2. StISys

Das Studierenden Informations System des Departments Elektrotechnik und Informatik, in dem der Studiengang Technische Informatik angesiedelt ist, dient der Verwaltung von Studienleistungen sowie der Anmeldung zu Prüfungen, Seminaren und Praktika. Auf StISys kann sowohl von den Studierenden zugegriffen werden, die dort ihre Noten abfragen können, als auch von der Verwaltung, um Noten einzutragen bzw. Zeugnisse und Leistungsnachweise (BAföG, Zeugnis über das 1. Studienjahr) zu erstellen.

¹im Folgenden StISys

In StISys sind die Ergebnisse aller Prüfungen vermerkt, an denen ein Student im Lauf seines Studiums teilgenommen hat. Neben dieser elektronischen Verwaltung wird für jeden Studierenden eine physikalische Prüfungsakte geführt, in der Gutachten über Prüfungsleistungen verwahrt werden. Die Noten finden sich in jedem Fall in StISys, welches der Fakultät außerdem zur Plausibilitätskontrolle bei Prüfungsanmeldungen (Vorleistungen erbracht etc.) dient.

2.1. Probleme mit dem Export aus StISys

Probleme insbesondere für diese Arbeit können dadurch entstehen, dass StISys nur einen Ist-Zustand und keine Historie verwaltet.

Wenn ein Studierender beispielsweise zuerst ein Studium der Elektrotechnik aufnimmt und dann nach 2 Semestern in die Technische Informatik wechselt, wird er in StISys als TI-Student geführt. Die in Elektrotechnik bereits abgelegten Prüfungen werden jedoch weiterhin an seinem Datensatz gepflegt, ohne dass es einen expliziten Marker gibt, der beschreibt, dass diese Prüfungen in einen anderen Studiengang gehören. In den vorliegenden Daten tritt dieser Fall in deutlich weniger als 1% der Fälle ($n = 23$) auf.

Analog zu diesem messbaren Rauschen in den Daten kann es Fälle geben, in denen Veranstaltungen, die in anderen Studiengängen und der Technischen Informatik mit den gleichen Fachkürzeln identifiziert sind, hier fehlerhaft zum Modul der TI werden. Hier wird davon ausgegangen, dass diese Fälle von der Größe her ähnlich sind und in erster Näherung vernachlässigt werden können.

Ähnliches gilt für Studenten mit Abschluss. Es wird in StISys nicht explizit geführt, dass ein Student erfolgreich abgeschlossen hat, diese Information ergibt sich nur aus der Gesamtschau auf seine bestandenen Prüfungen. Sollte sich im späteren Verlauf herausstellen, dass diese Informationen explizit benötigt werden, wäre ein weiterer Datenimport aus den Verwaltungssystemen der Hochschule², in denen diese Daten geführt werden, notwendig.

3. Data Mining und Wissensfindung in Datenbanken

Data Mining und Wissensfindung in Datenbanken (Knowledge Discovery in Databases)³ besteht aus den folgenden 7 Schritten:

1. Hintergrundwissen und Zielsetzung

²Helios ([HAW - elektronisches Informations- und Organisationssystem)

³KDD

2. Datenauswahl
3. Datenbereinigung
4. Modellfunktionalität
5. Verfahrensauswahl
6. Data Mining
7. Interpretation (hier noch nicht durchgeführt)

(vgl. [Beierle und Kern-Isberner, 2008](#), S. 153 ff.)

