



Berliner
Hochschule
für Technik

Identifikation typischen Benutzerverhaltens in digitalen Studienformaten

Bachelorarbeit zur Erlangung des akademischen Grades Bachelor of Science
Berliner Hochschule für Technik · Fachbereich VI · Informatik und Medien

AUTOR

Werner Breitenstein
Matrikelnr.: 866059

BETREUER

Prof. Dr. Petra Sauer

GUTACHTER

Prof. Dr. Heike Ripphausen-Lipa

ABGABE

dd.mm.2022

Inhaltsverzeichnis

1. Einleitung	10
2. Grundlagen	11
2.1. Theorie	11
2.1.1. Standardisierte Vorgehensmodelle der Datenanalyse	12
2.1.2. Angepasstes Vorgehensmodell für diese Arbeit	16
2.1.3. Explorative Datenanalyse	20
2.1.4. Formen der Datenvizualisierung	20
2.2. Technik	20
2.3. Datenbasis	20
2.3.1. Beschreibung der Daten	20
2.3.2. Visualisierung der Daten	28
3. Analyse	34
3.1. Identifikation von Studenten	34
3.1.1. Ermittlung des Benutzerstatus	35
3.1.2. Kennzeichnung des Benutzerstatus	45
3.1.3. Zusammenfassung	48
3.2. Konkretisierung der zu untersuchenden Datenbasis	49
3.2.1. Betrachtung von viewed-Action und viewed-Events	49
3.2.2. Entscheidung für viewed-Events als Grundlage	51
3.3. Lokalität des Lern- und Kommunikationsverhaltens	52
3.3.1. Vergleich des Lern- und Kommunikationsverhaltens	52
3.3.2. Vergleich der Studiengänge	52
3.4. Kontinuität des Lern- und Kommunikationsverhaltens	53
3.4.1. Bestimmung des zeitlichen Maßstabs	53
3.4.2. Ermittlung der Vergleichsgöße	59
3.4.3. Kategorisierung nach IKK	64
3.4.4. Prüfung des Untersuchungsergebnisses	65
3.4.5. Vergleich des Lern- und Kommunikationsverhaltens	68
3.4.6. Vergleich der Studiengänge	68
3.5. Dynamik des Lern- und Kommunikationsverhaltens	68
3.5.1. Vergleich des Lern- und Kommunikationsverhaltens	71
3.5.2. Vergleich der Studiengänge	71
4. Ergebnisse	72
5. Fazit	73
6. Ausblick	74
Literaturverzeichnis	75
A. Anhang	76
A.2. Grundlagen	76

Erklärung zur Urheberschaft	90
Inhalt des beigefügten Datenträgers	91

Abbildungsverzeichnis

1.	Phasen des KDD-Prozesses. Original von Fayyad et al. (1996).	13
2.	Phasen des CRISP-DM. Original von Shearer (2000).	14
3.	KDD, SEMMA, CRISP-DM. Original von Azevedo & Santos (2008). . .	16
4.	Phasen des verwendeten Vorgehensmodells.	19
5.	Struktur und Art der importierten Originaldaten	22
6.	Menge aller Benutzer	24
7.	Menge der Log-Einträge pro Benutzer	25
8.	Menge der Benutzer pro Studiengang	25
9.	Menge der Kurse pro Benutzer	26
10.	Benutzer mit überdurchschnittlich vielen Kursen	27
11.	Menge der Studiengänge 1 bis 4 pro Benutzer	28
12.	Menge der Log-Einträge pro Benutzer (s. Anhang)	30
13.	Menge der Benutzer pro Studiengang	31
14.	Menge der Kurse pro Benutzer (s. Anhang)	32
15.	Mengen aller Actions in der Gesamtbetrachtung (s. Anhang)	37
16.	Menge der viewed-Actions pro Benutzer (s. Anhang)	38
17.	Anteil der viewed-Actions an der Gesamtaktivität	40
18.	Kombiniertes Datenset für Studenten und Andere	42
19.	Menge der Log-Einträge pro Aktivität und Benutzergruppe	43
20.	Identifikation von Studenten	46
21.	Überprüfung der Änderungen auf Vollständigkeit	47
22.	Überprüfung der Änderungen auf Richtigkeit	48
23.	Ermittlung korrespondierender viewed-Events	50
24.	Ermittlung korrespondierender sent-Events	52
25.	Verteilung der Log-Einträge pro Tag (s. Anhang)	54
26.	Verteilung der Log-Einträge pro Tag (Ausschnitt)	55
27.	Menge der Log-Einträge pro Student (s. Anhang)	57
28.	Menge der Arbeitswochen pro Student (s. Anhang)	58
29.	Menge der Arbeitstage pro Student (s. Anhang)	59
30.	Datenset zur Untersuchung zeitlicher Beziehungen (s. Anhang) . . .	61
31.	Individueller Kontinuitätskoeffizient (s. Anhang)	63
32.	Typisierung der Studenten nach IKK (s. Anhang)	65
33.	Typisierung nach IKK mit Bezug auf Log-Einträge (s. Anhang) . . .	66
34.	Menge der Log-Einträge für Student 62 und 64 (s. Anhang)	67
35.	Individueller Dynamikkoeffizient (s. Anhang)	71
36.	Typisierung der Studenten nach IDK (s. Anhang)	71
37.	Menge der Log-Einträge pro Benutzer	77
38.	Menge der Kurse pro Benutzer	78
39.	Mengenverteilung aller Actions in der Gesamtbetrachtung	79
40.	Menge der viewed-Actions pro Benutzer	80
41.	Verteilung der Log-Einträge im Gesamtzeitraum pro Tag	81
42.	Menge der Log-Einträge im Gesamtzeitraum pro Student	82

43.	Menge der Arbeitswochen im Gesamtzeitraum pro Student	83
44.	Menge der Arbeitstage im Gesamtzeitraum pro Student	84
45.	Datenset zur Untersuchung zeitlicher Beziehungen	85
46.	Individueller Kontinuitätskoeffizient (IKK)	86
47.	Typisierung der Studenten nach IKK mit Bezug auf IKK	87
48.	Typisierung der Studenten nach IKK mit Bezug auf Log-Einträge . .	88
49.	Menge der Log-Einträge pro Student im Gesamtzeitraum nach Tagen	89

Tabellenverzeichnis

1.	Schema des Datenbestandes mit Erläuterungen	23
----	---	----

Quellcodeverzeichnis

1.	Abfrage zu Struktur und Art der importierten Originaldaten	22
2.	Abfrage zur Menge aller Benutzer	24
3.	Abfrage zur Menge der Log-Einträge pro Benutzer	24
4.	Abfrage zur Menge der Benutzer pro Studiengang	25
5.	Abfrage zur Menge der Kurse pro Benutzer	25
6.	Abfrage zu Benutzern mit überdurchschnittlich vielen Kursen	27
7.	Abfrage zur Menge der Studiengänge 1 bis 4 pro Benutzer	27
8.	Auswahl der Arbeitsdaten	29
9.	Menge der Log-Einträge pro Benutzer	30
10.	Menge der Benutzer pro Studiengang	31
11.	Menge der Kurse pro Benutzer	32
12.	Mengen aller Actions in der Gesamtbetrachtung	36
13.	Menge der viewed-Actions pro Benutzer	37
14.	Anteil der viewed-Actions an der Gesamtaktivität	39
15.	Auswahl der Log-Einträge der Studenten	41
16.	Auswahl der Log-Einträge der Anderen	41
17.	Konkatenation der Datensets von Studenten und Anderen	41
18.	Menge der Log-Einträge pro Aktivität und Benutzergruppe	42
19.	Identifikation von Studenten	44
20.	Erstellen der neuen Tabelle moodle_data_students	47
21.	Kennzeichnung von Studenten	47
22.	Überprüfung der Änderungen auf Vollständigkeit	47
23.	Überprüfung der Änderungen auf Richtigkeit	48
24.	Ermittlung korrespondierender viewed-Events	49
25.	Ermittlung korrespondierender sent-Events	51
26.	Ergänzung des Merkmals <i>behaviour</i>	53
27.	Verteilung der Log-Einträge im Gesamtzeitraum pro Tag	54
28.	Menge der Log-Einträge pro Student	56
29.	Menge der Log-Einträge pro Student	56
30.	Menge der Arbeitswochen pro Student	57
31.	Menge der Arbeitswochen pro Student	57
32.	Menge der Arbeitstage pro Student	58
33.	Menge der Arbeitstage pro Student	59
34.	Erstellung des Datensets zur Untersuchung zeitlicher Beziehungen .	61
35.	Ermittlung der Arbeitswochen im Toleranzbereich	62
36.	Ermittlung der Arbeitstage im Toleranzbereich	62
37.	Ergänzung des IKK im Datenset <i>time_relation</i>	64
38.	Typisierung der Studenten nach Kontinuität ihrer Aktivitäten	64
39.	Typisierung der Studenten nach IKK	65
40.	Typisierung der Studenten nach IKK	66
41.	Menge der Log-Einträge im Gesamtzeitraum nach Tagen	67

42.	Import von Bibliotheken und anderen Erweiterungen	76
43.	Definitionen zur Darstellung der Visualisierungen	76
44.	Herstellung der Verbindung zur MySQL-Datenbank	76
45.	Import der Arbeitsdaten aus der MySQL-Datenbank	76

Zusammenfassung

...

Abstract

...

1. Einleitung

Ziel- und Endpunkt der Arbeit ist die detaillierte Analyse und Dokumentation des IST-Zustands. Es werden weder Prognosen abgeleitet noch Empfehlungen gegeben.

...

2. Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen dieser Arbeit beleuchtet und mithin wichtige Informationen zur angewandten Methodik, zu technischen Mitteln und zu dem zu untersuchenden Gegenstand bereitgestellt.

Ausgehend von in der Wissenschaft und in der Industrie seit langer Zeit anerkannten standardisierten Vorgehensmodellen wie dem *KDD – Knowledge Discovery in Databases Process* – (Fayyad, Piatetsky-Shapiro & Smyth, 1996) bzw. dem etwas jüngeren *CRISP-DM – Cross Industry Standard Process for Data Mining* – (Shearer, 2000) wird zunächst das im Rahmen dieser Arbeit praktizierte Analyseverfahren skizziert sowie die wesentlichen Grundlagen der explorativen Datenanalyse und der Visualisierung von Daten beschrieben.

Im folgenden zweiten Abschnitt werden die im Zuge der zahlreichen praktischen Untersuchungen eingesetzten Werkzeuge und Technologien vorgestellt.

Unter verschiedenen Aspekten wird abschließend die Datenbasis betrachtet und präsentiert. So werden hier die Daten u. a. durch Angaben zu ihrer Herkunft, ihrer Zusammensetzung und ihrer Qualität zum einen formal beschrieben. Statistische Abfragen sowie erste Visualisierungen z.B. zu bestehenden Mengengerüsten geben hier aber auch bereits interessante Einblicke in Struktur und Inhalt der Daten.

2.1. Theorie

Der Wunsch, Wissen aus Daten zu extrahieren, ist nicht nur sinnstiftend für diese Arbeit. Vielmehr ist er in der heutigen Informationsgesellschaft, in der viele erfolgreiche Geschäftsmodelle wie die der Big Five¹ gerade auf einer intelligenten wirtschaftlichen Verwertung dieser Ressource beruhen, nahezu allgegenwärtig.

¹ Die Bezeichnungen *The Big Five* oder auch *GAFAM* gelten den fünf größten globalen Technologieunternehmen: Google, Apple, Facebook, Amazon und Microsoft: [Statista, 01/2020](#)

2. Grundlagen

Aber nicht nur Google, Apple und andere haben früh erkannt, dass Daten gerade auch mit Blick auf ihr expansives Wachstum eine sehr ergiebige Quelle wertvoller Informationen² darstellen, sondern auch die Wissenschaften.