3.1. Hintergrundwissen und Zielsetzung

Zielsetzung ist es hier, zu zeigen, dass Analysen wie in [Golding und Donaldson \(2006\)](#) und [Borrego u. a. \(2005\)](#) mit den vorhandenen Daten und technischen Mitteln prinzipiell durchgeführt werden können. Hierzu wurde neben den in Kapitel 4.1 beschriebenen Daten auch Hintergrundwissen über die Struktur und Zielsetzung der Prüfungsordnungen 2004 und 2008⁴ verwendet. So wurde von einer Inhaltsgleichheit der Kurse aus den beiden Prüfungsordnungen ausgegangen, wenn diese von der Fakultät in der entsprechenden Äquivalenzliste⁵ festgestellt wurde. Makroskopisch ist der Studiengang Technische Informatik als 3-jähriger Bachelor mit 30 CP pro Semester geplant, die in jeweils 5 Modulen, die mit einer Klausur abschließen, erbracht werden. Details zum Inhalt der Module können dem Modulhandbuch⁶ der Technischen Informatik entnommen werden.

Im Speziellen ist dieses zusätzliche Wissen notwendig, um die Daten aus dem StlSys-Export zu bereinigen und zu normalisieren (siehe 3.3).

Einige Faktoren von Rauschen in den Daten können hier nicht beseitigt werden. So können z.B. Studenten ein Urlaubssemester nehmen, das nicht als Hochschulsesemester zählt, diese Information ist in StlSys nicht geführt, sodass das Semester hier als eines ohne Leistung gewertet werden würde. Laut dem administrativ zuständigen Fakultätsservicebüro ist dieser Fall (Urlaubssemester) selten⁷.

⁴<http://www.haw-hamburg.de/fakultaeten-und-departments/ti/ti-fsb/pruefungsordnungen.html>

⁵<http://www.haw-hamburg.de/fakultaeten-und-departments/ti/ti-fsb/pruefungsordnungen.html>

⁶<http://www.haw-hamburg.de/en/ti-i/studium/technische-informatik-ba/modulhandbuecher.html>

⁷In einer zukünftigen Arbeit ist dieser Fall statistisch genauer abzugrenzen, hierzu sind weitere Auswertungen notwendig.

3.2. Datenauswahl

Datenbasis der geplanten Analysen sind insgesamt 32604 Datensätze vom Sommersemester 2004 bis zum Wintersemester 2012 aus dem Studiengang Technische Informatik der HAW Hamburg. Wobei zu diesem Zeitpunkt entsprechend der Zielsetzung nur die Studierenden mit erfolgreichem Abschluss ausgewertet wurden.

Jeder Datensatz entspricht einer benoteten bzw. unbenoteten Prüfung, wobei durch die Prüfungsordnungen Abhängigkeiten zwischen diesen definiert werden. Die benoteten Prüfungen werden auf einer Skala von 00 bis 15 Punkten bewertet, wobei höhere Punktzahlen besser sind und die Noten 00-04 einer nicht bestandenen Prüfung entsprechen. Kapitel 4.2.1 beschreibt das Format, in dem die Quelldaten vorliegen.

3.3. Datenbereinigung

Im Zuge der Datenbereinigung wurden im Speziellen die Studenten mit Immatrikulation im Wintersemester 2004 aussortiert da aufgrund der von der Fakultät kommunizierten Plausibilitätskriterien (ca. 50 Immatrikulationen pro Semester) angenommen werden kann, dass hier Daten der vorherigen Semester eingemischt wurden. Laut Datenbank sind 395 Studierende im WS 2004 immatrikuliert worden, im Gegensatz zu ca. 46 im Durchschnitt der anderen 14 Semester. Weiterhin werden in dieser Projektarbeit nur die Studenten betrachtet, die bereits einen Abschluss erworben haben. Sowie bei diesen nur die jeweils (letzte) erfolgreiche Klausur in einem Modul, eine solche muss unter der Voraussetzung, dass der Student einen Abschluss besitzt, für jedes Modul vorliegen.

Im Parserschritt wurden ebenfalls einige Probleme mit der Datenqualität bereinigt, die auf unterschiedliche Eingabekonventionen sowie die mehrmalige Neuanlage von Modulen zurückzuführen sind⁸.