Diese letzteren waren es, die schon in den 1980er Jahren damit begonnen haben, Daten nicht nur sporadisch auf interessante Muster hin zu untersuchen, sondern unter dem Begriff *Data Mining* und später auch *Data Analytics* strategisch sinnvolle und allgemeingültige Prozesse zu etablieren (Runkler, 2020).

2.1.1. Standardisierte Vorgehensmodelle der Datenanalyse

Neben organisatorischen und wirtschaftlichen Erwägungen waren und sind es auch einfach faktische Gegebenheiten, die die Notwendigkeit der Standardisierung und Automatisierung von Analyseprozessen früh verdeutlichte und über die Jahre viele Experten zu entsprechenden Lösungsansätzen motivierte.

Denn wie Runkler (2020) und andere schreiben, ist die Datenanalyse ein stark interdisziplinärer Prozess, bei dem je nach Kontext oft mehrere Personen aus ganz unterschiedlichen Fachbereichen zusammenkommen. Damit liegt es auf der Hand, dass hier in einem äußerst heterogenen Umfeld von Experten, u. a. für Statistik, für maschinelles Lernen oder für Datenbanksysteme, die Orientierung an einem klar strukturierten Verfahren die Zusammenarbeit erheblich vereinfacht.

Konkrete wirtschaftliche Vorteile durch Zeit- und Kosteneinsparungen und die größere Objektivität bei der Durchführung der Analyse werden von Fayyad et al. (1996) als wichtige weitere Motive genannt. Schon im Jahr 1996 erkannten sie aber auch das Problem des *Data Overload* in manchen Bereichen der Forschung und sie wiesen darauf hin, dass ein organisierter Prozess unbedingt erforderlich ist, um die faktische Durchführbarkeit einer Datenanalyse überhaupt zu gewährleisten.

² Siehe hierzu die geschätzten Mengen der E-Mails, WhatsApp-Nachrichten oder YouTube-Uploads, die jede Minute allein im Internet entstehen bzw. verarbeitet werden: [Statista, 06/2021](#)

KDD – Knowledge Discovery in Databases Process

Der *Knowledge Discovery in Databases Process* (KDD), wie er von Fayyad et al. (1996) geprägt wurde, beschreibt einen umfassenden Datenanalyseprozess, in dessen Kern Verfahren des Data Mining zur Anwendung kommen.³

Die folgende Übersicht veranschaulicht die fünf verschiedenen Phasen des KDD – *Selektion, Vorverarbeitung, Transformation, Data Mining, Interpretation / Evaluierung* –, die, wie durch die gestrichelten Pfeile angedeutet, bei einer Analyse in vielen Fällen auch wiederholt durchlaufen werden müssen, bis tatsächlich ein aussagekräftiges Ergebnis vorliegt.

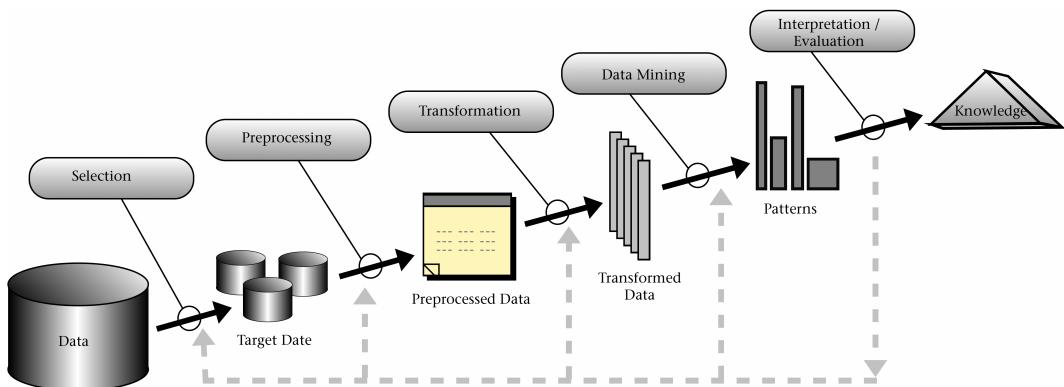


Abbildung 1.: Phasen des KDD-Prozesses. Original von Fayyad et al. (1996).

Über die genaue Zuordnung und Differenzierung von Arbeitsschritten innerhalb der oben dargestellten Hauptphasen des KDD, gibt es in der Literatur verschiedene Meinungen. Azevedo & Santos (2008) ordnen diese wie folgt ein:

1. *Selektion:* Auswahl des relevanten Teils des Datenbestands, der als Gegenstand der Untersuchung geeignet erscheint.
2. *Vorverarbeitung:* Zusammenführung und Bereinigung der selektierten Daten, bei der u. a. falsche und inkonsistente Daten entfernt werden sollten.
3. *Transformation:* Überführung der Daten u. a. mittels Konvertierung von Datentypen, wodurch z. B. verschiedene Datumsformate vereinheitlicht werden.

³ Unter dem folgenden Link findet sich dauerhaft die zitierfähige Version der im Text erwähnten Definition: [Gabler Wirtschaftslexikon, Springer Gabler, 04/2022](#)

4. *Data Mining*: Anwendung von Methoden und Algorithmen mit deren Unterstützung möglichst automatisch empirische Zusammenhänge aus der bereitgestellten Datenbasis extrahiert werden sollen.⁴
5. *Interpretation/Evaluierung*: Auslegung und Prüfung der gewonnenen Erkenntnisse, ggf. unterstützt durch Visualisierung extrahierter Muster.

CRISP-DM – Cross Industry Standard Process for Data Mining

Der *Cross Industry Standard Process for Data Mining* (CRISP-DM) ist ein auf Basis eines ehemals durch die EU geförderten Projekts entstandenes anwendungs- und branchenunabhängiges Vorgehensmodell für das Data Mining.

Konzipiert und entwickelt wurde das Vorhaben in den Jahren 1996 bis 2000 durch ein Konsortium namhafter Industrieunternehmen, der CRISP-DM Special Interest Group, der damals u. a. Daimler-Benz, NCR und ISL angehörten. Ihr Ziel war es, für Data Mining-Projekte ein nicht-propriätes Standard-Prozessmodell zu etablieren, das konkret als Blaupause dienen kann, um Datenbestände z. B. nach interessanten Mustern und Trends zu durchsuchen (Shearer, 2000).

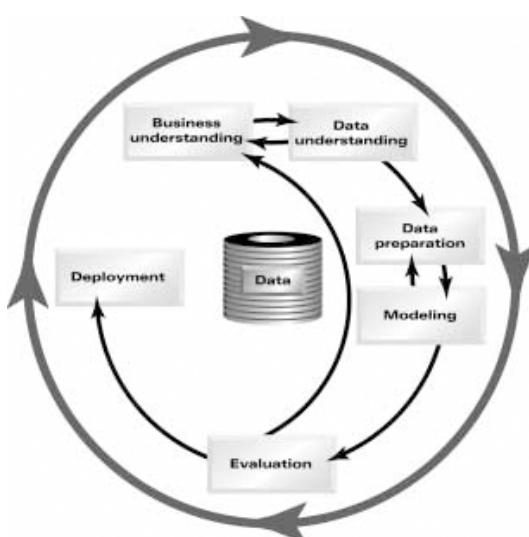


Abbildung 2.: Phasen des CRISP-DM. Original von Shearer (2000).

⁴ Unter dem folgenden Link findet sich dauerhaft die zitierfähige Version der im Text erwähnten Definition: [Gabler Wirtschaftslexikon, Springer Gabler, 04/2022](#)

2. Grundlagen

Wie in der obigen Abbildung ersichtlich, umfasst der CRISP-DM insgesamt sechs Phasen, die hiernach in einem normalen Data Mining-Projekt zu durchlaufen sind. Ähnlich wie beim KDD können sich verschiedene Phasen dabei wiederholen oder es wird auch ein Springen zwischen den einzelnen Phasen erforderlich.

Die Ziele und Aufgaben der einzelnen Phasen des CRISP-DM lassen sich nach Shearer (2000) folgendermaßen kurz zusammenfassen:

1. *Geschäftsverständnis*: Beschreibung übergeordneter Ziele, Anforderungen und Beschränkungen; Definition von Strategien, Aufgaben und Methoden.
2. *Datenverständnis*: Sammlung und Beschreibung der Rohdaten; Prüfung und Bewertung der Datenqualität; Feststellung von Datenmängeln.
3. *Datenaufbereitung*: Auswahl, Zusammenführung, Bereinigung und Transformation der Daten zur Erstellung des zu untersuchenden Datenbestands.
4. *Modellierung*: Auswahl und Anwendung geeigneter Modellierungstechniken; Erstellung von Tests; Bewertung und Optimierung von Modellen.
5. *Evaluierung*: Bewertung der Analyseergebnisse und der genutzten Modelle; Prüfung des Gesamtprozesses; Ableitung nachfolgender Verfahrensschritte.
6. *Einsatz*: Aufbereitung und Vorstellung der gewonnenen Erkenntnisse; Ausarbeitung von Strategien und Maßnahmen zur Einführung und dauerhaften Verwendung;

Vergleich der standardisierten Vorgehensmodelle

Zum Abschluss dieses Kapitels über die standardisierten Vorgehensmodelle in der Datenanalyse soll hier noch einmal auf die Arbeit von Azevedo & Santos (2008) hingewiesen werden, die zum Ziel hatte die Gemeinsamkeiten und Unterschiede von KDD, CRISP-DM und SEMMA⁵ miteinander zu vergleichen.

⁵ Unter dem folgenden Link findet sich eine kurze Einführung zu SEMMA, das den übergeordneten Prozess für den SAS® Enterprise Miner™ darstellt: [Introduction to SEMMA, SAS, 04/2022](#)

2. Grundlagen

Im Ergebnis bestätigt diese Vergleichsstudie die vollkommene Übereinstimmung von KDD und SEMMA, bzw. definiert SEMMA als praktische Implementation des älteren KDD-Prozesses, weshalb auch in dieser Arbeit auf eine Darstellung dieses Standardprozesses verzichtet wurde.

Im Vergleich von KDD und CRISP-DM gibt es dagegen erkennbare Unterschiede, die sich darin zeigen, dass der CRISP-DM die im KDD implizit enthaltenen vor- und nachgelagerten Stufen explizit als separate Teil des Prozesses ausführlich beschreibt. Weitere Abweichungen lassen sich feststellen bei der Zuordnung von Teilschritten innerhalb des *Data Understanding* und *Data Preparation*. Interessanterweise wird dies in dieser Studie nicht konsistent behandelt, und stimmt daher auch nur bedingt mit dem ursprünglich von Shearer (2000) skizzierten Prozess überein.

KDD	SEMMA	CRISP-DM
Pre KDD	-----	Business understanding
Selection	Sample	Data Understanding
Pre processing	Explore	
Transformation	Modify	Data preparation
Data mining	Model	Modeling
Interpretation/Evaluation	Assessment	Evaluation
Post KDD	-----	Deployment

Abbildung 3.: KDD, SEMMA, CRISP-DM. Original von Azevedo & Santos (2008).

2.1.2. Angepasstes Vorgehensmodell für diese Arbeit

Die im vorausgegangenen Abschnitt präsentierten Vorgehensmodelle haben alle-samt dasselbe Ziel: Sie wollen den äußerst vielfältigen Prozess einer Datenanalyse möglichst vollständig und genau in einem Standardverfahren abbilden und für den Anwender sinnvolle Handlungsempfehlungen formulieren.

Diese Verfahren sind also keineswegs verpflichtend. Sie sollen zur Orientierung dienen, aber es obliegt demnach stets dem Anwender je nach Anwendungskontext die standardisierten Verfahrensschritte auf die im konkreten Fall vorliegenden Anforderungen anzupassen (Shearer, 2000).

Grundzüge des verwendeten Vorgehensmodells

Im Hinblick auf die anstehenden Untersuchungen im Rahmen dieser Arbeit, wird das im weiteren Verlauf verwendete Vorgehensmodell – auf Basis des von Shearer (2000) beschriebenen CRISP-DM – wie folgt skizziert:

1. *Geschäftsverständnis:* Das Thema dieser Arbeit definiert gleichzeitig auch das übergeordnete Ziel, die *Identifikation typischen Benutzerverhaltens in digitalen Studienformaten*. Untergeordnete Ziele lassen sich mit Blick auf die Methodik und den Gegenstand der Untersuchung beschreiben. So gilt es, wie in der Einleitung zu dieser Arbeit beschrieben, mit Mitteln der explorativen Datenanalyse den Ist-Zustand studentischen Lern- und Kommunikationsverhaltens möglichst detailliert zu skizzieren und das jeweilige Vorgehen dabei verständlich und nachvollziehbar zu dokumentieren. Dazu bedarf es im Rahmen der eigentlichen Analyse neben der bestimmten Auswahl von Daten gerade auch der gezielten Entwicklung von Fragen, die geeignet sein könnten, das in den Daten verborgene Benutzerverhalten zu offenbaren und davon ausgehende neue Annahmen zu formulieren.
2. *Datenverständnis:* Ein fundiertes Verständnis über die Herkunft der zu untersuchenden Daten, deren Bedeutung und Qualität ist essentiell, um mögliche Zusammenhänge zu verstehen oder neues Wissen aus den Daten extrahieren zu können. Das nachfolgende Kapitel [Datenbasis](#) trägt diesem grundlegenden Erfordernis Rechnung und gibt detailliert Aufschluss über den Gegenstand der Untersuchung.
3. *Datenaufbereitung:* Im Fokus dieser Phase steht der konkrete Untersuchungsgegenstand. Dessen Bereitstellung vollzieht sich entsprechend der gegebenen Zielsetzung in mehreren Schritten. Zu nennen sind hier in erster Linie:
 - Datenauswahl: Die für die Untersuchung relevanten Daten sind nach Art und Umfang aus den Spalten und Zeilen der initial vorbereiteten Daten zu selektieren. Warum gewisse Daten relevant sind bzw. diese nicht in der Auswahl berücksichtigt werden, sollte begründet werden können.

2. Grundlagen

- Datenbereinigung: Da die Daten initial keine falschen Werte aufweisen, entfällt naturgemäß eine entsprechende Korrektur. Gegebenfalls müssen aber fehlende Werte ergänzt werden, um bestimmte Abfragen sinnvoll durchführen zu können.
 - Datentransformation: Für eine Untersuchung kann es erforderlich sein, zuvor aus den Daten ein neues Attribut abzuleiten, den Datentypen eines Attributs zu konvertieren oder auch weitere Datensätze zu ergänzen. Die Gründe hierfür sollten ebenfalls klar ersichtlich dokumentiert werden.
4. *Datenanalyse*:⁶ Das Verfahren, das bei den eigentlichen Untersuchungen zur Anwendung kommen soll, orientiert sich an der Methodik der explorativen Statistik bzw. der [explorativen Datenanalyse](#). Insbesondere durch geeignete visuelle Darstellungen⁷ sollen in den Daten bemerkenswerte Strukturen und Zusammenhänge aufgezeigt werden, die zur Formulierung von Hypothesen anregen. Mögliche Darstellungsformen sind beispielsweise:
- Balkendiagramm
 - Streudiagramm
 - Liniendiagramm
- Aufgrund komplexer Fragestellungen und Zwischenbewertungen sind bei der Analyse oft mehrere Anläufe nötig, um schließlich interessante Hypothesen generieren zu können. Gegebenenfalls muss auch die Frage selbst angepasst werden bzw. sind auch die Daten erneut aufzubereiten.
5. *Evaluierung*: Die Interpretation und die Bewertung von Analyseergebnissen vollzieht sich typischerweise im stetigen Wechsel mit der Optimierung der Methoden in der vorhergehenden Analysephase. Das Ziel ist dabei nur die Entwicklung einer Hypothese auf den erkannten Mustern oder Verbindungen in den Daten, nicht aber die Evaluierung der Hypothese selbst oder die Ableitung weiterer Verfahrensschritte aus einer gewonnenen Hypothese.

⁶ Im weiteren Verlauf der Arbeit soll diese Phase vorzugsweise *Datenanalyse* genannt werden, da der Begriff Modellierung häufig die Anwendung komplexer Machine Learning Modelle impliziert.

⁷ Siehe hierzu auch das nachfolgende Kapitel [Formen der Datenvisualisierung](#)

6. *Dokumentation:*⁸ Erkenntnisse aus den Untersuchungen sind letztlich noch verständlich aufzubereiten und umfassend zu dokumentieren, so dass diese z. B. auch in einer neuen Studie zur Entwicklung von Kursempfehlungen genutzt werden könnten. Im Kapitel [Ergebnisse](#) werden dazu wichtige Erfahrungen aus dieser Arbeit zusammengefasst sowie bemerkenswerte Untersuchungsansätze und deren Resultate betrachtet bzw. miteinander verglichen.

Dieses Modell wird später bei der tatsächlichen Durchführung der Analyse (siehe das folgende Kapitel [Analyse](#)) erneut als Vorlage dienen und wie erwähnt in den Phasen *Datenaufbereitung*, *Datenanalyse* und *Evaluierung* je nach Anforderung auch mehrmals spezifisch angepasst werden müssen.

Die nachfolgende Grafik zeigt das in dieser Arbeit verwendete Vorgehensmodell mit den oben beschriebenen Phasen. Die nur im Rahmen der konkreten Analyse zu durchlaufenden Phasen sind dabei farblich hervorgehoben.

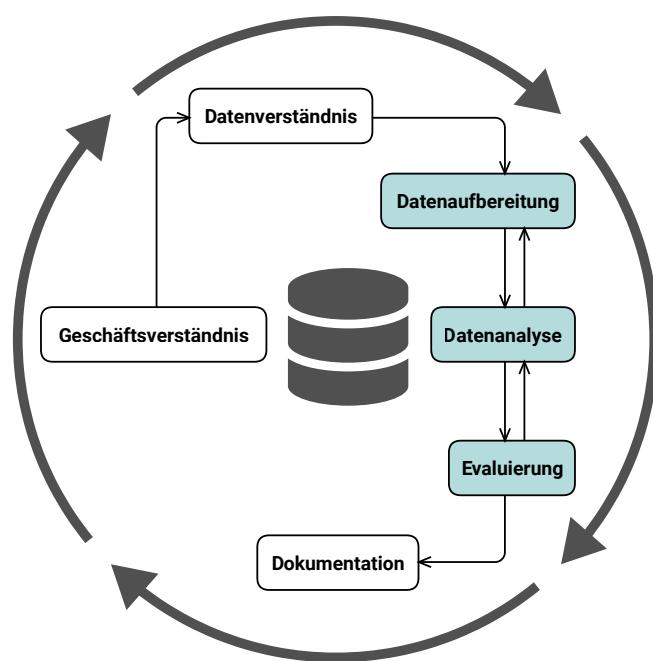


Abbildung 4.: Phasen des verwendeten Vorgehensmodells.

⁸ In dieser Arbeit soll diese Phase bevorzugt mit *Dokumentation* bezeichnet werden, da der Begriff *Einsatz* zu sehr auf die praktische Anwendung konkreter Untersuchungsergebnisse abzielt.

2.1.3. Explorative Datenanalyse

...

2.1.4. Formen der Datenvisualisierung

...

2.2. Technik

Hier finden sich Ausführungen zu den verwendeten Technologien, Tools, Libraries, etc.

...

2.3. Datenbasis

Gegenstand der Untersuchungen zu dieser Arbeit ist ein vom *Projektteam DiSEA* zur Verfügung gestellter Datenbestand aus dem Wintersemester 2020/2021⁹. In diesem enthalten sind die anonymisierten Moodle-Daten von Studenten, Dozenten sowie anderem Personal (in der weiteren Arbeit ‹Andere› genannt) der *Berliner Hochschule für Technik (BHT)* und der *Alice Salomon Hochschule Berlin (ASH)* aus den folgenden Studiengängen:

- Master-Studiengang Medieninformatik Online (MMIO)
- Bachelor-Studiengang Wirtschaftsingenieurwesen Online (BWIO)
- Bachelor-Studiengang Wirtschaftsinformatik Online (BWINF)
- Bachelor-Studiengang Soziale Arbeit Online (BSAO)

2.3.1. Beschreibung der Daten

Um den Zugriff auf die Daten und deren praktische Untersuchung zu erleichtern, wurden diese zunächst vom Projektteam aus der Datenbank des Moodle-Systems

⁹ Das gesamte Semester musste nach der SARS-CoV-2-Infektionsschutzmaßnahmenverordnung des Berliner Senates unter erhöhten Sicherheitsbedingungen stattfinden. Die Regelungen für das Lehr- und Prüfungsgeschehen wurden an der BHT infolgedessen wie folgt angepasst:

- keine Lehrveranstaltungen und Prüfungen in Präsenz
- keine Zählung des Semesters als Fachsemester
- keine Zählung von Prüfungsfehlversuchen

2. Grundlagen

(Green, 2022) extrahiert und in einem ersten Arbeitsschritt in nur einer Relation zusammengeführt.

Hierbei wurden Merkmale, die für diese Arbeit erwartungsgemäß keinen Mehrwert besitzen bereits eliminiert, während z. B. das Attribut *Studiengang* als neue Spalte in die Tabelle aufgenommen wurde, um die Zuordnung der Datensätze zu den jeweiligen Studiengängen¹⁰ unmittelbar erkennen zu können.

Des weiteren wurden vorab die Merkmale *course_module_type* und *instanceid* eingefügt, um bei der Datenanalyse auch deren Informationsgehalt zur Identifikation typischen Benutzerverhaltens sinnvoll nutzen zu können.

Damit die Daten in einem beliebigen IT-Umfeld einfach weiterverarbeitet werden können, wurden sie im Anschluss an ihre Vorbereitung in einem für diesen Zweck typischen CSV-Format exportiert. Übergeben wurden die CSV-Daten schließlich als offene und komprimierte Textdateien in ASCII-Kodierung, in der die Daten entgegen der üblichen Praxis jedoch nicht durch Kommata, sondern durch Semikola strukturiert waren.

Der bereitgestellte Datenbestand umfasst insgesamt 969032 Datensätze. Dabei handelt es sich um eine Teilmenge von Log-Einträgen auf dem Moodle-Server, mit denen client- und serverseitige Aktionen fortlaufend protokolliert werden. Typische Aktionen, die so u. a. aufgezeichnet werden sind das Aufrufen eines Kursmoduls, das Starten eines Uploads, das Senden einer Nachricht oder auch die Bewertung einer Aufgabe.

¹⁰ Ergänzend zu den genannten offiziellen Studiengängen, sind in den Daten ferner auch Datensätze zu einem Studiengang 0 enthalten. Hierbei handelt es sich jedoch um eine besondere Entität, die sich nur auf Aktivitäten bezieht, die außerhalb des eigentlichen Kursgeschehens stattfanden, z. B. Logins, Chats oder Aufrufe des Kalenders bzw. Dashboards.