3.4. Modellfunktionalität

Die im Rahmen des Projektes erstellte technische Infrastruktur orientiert sich in ihrer Zielsetzung an statistischen Analysen aus veröffentlichten Arbeiten wie beispielsweise [Golding und Donaldson \(2006\)](#) und [Borrego u. a. \(2005\)](#) und soll es ermöglichen, diese auf dem vorhandenen Datenbestand durchzuführen.

Fachlich handelt es sich hierbei beispielhaft um Thesen wie die untenstehenden aus [Golding und Donaldson \(2006\)](#).

⁸unterschiedliche IDs, leicht unterschiedliche Schreibweisen für einen Kurs in verschiedenen Semestern

- H_{02} : Performance in 1st year programming and computer science courses does not have an impact on student's performance : rejected
- H_{03} : Gender and age does not determine student's level of success in computer science : accepted

Statistisch werden die in 3.5 angeführten Verfahren verwendet.

3.5. Verfahrensauswahl

Die im Rahmen der Projektarbeit prototypisch durchgeführten statistischen Verfahren sind (siehe auch [Papula \(2008\)](#)):

Analysen des Pearson-Korrelationskoeffizienten zur Bestimmung des Zusammenhanges kontinuierlicher bzw. intervallskalierteter Merkmale (Notenpunkte, Alter).

Chi-Quadrat-Homogenitätstest und t-Test zur Analyse des Zusammenhanges zwischen nicht kontinuierlich ausgeprägten Merkmalen (z.B. Geschlecht und Abschlussquote).

Linear Regression, um die Beziehung zwischen einzelnen Veranstaltungen und dem Notendurchschnitt zu untersuchen.

Listing 1 zeigt beispielhaft, wie ein Modell Fit zur Regressionsanalyse in R durchgeführt wird. Diese Analyse führt zu Ergebnissen, wie sie in Tabelle 1 dargestellt sind, und ist prinzipiell geeignet, veröffentlichte Ergebnisse auf den Datenbestand der HAW anzuwenden bzw. wird in Veröffentlichungen genau diese Methodik verwendet ([Golding und Donaldson \(2006\)](#))⁹

Modell Fit in R

```
1 pr1 <- read.csv("../pr1.dat")
2 pr1GpaFit <- lm(pr1$gpa ~ pr1$pr1)
3 summary(pr1GpaFit)
```

Im hier gezeigten Prozess werden die Daten in Ruby aus der Datenbank ausgelesen und für die weitere Verarbeitung durch R in komprimierter Form als csv-Datei abgelegt. Prinzipiell wäre auch ein Einlesen der Daten direkt in R möglich, hier wurde der Umweg über Ruby gewählt, da geplante erweiterte grafische Auswertungen den/die Datenzugriff/Modellierung in Ruby notwendig gemacht haben.

⁹beispielhaft zeigt Tabelle 1 im Kapitel Data Mining (3.6) das Ergebnis einer Auswertung für einen Kurs.

Vorlesung	R	R Square	Adj. R Square	Std. Error
Programmieren I	0.609	0.371	0.368	0.027

Tabelle 1: Regression Programmieren 1 und Durchschnittsnote

3.6. Data Mining

Abbildung 1 zeigt als Beispiel der grafischen Auswertung das Alter und die Durchschnittsnote. Tabelle 1 zeigt den Zusammenhang zwischen Vorlesungen des ersten Semesters und der Durchschnittsnote des Abschlusses, relevant ist hier die quadrierte Standardabweichung R^2 Squared.