Formale Angaben über die Daten

Ein erster informativer Einblick in die Struktur und die Art der zu untersuchenden Daten ergibt sich nach deren Import in eine MySQL-Datenbank mithilfe der folgenden einfachen SQL-Abfrage:

```
1 DESCRIBE moodle_data;
```

Listing 1: Abfrage zu Struktur und Art der importierten Originaldaten

Field	Type	Null	Key	Default	Extra
courseid	int(11)	YES		NULL	
Studiengang	varchar(11)	YES		NULL	
userid	int(11)	YES	MUL	NULL	
relateduserid	int(11)	YES		NULL	
action	varchar(10)	YES		NULL	
eventname	varchar(57)	YES		NULL	
objecttable	varchar(27)	YES		NULL	
objectid	int(11)	YES		NULL	
timecreated	int(11)	YES		NULL	
course_module_type	varchar(18)	YES		NULL	
instanceid	int(11)	YES		NULL	

Abbildung 5.: Struktur und Art der importierten Originaldaten

Die obige Ausgabe beschreibt das Schema der importierten Daten. Von Interesse für diese Arbeit sind hier aber nur die Werte zu *Field* und *Type*, die die Spaltennamen der Tabelle und die Datentypen der darin enthaltenen Werte angeben.

Informationen und deren Beziehungen

Die nachfolgende tabellarische Übersicht zeigt nun, welche Informationen in den Feldern der verschiedenen Merkmale des Datenbestandes tatsächlich enthalten sind und in welchen Beziehungen diese innerhalb der aktuell betriebenen relationalen Datenbank des VFH-Moodle stehen.¹¹

¹¹ Siehe auch die Moodle Entity Relationship Documentation (Green, 2022): [Moodle ERD, 05/2022](#)

2. Grundlagen

Merkmal	Information / Beziehung innerhalb des VFH-Moodle
courseid	Studienmodul, das im WS 2020/2021 belegt wurde. <i>Fremdschlüssel zur Identifikation eines bestimmten Studienmoduls in der Relation course.</i>
Studiengang	Studiengang, in dem aktuell studiert wird. <i>Frei gewählte Kennziffer zur eindeutigen Unterscheidung der Studiengänge; bedeutet keine Referenz auf eine andere Entität.</i>
userid	Kennzahl zur Identifikation des Benutzers. <i>Aus Datenschutzgründen verschlüsselte ID zur Identifikation eines bestimmten Benutzers (z. B. der Sender einer Nachricht).</i>
relateduserid	Kennzahl zur Identifikation eines weiteren Benutzers. <i>Verschlüsselte ID des interagierenden Benutzers, der z. B. bei einem Chat den Empfänger einer Nachricht repräsentiert.</i>
action	Interaktion, die im Moodle-System ausgeführt wurde. <i>Allgemeinere Form des eventtype, der auch im eventname als notwendiger Bestandteil redundant enthalten ist.</i>
eventname	Mehrteiliger Bezeichner für das ausgelöste Event. <i>Ausgelöst durch eine Interaktion wird ein Bezeichner durch die drei Werte modulename, instance und eventtype der Relation event generiert und eingetragen.</i>
objecttable	Relation zur Verwaltung von Objekttabellen. <i>Abhängig von der Art des Kursmoduls und der Interaktion werden die durch Verwendung bestimmter Objekte tangierten Tabellen dokumentiert, z. B. assign_grades, course_modules oder forum_discussions</i>
objectid	Kennzahl zur Identifikation des verwendeten Objekts. <i>Fremdschlüssel zur Identifikation des durch die Interaktion tangierten Objekts in der zugehörigen Relation objecttable.</i>
timecreated	Zeitpunkt der ausgeführten Interaktion. <i>10-stelliger Unix Epoch Timestamp, der seit Donnerstag, dem 01.01.1970, 00:00 Uhr UTC die vergangenen Sekunden zählt.</i>
course_module_type	Typ des verwendeten Kursmoduls. <i>Zur Anreicherung des Informationsgehalts aus der Relation course_modules entnommener Bezeichner des Modultyps, z. B. assign, forum, label oder resource</i>
instanceid	Kennzahl zur Identifikation des Kursmodultyps. <i>Fremdschlüssel zur Identifikation des Kursmodultyps in der zugehörigen Relation course_modules.</i>

Tabelle 1.: Schema des Datenbestandes mit Erläuterungen

Erste Erkenntnisse über die Daten

Um die Beschreibung der Daten zu vervollständigen, soll im Folgenden anhand einiger statistischer Abfragen der Gegenstand der Untersuchung, die sogenannten Arbeitsdaten, inhaltlich genauer betrachtet und mithin erste Erkenntnisse daraus gewonnen werden.

```
1 SELECT COUNT(DISTINCT userid) AS "total_number_users"
2 FROM moodle_data;
```

Listing 2: Abfrage zur Menge aller Benutzer

```
+-----+
| total_number_users |
+-----+
|           144 |
+-----+
1 row in set (0,00 sec)
```

Abbildung 6.: Menge aller Benutzer

Im Ergebnis inkludiert sind neben Einzelpersonen auch zwei Benutzergruppen, die einer Beobachtung ihres Verhaltens nicht zugestimmt haben (userid = -2) oder die im Bachelor-Studiengang Medieninformatik aktiv waren (userid = -3)¹². Abzüglich dieser beiden Gruppen erhielte man im Ergebnis somit 142 Einzelpersonen.

```
1 SELECT userid, COUNT(userid) AS "total_number_records"
2 FROM moodle_data
3 GROUP BY userid;
```

Listing 3: Abfrage zur Menge der Log-Einträge pro Benutzer

¹² Um die Privatsphäre meiner Komilitonen zu schützen und meine Unvoreingenommenheit bei den Untersuchungen zu wahren, wurde vom Projektteam entschieden, alle Studenten im Bachelor-Studiengang Medieninformatik in einer Gruppe zusammenzufassen und diese nur bei Kontextbetrachtungen zu berücksichtigen.

2. Grundlagen

userid	total_number_records
...	...
1	3865
2	4706
3	3373
...	...
26	92242
...	...
142	10
143	1387
144	240
+-----+	

144 rows in set (0,27 sec)

Abbildung 7.: Menge der Log-Einträge pro Benutzer

Aus Platzgründen werden in der obigen Ergebnistabelle nur wenige der insgesamt 144 Zeilen des Abfrageergebnisses angezeigt. Es wird aber auch bereits in diesem kleinen Ausschnitt deutlich, wie unterschiedlich die Benutzeraktivitäten über das Semester hinweg in ihrem Umfang waren.

```

1 SELECT Studiengang, COUNT(DISTINCT userid) AS "total_number_users"
2 FROM moodle_data
3 GROUP BY Studiengang;
```

Listing 4: Abfrage zur Menge der Benutzer pro Studiengang

Studiengang	total_number_users
0	144
1	54
2	40
3	33
4	25
+-----+	

5 rows in set (0,46 sec)

Abbildung 8.: Menge der Benutzer pro Studiengang

Bemerkenswert am Ergebnis ist, dass dem allgemeinen Studiengang 0 alle zuvor ermittelten Benutzer zugeordnet sind, deren Summe in den Studiengängen 1 bis 4 dagegen höher liegt. Insofern lässt sich an dieser Stelle bereits folgern, dass es auch Benutzer gegeben haben muss, die in mehreren Studiengängen aktiv waren, insbesondere auch deshalb, da manche Benutzer wie z. B. Angehörige der Hochschulverwaltung nicht am Geschehen in den offiziellen Studiengängen teilnehmen.

```

1 SELECT userid, COUNT(DISTINCT courseid) AS "total_number_courses"
2 FROM moodle_data
3 GROUP BY userid
```

```
4 ORDER BY total_number_courses;
```

Listing 5: Abfrage zur Menge der Kurse pro Benutzer

userid	total_number_courses
144	2
...	...
130	3
...	...
42	4
...	...
47	5
...	...
95	6
...	...
63	7
...	...
67	8
...	...
48	9
...	...
81	10
...	...
111	12
...	...
69	16
...	...
16	20
...	...
18	24
...	...
35	28
...	...
114	30
...	...
-3	34
...	...
32	39
26	168
-2	195

144 rows in set (1,96 sec)

Abbildung 9.: Menge der Kurse pro Benutzer

2. Grundlagen

Auch wenn die Tabelle die Ergebnisse aus Platzgründen wiederum nur teilweise darstellt, ist sofort zu erkennen, dass die Menge an Kursen pro Benutzer mitunter weit über der empfohlenen Menge von sechs Kursmodulen für ein Vollzeitstudium in Regelstudienzeit lag. Dies könnte in manchen Fällen wie z. B. beim Benutzer mit der userid 26 mit einer Dozententätigkeit zu begründen sein oder auf eine andere Rolle hindeuten, was aber erst im Hauptteil dieser Arbeit untersucht werden soll.

Den beiden Benutzergruppen mit der userid -2 und -3 sind erwartungsgemäß ebenfalls große Kursmengen zugeordnet, da diese Gruppen eine unbekannte Zahl an Einzelpersonen umfassen. Infolgedessen nehmen sie hier eine Sonderrolle ein und werden nur der Vollständigkeit halber ebenfalls angezeigt. Bei den weiteren Untersuchungen wird je nach Anforderung stets abzuwägen sein, inwiefern diese beiden Personengruppen bei der Interpretation der Ergebnisse tatsächlich berücksichtigt werden dürfen.

Mit Blick auf die unerwartet hohen Mengen an Kursen pro Benutzer soll zum Schluss dieses Kapitels die Anzahl an Benutzern mit überdurchschnittlich vielen Kursen und die Zuordnung von Benutzern und Studiengängen betrachtet werden.

```
1 SELECT userid, COUNT(DISTINCT courseid) AS "total_number_courses"
2 FROM moodle_data
3 WHERE userid > 0
4 GROUP BY userid
5 HAVING total_number_courses >= 12
6 ORDER BY total_number_courses;
```

Listing 6: Abfrage zu Benutzern mit überdurchschnittlich vielen Kursen

userid	total_number_courses
68	12
...	...
114	30
78	31
53	33
133	34
32	39
26	168

84 rows in set (1,71 sec)

Abbildung 10.: Benutzer mit überdurchschnittlich vielen Kursen

```
1 SELECT userid, COUNT(DISTINCT Studiengang) AS "total_number_studies"
2 FROM moodle_data
```

```

3 WHERE Studiengang > 0 AND userid > 0
4 GROUP BY userid
5 HAVING total_number_studies > 1
6 ORDER BY total_number_studies;

```

Listing 7: Abfrage zur Menge der Studiengänge 1 bis 4 pro Benutzer

userid	total_number_studies
44	2
6	2
81	2
27	2
28	2
50	2
29	2
30	2
31	2
32	2
55	2
88	2
21	3
26	4

14 rows in set (1,71 sec)

Abbildung 11.: Menge der Studiengänge 1 bis 4 pro Benutzer

Auch die letzten zwei Abfragen, bei denen nur Einzelbenutzer (s. WHERE-Klausel) betrachtet wurden, können mit ihren Ergebnissen überraschen. So waren 84 von 142 Benutzern und damit wohl auch eine höhere Zahl an Studenten über das Semester hinweg in mindestens doppelt so vielen Kursen aktiv, wie es von den Hochschulen für ein Vollzeitstudium in der Regel empfohlen wird.

Der Gedanke, dass es dann auch Benutzer geben haben könnte, die außer dem unspezifischen Studiengang 0 (s. WHERE-Klausel) vielleicht mehrere der eingangs genannten Studiengänge besucht haben, wird durch die Abfrage zur Anzahl der Studiengänge pro Benutzer eindrucksvoll bestätigt: Insgesamt 14 Benutzer waren in mehr als einem der **Studiengänge 1 bis 4** tätig. Dieser Umstand könnte ebenfalls für eine Dozententätigkeit der im Ergebnis enthaltenen Benutzer sprechen und soll im weiteren Verlauf der Arbeit noch genauer untersucht werden.

2.3.2. Visualisierung der Daten

Ergänzend zur vorhergehenden Beschreibung der Daten mittels allgemeiner Ausführungen zum Untersuchungsgegenstand und verschiedener SQL-Abfragen über

dessen Struktur und Inhalt, soll nun in diesem Abschnitt die Datenbasis anhand graphischer Untersuchungsmethoden anschaulich dargestellt werden.

Dabei soll es aber nicht nur darum gehen, die Abfrageergebnisse des vorherigen Kapitels ansprechend zu visualisieren. Vielmehr soll hier bereits mit Blick auf den nachfolgenden Hauptteil praktisch gezeigt werden, wie bei Analysen methodisch vorzugehen ist. Die Analysen selbst sind dabei in ihrem Umfang kurz gehalten.

Beispiele mit Hinweisen zur Durchführung von Analysen

Der Ablauf von Analysen orientiert sich an dem zuvor im Kapitel *Grundzüge des verwendeten Vorgehensmodells* vorgestellten **Vorgehensmodell** für Datenanalysen und ist demnach unterteilt in Datenaufbereitung, Datenanalyse und Evaluierung.

Anhand einer beispielhaften ersten Untersuchung soll nun dieser Ablauf in ein Schema konkreter Verfahrensschritte übersetzt werden, das wiederum i. S. einer Vorlage referenziert werden kann.¹³

Um das Vorgehen vollständig aufzuzeigen, den Text hier jedoch nicht mit Nebeninformationen zu überladen, werden die für dieses Analysebeispiel notwendigen Vorbereitungen im **Anhang** im einleitenden Prolog exemplarisch vorgestellt. Die untenstehende Datenaufbereitung schließt sich hieran nahtlos an.

Datenaufbereitung

Gegenstand der Untersuchung sind an dieser Stelle nur Datensätze mit einer userid größer als 0. Damit werden jene Benutzer bei der Analyse nicht beachtet, die einer Beobachtung ihres Verhaltens nicht zugestimmt haben (userid = -2) oder die im Bachelor-Studiengang Medieninformatik Online studierten (userid = -3).

```
1 # Konvertierung des Datentyps des Tabellenmerkmals timecreated
2 moodle_data['timecreated'] =
3     pd.to_datetime(moodle_data['timecreated'], unit='s')
4 moodle_data = moodle_data[moodle_data.userid > 0]
5 moodle_data
```

Listing 8: Auswahl der Arbeitsdaten

Datenanalyse: Menge der Log-Einträge pro Benutzer

¹³ Siehe auch die zu dieser Arbeit beigefügten Jupyter Notebook Dokumente.

```

1 # Spezifische Definitionen zur Darstellung der Visualisierung
2 plt.figure(figsize=(64, 36)) # Größe der Visualisierung (in inch)
3
4 # Visualisierung der Menge der Log-Einträge pro Benutzer
5 chart = sns.countplot(x=moodle_data.userid)
6
7 # weitere Anweisungen zur Darstellung der Visualisierung
8 chart.grid(axis='y')
9 chart.set_axisbelow(True)
10 chart.set_xlabel('moodle_data.userid')
11 chart.set_ylabel('total number records')
12 chart.tick_params(left=False, bottom=False)
13 sns.despine(left=True)
14 plt.show()

```

Listing 9: Menge der Log-Einträge pro Benutzer

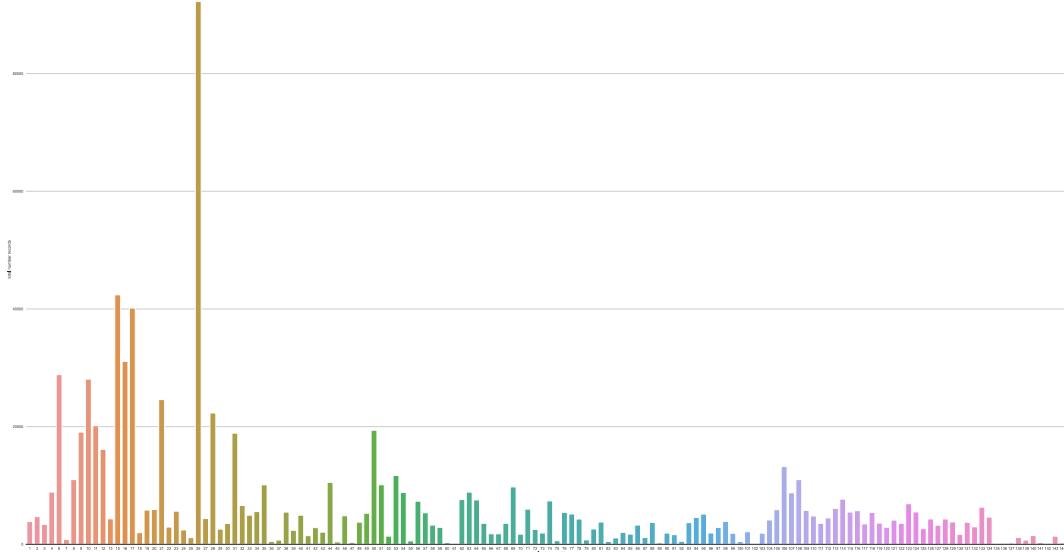


Abbildung 12.: Menge der Log-Einträge pro Benutzer ([s. Anhang](#))

Um in dieser Arbeit auch größere Visualisierungen leicht verständlich abbilden und evaluieren zu können, sollen diese im Hauptteil nach Möglichkeit nur in relevanten Ausschnitten inklusive einem Verweis auf den Anhang präsentiert werden. In Fällen wie oben, wo dieses dagegen wenig sinnvoll erscheint, weil z. B. eine Gesamtbetrachtung erfolgen soll, sind die Abbildungen insgesamt einfach zu verkleinern und mit einem Link auf das großformatige Original zu versehen. Ergänzend sei hier auch noch einmal auf die Plots in den beigefügten Jupyter Notebooks verwiesen.

Evaluierung

Die obige Abbildung lässt erahnen, warum Visualisierungen für die Datenanalyse bestens geeignet sind: In der kompakten Darstellung zeigen sich z. B. die Benutzer

mit minimalen oder maximalen Werten, wie auch die Häufung höherer Werte bei Benutzern mit einer niedrigen userid deutlich schneller als in jeder Ergebnistabelle.

Als Basis der folgenden Analyse diente erneut die oben im Listing [Auswahl der Arbeitsdaten](#) definierte Datenaufbereitung, d. h. die Benutzer, die der Beobachtung ihres Verhaltens nicht zugestimmt haben oder jene die im Bachelor-Studiengang Medieninformatik studierten, wurden bei der Untersuchung nicht berücksichtigt.

Aus Gründen der Vergleichbarkeit mit dem Ergebnis der korrespondierenden SQL-Abfrage zur [Menge der Benutzer pro Studiengang](#) und um die Größenunterschiede der Benutzermengen noch einmal besser verständlich aufzuzeigen, wird auch bei dieser Analyse der übergeordnete Studiengang 0 mitberücksichtigt.

Aus Gründen der Übersichtlichkeit werden im weiteren Verlauf der Arbeit die Anweisungen zur Darstellung von Visualisierungen nur noch in begründeten Fällen explizit angegeben. Bei Interesse können gerne die detaillierten Jupyter Notebook Dokumente eingesehen werden, die dieser Arbeit beiliegen.

Datenanalyse: Menge der Benutzer pro Studiengang

```

1 # Ermittlung der Menge der Benutzer pro Studiengang
2 result = moodle_data.userid.groupby(moodle_data.Studiengang).nunique()
3 # Visualisierung der Menge der Benutzer pro Studiengang
4 chart = sns.barplot(x=result.index, y=result)

```

Listing 10: Menge der Benutzer pro Studiengang

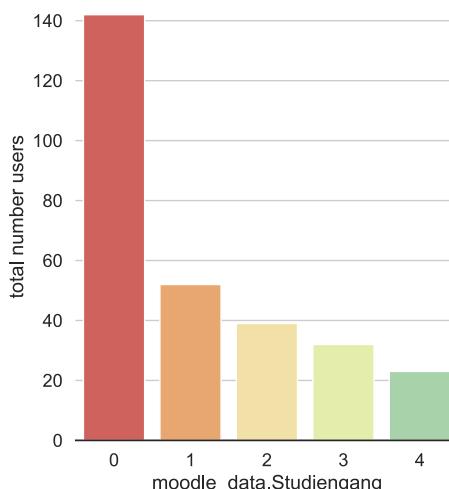


Abbildung 13.: Menge der Benutzer pro Studiengang

Evaluierung

Die Abbildung zur Menge der Benutzer pro Studiengang präsentierte nicht nur die reinen Zahlen, die auch die entsprechende Ergebnistabelle im vorheigen Abschnitt bereits auflistete. Sie verdeutlicht darüberhinaus auch sehr schnell die Größenverhältnisse zwischen den einzelnen Werten des Diagramms. Dies ist ein weiterer großer Vorteil gegenüber Ergebnistabellen, deren Aussagen sich durch analytische Überlegungen manchmal erst recht langsam erschließen.

Wie eingangs erwähnt, sind die hier gezeigten ersten Untersuchungen einfach und nur wenig umfangreich. Bei komplexeren Aufgabenstellungen wie sie im folgenden Kapitel zu lösen sind, sind die Phasen der Datenaufbereitung bzw. Datenanalyse und Evaluierung dagegen häufig in mehreren Schritten wiederholt zu durchlaufen.

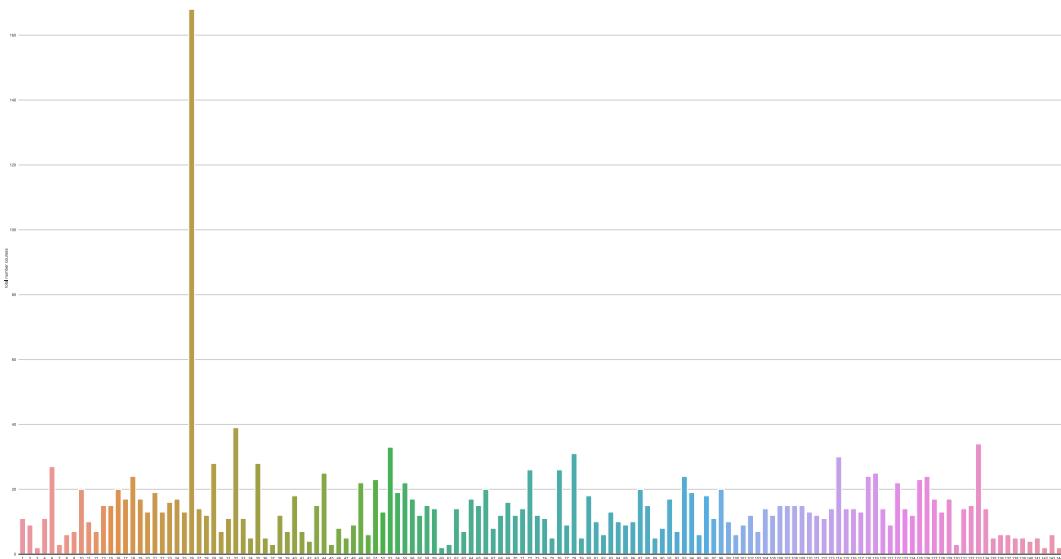
Datenanalyse: Menge der Kurse pro Benutzer

```

1 # Ermittlung der Menge der Kurse pro Benutzer
2 result = moodle_data.courseid.groupby(moodle_data.userid).nunique()
3 # Visualisierung der Menge der Kurse pro Benutzer
4 chart = sns.barplot(x=result.index, y=result)

```

Listing 11: Menge der Kurse pro Benutzer



2. Grundlagen

Betrachtet man das folgende Diagramm, so fällt erneut der Benutzer mit der userid 26 auf. Wie schon im Plot zur [Menge der Log-Einträge pro Benutzer](#) überragt sein Wert den der anderen bei weitem und man könnte hier bereits vermuten, dass es sich dabei nicht um einen Studenten, sondern um einen Angehörigen des Hochschulpersonals handelt.