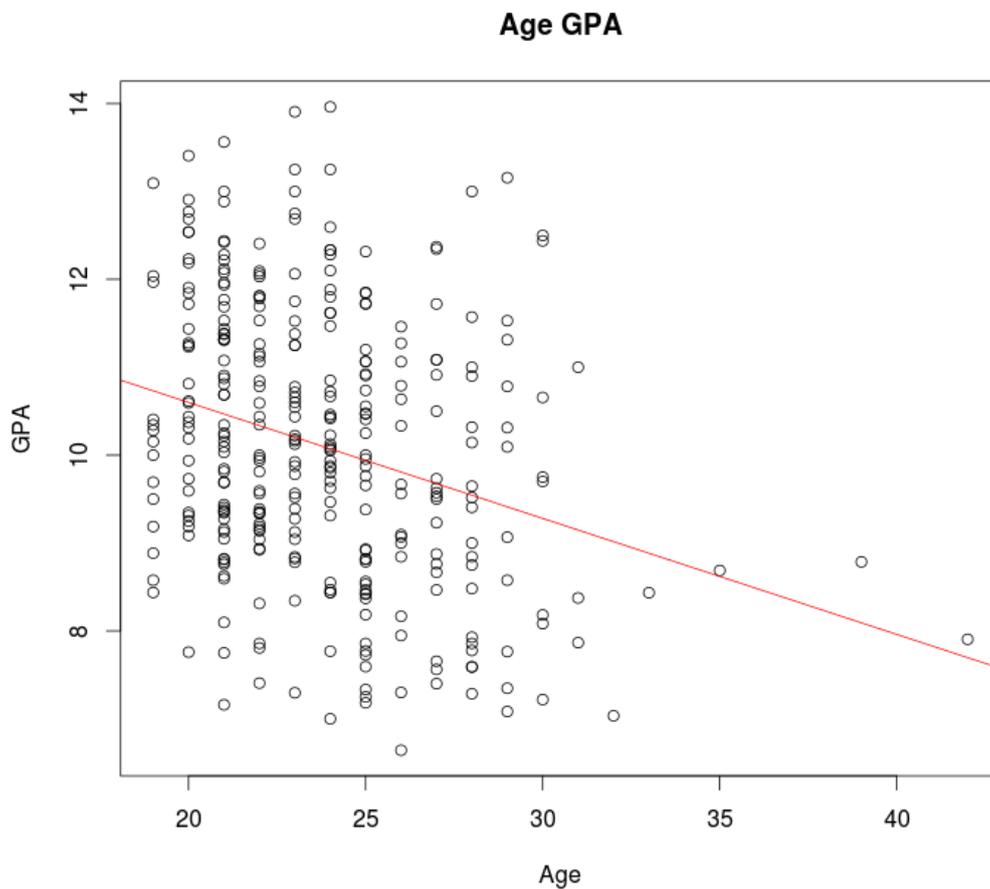


Abbildung 1: Alter bei Immatrikulierung zu Durchschnittsnote (Regressionsgerade in Rot)

4. Technischer Aufbau

In diesem Kapitel werden nach einem Überblick über den Datenfluss in der Anwendung die Datenbank und damit die Zieldatenstruktur sowie die Transformation der Quelldaten in die Zieldatenstruktur beschrieben.

Die Prüfungsdaten wurden einmalig in einem CSV-ähnlichen Format aus StISys exportiert und dann im Rahmen dieser Arbeit verwendet. Die im CSV denormalisierten Daten werden über einen Antlr generierten Parser¹⁰ weiterverarbeitet, der SQL als Ausgabeformat hat. In diesem Verarbeitungsschritt werden die Daten mit berechneten Attributen¹¹ angereichert sowie bereinigt¹².

Mit den so generierten Skripten wird eine normalisierte MySQL-Datenbank erzeugt, die dem in 4.1 gezeigten Datenbankmodell entspricht.

Die weitere Verarbeitung der Daten findet in *R* für statistische Auswertungen bzw. *Ruby on Rails* für interaktive Abfragen und Generierung von Grafiken oder *GnuPlot* für Batchgenerierung von Grafiken statt. Dieser Datenfluss ist in Abbildung 2 beschrieben. Hier ist auch die in Ruby erfolgende Vorverarbeitung der Daten vor der statistischen Analyse bzw. der Batchherzeugung von Grafiken abgebildet. Ruby wird hier genutzt, um die Verbindung zwischen *R* und der Datenbank zu vereinfachen. Die Kommunikation findet über flat files (csv) statt, was die in Listing 1 gezeigte einfache Weiterverarbeitung in *R* ermöglicht.