3. Analyse

Hier steht die Einleitung zum Hauptteil dieser Arbeit mit Ausführungen zu dessen Bedeutung, Inhalt (aktivitäts- und zeitbezogene Analysen) und Aufbau. Abschließend sind hier Gedanken zur Notwendigkeit der Identifikation von Studenten als der zu untersuchenden Benutzergruppe zu formulieren und Überleitungen zu den weiteren Unterkapiteln mit zeitbezogenen Analysen herzustellen, die auf der Identifikation von Studenten aufbauen.

Übergeordnetes Ziel war dabei stets, auf Basis der aus den Analysen abgeleiteten Erkenntnisse typische Verhaltensweisen zu ermitteln, um die Studenten anschließend nach zeitlichen Bezügen einordnen zu können.

Bei der praktischen Umsetzung dieser Vorgabe sollte es im wesentlichen um die Beantwortung folgender grundlegender Fragen gehen:

1. Wie lassen sich Studenten nach der zeitlichen Lokalität der gezeigten Lern- und Kommunikationsaktivitäten einordnen?
2. Auf welche Weise lassen sich Studenten nach der Kontinuität ihres Handelns in verschiedene Gruppen unterscheiden?
3. Inwiefern kann man Studenten nach der Dynamik ihres Verhaltens beurteilen und hiernach in unterschiedliche Kategorien einteilen?

Hinweis geben auf die differenzierte Betrachtung von Lern- und Kommunikationsverhalten sowie nach Studiengängen.

3.1. Identifikation von Studenten

Im Grundlagenkapitel zur **Datenbasis** ist bereits mehrfach angeklungen, dass die im Rahmen dieser Arbeit zu betrachtenden Benutzer durchaus ganz verschiedenen Personengruppen angehören können.

3. Analyse

Neben den Studenten, deren Lern- und Kommunikationsverhalten ganz allein den Untersuchungsgegenstand darstellt, gibt es im Umfeld der Hochschule viele weitere Personen, deren Verhalten zwar möglicherweise im Kontext studentischer Aktivitäten eine gewisse Bedeutung zukommt, dieses für sich betrachtet in dieser Arbeit aber nicht weiter von Interesse sein sollte.

Dass die Identifikation von Studenten demnach eine notwendige Voraussetzung für die weiteren Untersuchungen darstellen würde, war also früh ersichtlich und so stellte sich damit auch unmittelbar die Frage, ob und wie sich Studenten mithilfe analytischer Untersuchungen des Datenbestands tatsächlich als eine ganz eigene Benutzergruppe identifizieren ließen.

Eine erste Überlegung war, die Benutzergruppen über die in Moodle definierten Rollen zu unterscheiden. Nach Informationen der Hochschule, wird in Moodle die Rolle eines Benutzers jedoch nur auf Kursebene festgelegt. Dies bedeutet, dass ein Benutzer, unabhängig von seinem offiziellen Status, in mehreren Kursen auch verschiedene Rollen einnehmen kann. Somit war schnell offensichtlich, dass sich diese Rollenzuweisung nicht als zuverlässiges Unterscheidungskriterium eignete.

Gesichert war hingegen der Umstand, dass in der Gesamtmenge der Benutzer insgesamt 75 *einzelne Studenten* enthalten sind.¹⁴ Diese vom Projektteam bestätigte Auskunft war zur Identifikation der Studenten wiederum nützlich, da sich daran schließlich die Qualität der Analyseergebnisse jederzeit messen lassen konnte.

3.1.1. Ermittlung des Benutzerstatus

Aber nicht nur die Qualität der Ergebnisse, sondern auch die des Datenbestands besitzt bei der Datenanalyse eine enorme Bedeutung. Daten müssen zwingend in einer entsprechend hohen Qualität vorliegen, damit im Nachhinein die gewonnenen Analyseergebnisse als fundiert gelten dürfen.

Wichtige Kriterien der Datenqualität sind u. a. die Vollständigkeit, die Richtigkeit sowie die Eindeutigkeit der Daten (Wang & Strong, 1996). Daneben ist aber auch die eigentliche Relevanz von grundlegendem Interesse, da die Einbeziehung nicht

¹⁴ Die genannte Menge an Studenten wurde im Rahmen einer Umfrage festgestellt, bei der Benutzer um ihr Einverständnis zur Nutzung ihrer Daten im Rahmen des DiSEA-Projekts gebeten wurden.

3. Analyse

relevanter Daten in eine Untersuchung die daraus resultierenden Ergebnisse stark negativ beeinflussen kann.

Mit Blick auf den Untersuchungsgegenstand dieser Arbeit – *das studentische Lern- und Kommunikationsverhalten* – wurde folglich mit dem Betreuerteam entschieden, nach einer Unterscheidung von Studenten und anderen Benutzern, jene Datensätze die sich nicht sicher auf studentische Aktivitäten beziehen, zu kennzeichnen und bei den anschließenden Untersuchungen gesondert zu behandeln.

Bedeutete die Identifikation der Studenten also die Grundlage für alle weiteren Analysen, so musste diese demnach zwingend ein hinreichend gesichertes Ergebnis erbringen. Die praktischen Schritte bei den Untersuchungen orientierten sich dabei erneut an dem in den Grundlagen vorgestellten [Vorgehensmodell](#).

Datenaufbereitung

Gegenstand der Untersuchung waren initial nur Datensätze mit einer userid > 0, d. h. es wurden nur Einzelbenutzer betrachtet ([s. Listing im Grundlagenkapitel](#)).

Datenanalyse: Untersuchungen verschiedener Tabellenmerkmale

Mehrere Überlegungen wie auch die Reflektion des eigenen Benutzerverhaltens als Student orientierten sich zunächst an der Frage, wie sich ein typisch studentisches Verhalten tatsächlich darstellen könnte und führten so zu einigen Untersuchungen über die Merkmale des Datenbestands, u. a. auch zum Merkmal *action*:

```
1 # Visualisierung der Mengen aller Actions in der Gesamtbeobachtung
2 chart = sns.histplot(data=moodle_data.action.sort_values(),
3                      stat='percent', color='#6DAEE2', alpha=1)
```

Listing 12: Mengen aller Actions in der Gesamtbeobachtung

Evaluierung

Während manche Betrachtungen gerade in zeitlicher Hinsicht auf den ersten Blick wenig aufschlussreiche Ergebnisse lieferten, fiel bei Untersuchung des Merkmals *action* sofort auf, dass Benutzer neben einem hohen Anteil an sent-Actions einen noch höheren Anteil an Werten vom Typ *viewed* aufwiesen. Mit einem Anteil von insgesamt über 80% bestimmten diese beiden Aktivitäten die Gesamtaktivität aller Benutzer im Untersuchungszeitraum.

3. Analyse

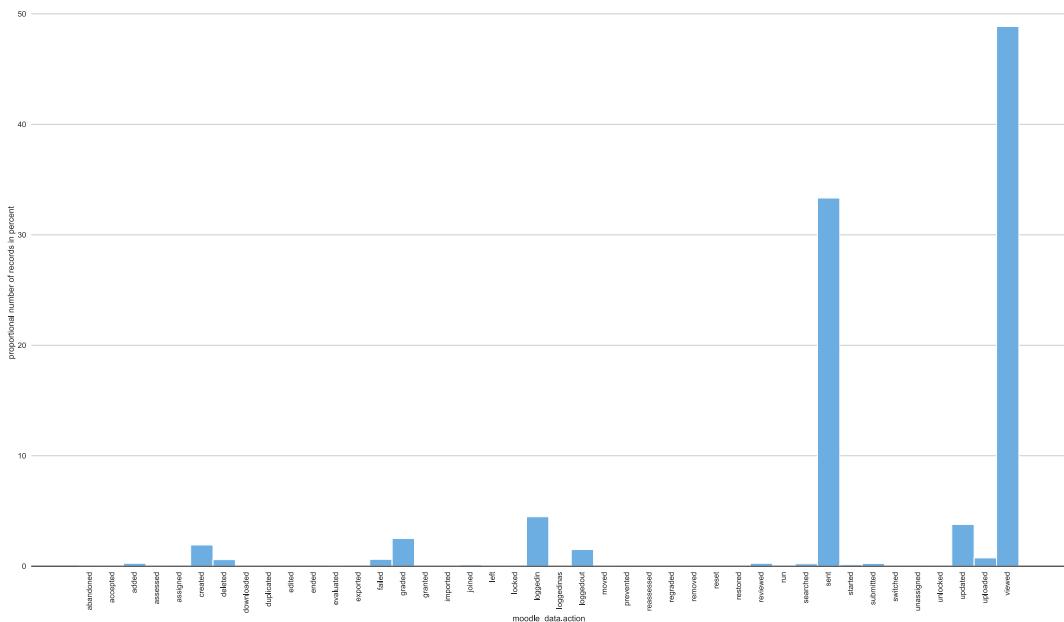


Abbildung 15.: Mengen aller Actions in der Gesamtbetrachtung ([s. Anhang](#))

Aus diesem Ergebnis nun schon ein typisch studentisches Verhalten abzuleiten war zwar nicht möglich, es widerlegte jedoch auch nicht direkt meine Vermutung, dass Studenten oft als Leser z. B. von Lehrmaterialien, Forumsdiskussionen oder Mitteilungen auftreten und ganz nebenbei deckte es sich ebenfalls weitgehend mit meinem eigenen Verhalten als Studenten.

Auch inspirierte das Ergebnis zu der Frage, wie sich gerade die Menge der viewed-Actions tatsächlich über das Semester hinweg auf die Benutzer verteilte.

Datenaufbereitung

Die Datenauswahl umfasste erneut alle Datensätze mit einer userid > 0, d.h. es wurden nur Einzelbenutzer betrachtet ([s. Listing im Grundlagenkapitel](#)).

Datenanalyse: Betrachtung der Menge an viewed-Actions pro Benutzer

```
1 md = moodle_data # Umbenennung zur kompakteren Darstellung des Codes
2 # Visualisierung der Menge der viewed-Actions pro Benutzer
3 chart = sns.countplot(x=md.userid[md.action == 'viewed'], alpha=1)
```

Listing 13: Menge der viewed-Actions pro Benutzer

Evaluierung

Wie im obigen Diagramm zu erkennen ist, gibt es einige Benutzer denen relativ hohe Mengen an viewed-Actions zuzuordnen sind. Gleichzeitig finden sich aber

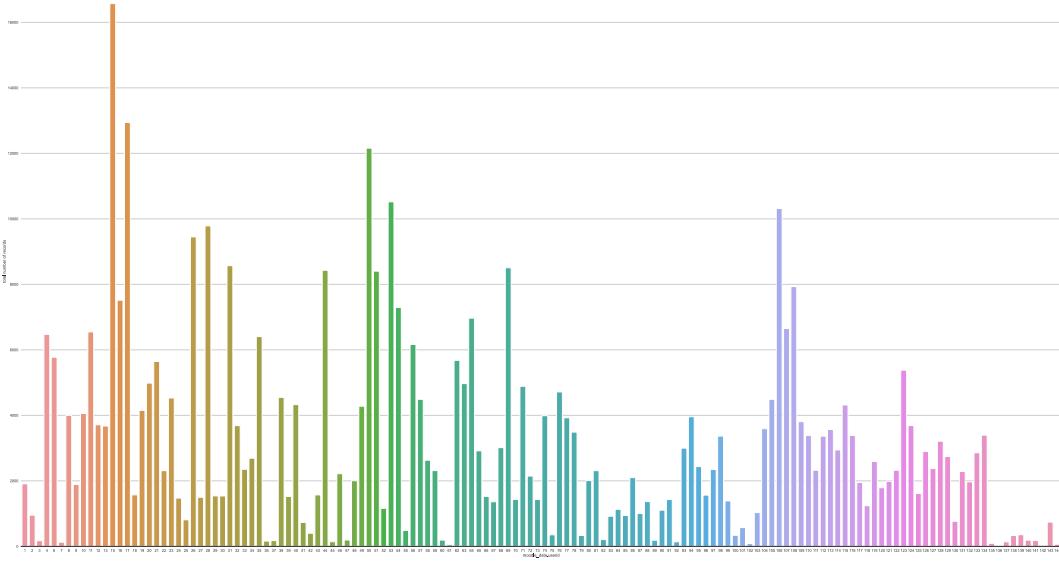


Abbildung 16.: Menge der viewed-Actions pro Benutzer ([s. Anhang](#))

auch Personen, die nur eine geringe Anzahl aufweisen. Ob und wie sich aus dieser einfachen Differenzierung vielleicht schon ein Hinweis ableiten lassen könnte auf ein echtes benutzertypisches Verhalten, war nun die spannende Frage.

Um diese Frage für die Gesamtheit aller Benutzer sicher beantworten zu können, war zum einen zu klären, welchen Anteil die viewed-Actions an der Gesamtaktivität der jeweiligen Benutzer tatsächlich hatte. Zum anderen war es aber auch notwendig, eine variable Vergleichsgröße zu definieren anhand derer es möglich war Benutzer beliebig ein- oder auszuschließen.

Auf diese Weise ließe sich dann auch konkret bestätigen oder widerlegen, ob die oben ermittelten Benutzer mit den hohen Mengen an Werten vom Typ viewed wirklich auch diejenigen waren, deren Verhalten maßgeblich durch die höheren viewed-Actions bestimmt war.

Aufschluss über all die Fragen gab schließlich die folgende SQL-Anweisung, die auf der gleichen Datenauswahl wie die vorausgehende Analyse basierte (s. WHERE-Klausel unten). Die Vergleichsgröße wurde dabei anfänglich mit einem viewed-Anteil von 50% an der Gesamtaktivität (s. HAVING-Klausel unten) definiert, da dies ziemlich genau dem zuvor ermittelten **Gesamtdurchschnitt** entsprach. Danach wurde sie in einem iterativen Prozess, begleitet von Einzelbenutzerbetrachtungen,

in mehreren Schritten angepasst.¹⁵

Datenanalyse: Anteile der viewed-Actions an der Gesamtaktivität

```

1  SELECT md1.userid,
2      COUNT(md1.action) AS 'all_actions',
3      (SELECT COUNT(md2.action) FROM moodle_data md2
4          WHERE md1.userid = md2.userid AND md2.action = 'viewed')
5          AS 'viewed_action',
6      (SELECT COUNT(md2.action) FROM moodle_data md2
7          WHERE md1.userid = md2.userid AND md2.action != 'viewed')
8          AS 'other_actions',
9      (SELECT COUNT(md2.action) FROM moodle_data md2
10         WHERE md1.userid = md2.userid AND md2.action = 'viewed') /
11      COUNT(action) AS 'percentage'
12  FROM moodle_data md1
13  WHERE userid > 0
14  GROUP BY md1.userid
15  HAVING percentage > 0.5
16  ORDER BY percentage DESC;

```

Listing 14: Anteil der viewed-Actions an der Gesamtaktivität

Evaluierung

Im Ergebnis der obigen SQL-Abfrage zeigten sich 99 Benutzer (ca. 70% der Gesamtbenutzeranzahl), bei denen die Menge der viewed-Actions mehr als die Hälfte der Gesamtaktivität ausmachte und die nun anhand von Stichproben exemplarisch zu prüfen waren. In einem stetigen Wechsel folgten dann weitere Abfragen immer mit angepasster Vergleichsgröße und weiteren Betrachtungen einzelner Benutzer.

Im Zuge dieses Vorgehens zeigten die zahlreichen Einzelanalysen, die ferner Art und Umfang weiterer Merkmale wie auch den zeitlichen Kontext betrachteten, dass sich die Trefferquote der SQL-Abfrage durch paralleles Anheben des Grenzwerts in der HAVING-Klausel sukzessive verbessern ließ. Es wurde dabei aber auch offensichtlich, dass hierdurch zunehmend mehr mutmaßliche Studenten ausgeschlossen wurden. *Bei einem Grenzwert von 0.8 wurde schließlich der iterative Prozess des stetigen Testens und Optimierens beendet: Mit 35 mutmaßlichen Studenten lag zwar ein sorgfältig getestetes und damit gesichertes Ergebnis vor, rein zahlenmäßig betrachtet war es aber unzureichend.*

Daneben sind bei den Einzelanalysen noch weitere Phänomene sichtbar geworden: Manche Benutzer unterschieden sich in ihren Kursprofilen, d. h. in der Art und der Menge ihrer Kurse deutlich, verhielten sich in Bezug auf andere aber wiederum

¹⁵ Siehe auch die zu dieser Arbeit beigefügten Jupyter Notebook Dokumente zu Einzelanalysen.

3. Analyse

userid	all_actions	viewed_action	other_actions	percentage
64	7544	6970	574	0.9239
53	11699	10520	1179	0.8992
40	4953	4328	625	0.8738
69	9756	8507	1249	0.8720
91	1641	1430	211	0.8714
104	4136	3592	544	0.8685
76	5434	4716	718	0.8679
94	4561	3958	603	0.8678
98	3894	3368	526	0.8649
87	1165	1006	159	0.8635
83	1084	922	162	0.8506
55	575	489	86	0.8504
66	1795	1526	269	0.8501
72	2526	2147	379	0.8500
13	4330	3675	655	0.8487
20	5909	4986	923	0.8438
68	3579	3015	564	0.8424
56	7335	6165	1170	0.8405
57	5361	4491	870	0.8377
52	1390	1162	228	0.8360
38	5478	4551	927	0.8308
51	10118	8404	1714	0.8306
70	1727	1434	293	0.8303
54	8813	7295	1518	0.8278
97	2861	2347	514	0.8203
71	5985	4889	1096	0.8169
65	3576	2918	658	0.8160
93	3685	3000	685	0.8141
96	1928	1566	362	0.8122
78	4300	3490	810	0.8116
49	5286	4280	1006	0.8097
58	3268	2632	636	0.8054
23	5634	4531	1103	0.8042
59	2885	2314	571	0.8021
44	10536	8430	2106	0.8001
...

99 rows in set (6,49 sec)

Abbildung 17.: Anteil der viewed-Actions an der Gesamtaktivität

recht ähnlich. Genau diese Besonderheit war interessanterweise aber auch bei anderen Benutzeraktivitäten zu beobachten. Auch hier schien es solche Unterschiede und Gemeinsamkeiten gleichzeitig zu geben, was zu dem Gedanken führte, dass gerade die genauere Betrachtung weiterer spezifischer Aktivitäten eine verbesserte Typisierung und folglich auch eine Unterscheidung von Studenten und Anderen ermöglichen könnte.

Die thematische Ausrichtung für die weiteren Schritte war damit klar. Fraglich war nur, ob an dieser Stelle wieder eine Gesamtbetrachtung aller Benutzer ratsam war oder ob nicht mittlerweile eine bessere Option bestünde. Dies war auch ein passender Moment, die praktischen Untersuchungen einmal zu pausieren und zu

reflektieren, wie bei einer Datenexploration methodisch sinnvoll vorzugehen ist.

Hier noch ein oder zwei Sätze zur explorativen Datenanalyse mit Literaturangaben bzw. Verweis auf das entsprechende Grundlagenkapitel ergänzen ...

Nach kurzer Überlegung stand fest, dass eine Betrachtung des gesamten Datenbestands und damit verbunden eine Rückkehr zum Startpunkt der Analysen zwar möglich wäre, die effiziente Nutzung bereits gewonnener Erkenntnisse als sicherer Ausgangspunkt für neue Schritte aber zu bevorzugen ist.

Datenaufbereitung

Gemäß den bei den durchgeführten Einzelanalysen erlangten Einsichten, wurden nun also für die weiteren Untersuchungen gezielt bestimmte Benutzer ausgewählt und deren Log-Einträge in neuen Datensets für Studenten und Andere (Others) zusammengefasst.

```

1 md = moodle_data # Umbenennung der Variable, um den Code zu verkürzen
2 records_students = [md[md.userid == 1], md[md.userid == 13],
3                      md[md.userid == 18], md[md.userid == 19],
4                      md[md.userid == 20], md[md.userid == 22],
5                      md[md.userid == 23], md[md.userid == 24],
6                      md[md.userid == 25], md[md.userid == 38]]
7 md_students = pd.concat(records_students)

```

Listing 15: Auswahl der Log-Einträge der Studenten

```

1 md = moodle_data # Umbenennung der Variable, um den Code zu verkürzen
2 records_others = [md[md.userid == 2], md[md.userid == 4],
3                     md[md.userid == 6], md[md.userid == 9],
4                     md[md.userid == 10], md[md.userid == 11],
5                     md[md.userid == 27], md[md.userid == 28],
6                     md[md.userid == 29], md[md.userid == 32]]
7 md_others = pd.concat(records_others)

```

Listing 16: Auswahl der Log-Einträge der Anderen

Anschließend wurden die spezifischen Aktivitäten der einzelnen Benutzergruppen ermittelt und ausgewertet sowie in einem gemeinsamen Datenset kombiniert.

```

1 # Ermittlung der Menge der Log-Einträge pro Action
2 students_actions = md_students.action.groupby(md.action).count()
3 others_actions = md_others.action.groupby(md.action).count()
4
5 # Erstellung eines kombinierten Datensets für Studenten und Andere
6 users_actions = pd.concat([students_actions, others_actions], axis=1,
7                           keys=['students', 'others']).sort_index()
8
9 # Ersetzung von NaN-Werten durch den Wert 0
10 users_actions = users_actions.fillna(0)
11
12 # Ausgabe des kombinierten Datensets

```

```
13 display(users_actions)
```

Listing 17: Konkatenation der Datensets von Studenten und Anderen

Die Tabelle unten zeigt im Ergebnis das fertig aufbereitete Datenset, das in der nachfolgenden Analyse dann noch einmal visualisiert und interpretiert wurde.

action	students	others
abandoned	0.0	2.0
accepted	28.0	3.0
added	21.0	403.0
created	392.0	2248.0
deleted	46.0	303.0
downloaded	2.0	170.0
duplicated	1.0	0.0
ended	4.0	6.0
evaluated	0.0	348.0
exported	0.0	4.0
graded	106.0	2304.0
granted	0.0	15.0
joined	127.0	26.0
left	15.0	20.0
moved	0.0	2.0
regraded	0.0	3.0
removed	2.0	32.0
restored	0.0	2.0
reviewed	94.0	93.0
searched	4.0	12.0
started	214.0	66.0
submitted	443.0	3.0
switched	0.0	16.0
updated	88.0	5106.0
uploaded	344.0	743.0
viewed	22718.0	26185.0

Abbildung 18.: Kombiniertes Datenset für Studenten und Andere

Datenanalyse: Menge der Log-Einträge pro Aktivität und Benutzergruppe

```
1 # Visualisierung der Menge der Log-Einträge pro Action
2 result = users_actions.stack().reset_index().set_index('action').
    rename(columns={'level_1': 'students', 0: 'others'})
3 chart = sns.barplot(x=result.index, y='others',
                      data=result, hue='students')
```

Listing 18: Menge der Log-Einträge pro Aktivität und Benutzergruppe

Die Visualisierung unten veranschaulicht noch einmal deutlich die bereits in der Ergebnistabelle sichtbaren Wertdifferenzen. Zu beachten ist, dass die Balken für die viewed-Actions beim Wert 5500 oben abgeschnitten wurde, um die Differenzen der anderen Werte in ihren Proportionen besser erkennen zu können.