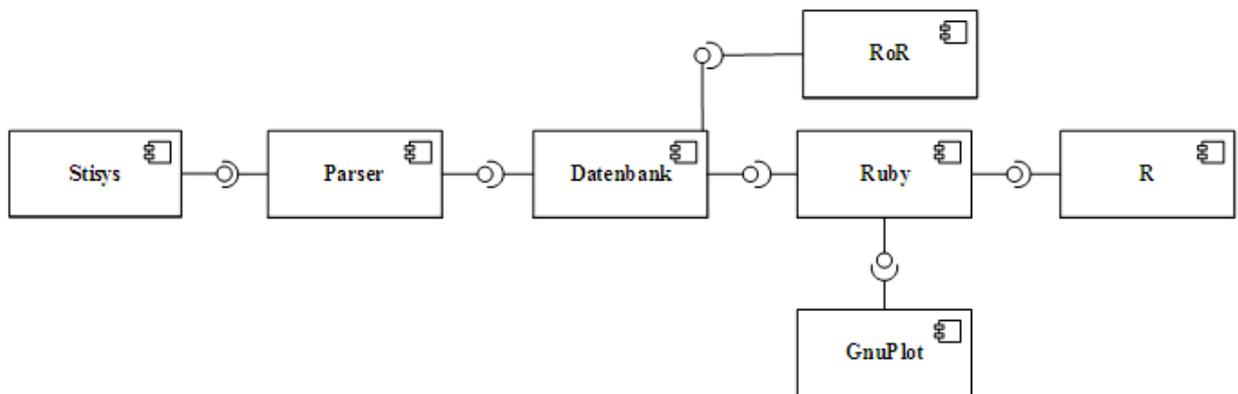


Abbildung 2: Technische Architektur/Datenfluss

¹⁰siehe Listing 2 für die Parser Grammatik

¹¹z.B. Alter bei Immatrikulierung, siehe Kapitel 4.2.2

¹²siehe auch Kapitel 3.3 Datenbereinigung

4.1. Datenbank

Die fachliche Zieldatenstruktur ist untenstehend in Abbildung 3 gezeigt. Listing A.2 im Anhang zeigt das SQL, mit dem die Datenbankstruktur generiert wird.

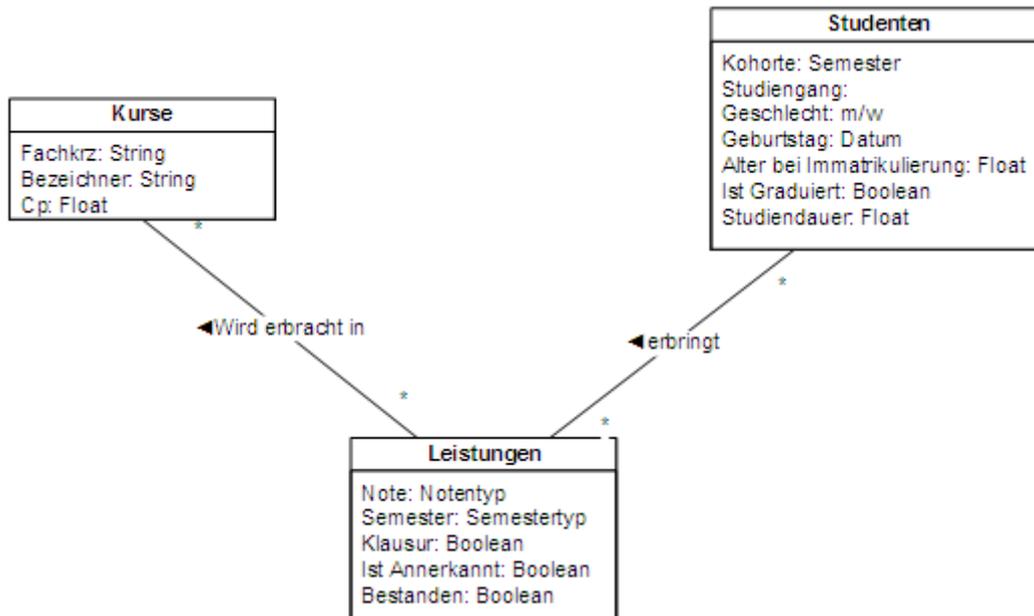


Abbildung 3: Fachliches Datenmodell

4.2. 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 4.2.1 die in den Quelldatensätzen vorliegenden Felder beschrieben und in den folgenden Kapiteln die Regeln für die berechneten Attribute.

4.2.1. Quelldatensätze

Listing 4.2.1 zeigt beispielhaft 3 Quelldatensätze, in denen farblich die unterschiedlichen enthaltenen Datenarten hervorgehoben sind. Die einzelnen Felder sind 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 4.1 beschrieben. Die Transformationsregeln werden in den darauffolgenden Kapiteln beschrieben.

4.2.2. Erzeugung Studentendatensätze

Die zu einem Studierenden gehörenden Datensätze werden an der `studId` erkannt und diesem zugeordnet. Nach dieser Gruppierung der Prüfungsdatsätze wird ein Studentendatsatz 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 – Semester des Studenten

4.2.3. Erzeugung Prüfungsdatsätze

Für die Prüfungsdatsätze 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'

4.2.4. 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.

5. Ausblick

Zusammenfassend wurde eine Plattform realisiert, um aus StlSys exportierte Daten automatisch einlesen und anreichern sowie analytisch und grafisch auswerten zu können. Aufbauend auf den Ergebnissen dieser Arbeit ist es möglich, weitere Analysen mit geringem zusätzlichen Implementierungsaufwand an den vorhandenen Datenbestand anzubinden und/oder diesen um weitere Datenquellen zu erweitern. Der geplante nächste Schritt ist die systematische Auswertung der hier nur prototypisch analysierten Aussagen aus der Literatur (speziell aus [Golding und Donaldson \(2006\)](#) und [Borrego u. a. \(2005\)](#)) auf dem in dieser Projektarbeit erzeugten mit relativ wenig Rauschen belasteten Datenbestand.

Weiterhin möglich ist in Zukunft z.B. die Einbindung von Datenquellen außerhalb des Departments TI, die die Zulassung der Studierenden betreffen. An der HAW ist dies konkret das System Helios, das im Bewerbungsverfahren um Studienplätze verwendet wird. Analytisch sind derzeit eine Clusteranalyse der vorhandenen Daten sowie weitere grafische Auswertungen geplant.

Literatur

- [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
- [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
- [Papula 2008] PAPULA, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler Band 3*. 5. Wiesbaden : Friedrich Vieweg & Sohn Verlagsgesellschaft mbH, 2008. – ISBN 978-3-8348-0225-5

A. Listings

A.1. Parser

Listing 2: 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 pjl.parser.generated;
}

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

    //add imports
    import pjl.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. Datenbankschema

Listing 3: Database Creation SQL

```

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 bool NOT NULL,
    studienDauer 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 attendings  
(  
student_id INT NOT NULL,  
course_id INT NOT NULL,  
note VARCHAR(255) NOT NULL,  
semester VARCHAR(255) NOT NULL,  
klausur bool NOT NULL,  
isAnnerkant bool NOT NULL,  
bestanden bool NOT NULL,  
FOREIGN KEY (student_id) REFERENCES students(id),  
FOREIGN KEY (course_id) REFERENCES courses(id)  
);
```