Evaluierung

Wie das oben aufbereitete Datenset und die Grafik auf den ersten Blick zeigen, überragen z. B. bei den Werten *created*, *graded* und *updated* die Log-Einträge der Anderen

3. Analyse

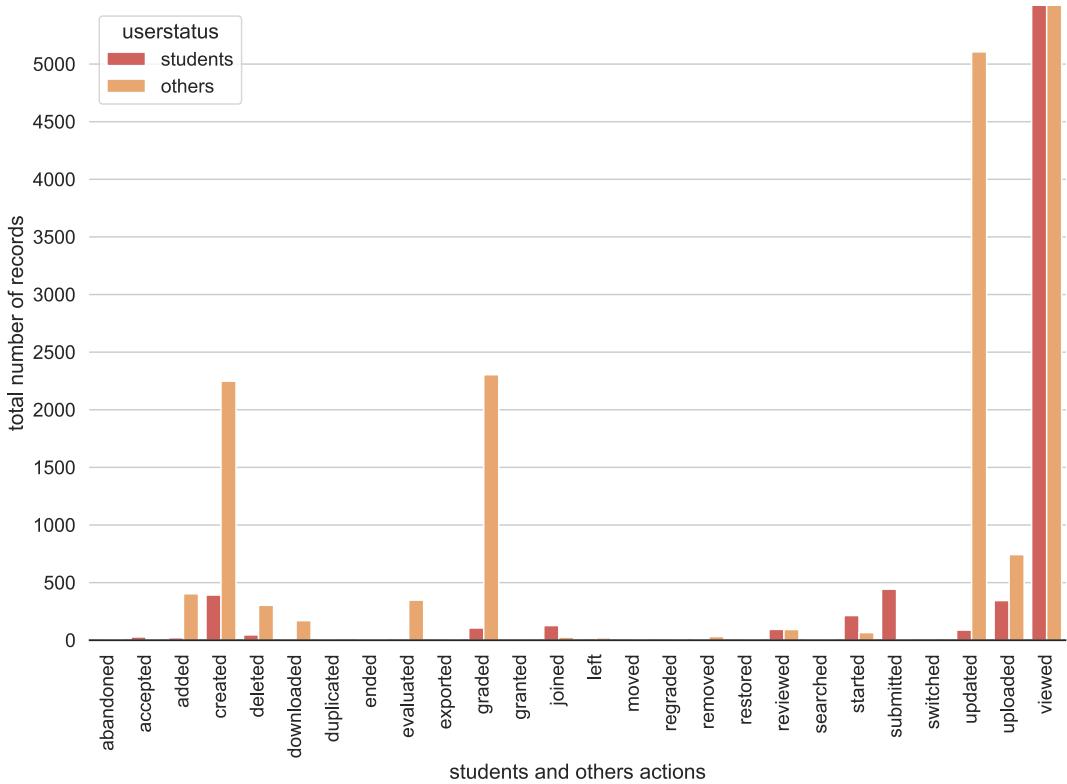


Abbildung 19.: Menge der Log-Einträge pro Aktivität und Benutzergruppe

die der Studenten um ein Vielfaches, während man erst bei genauerem Hinsehen erkennt, dass sich das Verhältnis beim Wert *submitted* andersherum darstellt.

Damit bestätigte also die Untersuchung mittels vordefinierter Benutzergruppen die zuvor formulierte Vermutung, dass sich Studenten und Andere durch die Art und den Umfang ihrer Aktivitäten unterscheiden.

Was nun zwangsläufig folgen musste, war die Beantwortung der Frage, ob und wie sich mit dieser Erkenntnis die Studenten im Gesamtkontext auch auf direktem Wege identifizieren ließen. Hierzu erschien es ratsam, die Mengen der Log-Einträge zu den einzelnen Aktivitäten erneut zu prüfen. Dabei kamen rasch auch noch die Werte *added*, *deleted* und *evaluated* in den Fokus, weil sie ebenfalls selbst eine gewisse Anzahl an Log-Einträgen besaßen, andererseits jedoch auch eine mindestens genauso beachtliche Mengendifferenz aufwiesen.

In dieser Phase waren noch einige weitere Untersuchungen notwendig. Insbesondere zum besseren Verständnis der Benutzeraktivitäten wurden wiederholt Einzelanalysen durchgeführt, bis schließlich deutlich wurde, dass die mögliche Lösung

des Problems vielleicht schon vorlag: *Die beabsichtigte direkte Identifikation von Studenten müsste, ähnlich dem Vorgehen bei der Betrachtung der viewed-Actions, über eine dem Gesamtkontext angemessene Gewichtung ausgewählter Aktivitäten herzustellen sein.*

Einzelnen oder in Gruppen zusammengefasst müssten die verschiedenen Mengen an Log-Einträgen zu den Aktivitäten wie Stellschrauben justiert werden können, um die Studenten aus der Gesamtmenge der Benutzer herauszufiltrern. Dabei sollte es von Vorteil sein, dass mit der Summe der anteiligen Mengen an added-, created-, deleted-, evaluated-, graded- und updated-Actions einerseits sowie der anteiligen Menge an submitted-Actions andererseits zwei Größen zur Verfügung standen, die grundsätzlich gegenläufig waren.

Für die praktische Umsetzung dieser Idee schien es am einfachsten, das vormals verwendete SQL-Statement zur Selektion von Benutzern mit einem hohen Anteil an viewed-Actions entsprechend zu modifizieren.

Datenaufbereitung

Die Datenauswahl umfasste erneut alle Datensätze mit einer userid > 0, d. h. es wurden nur Einzelbenutzer betrachtet (s. u. die WHERE-Klausel im SQL-Listing).

Die obige [SQL-Anweisung zur Betrachtung der viewed-Actions](#) wurde einsteils hinsichtlich der Unterabfragen geändert. Vielmehr wurden aber auch in dem oben beschriebenen iterativen Prozess die für die Selektion von Studenten relevanten Größen in der HAVING-Klausel angepasst.

Datenanalyse: Identifikation von Studenten

```

1  SELECT userid,
2      COUNT(action) AS "all_actions",
3      (SELECT COUNT(action) FROM moodle_data md2
4       WHERE md2.userid = md1.userid AND md2.action = "added")
5       AS "added",
6      (SELECT COUNT(action) FROM moodle_data md2
7       WHERE md2.userid = md1.userid AND md2.action = "created")
8       AS "created",
9      (SELECT COUNT(action) FROM moodle_data md2
10      WHERE md2.userid = md1.userid AND md2.action = "deleted")
11      AS "deleted",
12      (SELECT COUNT(action) FROM moodle_data md2
13      WHERE md2.userid = md1.userid AND md2.action = "evaluated")
14      AS "evaluated",
15      (SELECT COUNT(action) FROM moodle_data md2
16      WHERE md2.userid = md1.userid AND md2.action = "graded")

```

```

17      AS "graded",
18      (SELECT COUNT(action) FROM moodle_data md2
19       WHERE md2.userid = md1.userid AND md2.action = "submitted")
20      AS "submitted",
21      (SELECT COUNT(action) FROM moodle_data md2
22       WHERE md2.userid = md1.userid AND md2.action = "updated")
23      AS "updated"
24 FROM moodle_data md1
25 WHERE userid > 0
26 GROUP BY userid
27 HAVING ((added + created + deleted + evaluated +
28           graded + updated) < (0.25 * all_actions))
29   AND (submitted > (0.001 * all_actions));

```

Listing 19: Identifikation von Studenten

Evaluierung

Durch sorgfältiges Testen mittels Einzelanalysen und Optimieren der Vergleichsgrößen im stetigen Wechsel, *konnten schließlich die in der obigen Ergebnistabelle angezeigten 72 Benutzer ausreichend sicher als Studenten identifiziert werden.*

Ob es sich bei diesen 72 Studenten nun um eine exakte Teilmenge der bei einer Umfrage sicher ermittelten 75 Studenten handelt oder im Ergebnis eventuell auch Studenten erfasst wurden, die eventuell erst später auf Anfrage¹⁶ der Beobachtung ihres Verhalten zugestimmt haben, wäre bei Bedarf noch zu prüfen, soll hier für die weiteren Untersuchungen in dieser Arbeit jedoch nicht von Interesse sein.

Weitere Überprüfungen in Form neuer Einzelbetrachtungen könnten allerdings notwendig werden, sollte die beschriebene Methodik z. B. bei einer Untersuchung auf einem ganz neuen Datenbestand doch Fehler aufweisen. Ein grundsätzliches *Overfitting*, d. h. eine zu genaue Anpassung eines statistischen Modells an gewisse Besonderheiten eines Datensets aus der eine mangelhafte Übertragbarkeit resultiert (Dietterich, 1995), müsste bei der Einfachheit der gewählten Vergleichsgrößen aber auszuschließen sein.

3.1.2. Kennzeichnung des Benutzerstatus

Um im weiteren Verlauf der Analysen die Auswahl der identifizierten Studenten zu vereinfachen und damit auch den gesamten Arbeitsprozess zu beschleunigen, wurde entschieden, die identifizierten Studenten durch ein neues Tabellenmerkmal dauerhaft zu kennzeichnen.

¹⁶ Um die im Rahmen der Umfrage erfassten Daten um einen größeren Kontext zu ergänzen, wurden Benutzer auch direkt kontaktiert, ohne jedoch deren offiziellen Status zu dokumentieren.

3. Analyse

userid	all_actions	added	created	deleted	evaluated	graded	submitted	updated
1	3865	0	43	0	0	0	12	20
13	4330	2	40	2	0	15	51	11
18	1978	2	17	1	0	0	24	14
19	5823	2	77	10	0	24	75	11
20	5909	3	58	3	0	19	55	10
22	2932	1	26	0	0	0	22	5
23	5634	5	76	3	0	35	106	12
24	2444	0	17	36	0	0	13	3
25	1133	6	21	0	0	0	16	2
38	5478	0	46	5	0	13	94	10
40	4953	0	43	9	0	9	44	21
43	2068	1	5	0	0	2	8	4
49	5286	6	51	5	0	57	97	77
51	10118	1	49	2	0	17	58	157
52	1390	2	23	0	0	0	18	13
53	11699	2	35	2	0	10	34	46
54	8813	1	57	16	0	22	63	192
56	7335	3	51	2	0	63	164	71
57	5361	2	74	0	0	26	89	31
58	3268	0	27	0	0	8	27	104
59	2885	1	50	4	0	11	50	18
60	298	0	0	0	0	0	4	0
62	7606	4	61	0	0	21	72	10
64	7544	2	42	0	0	8	35	12
65	3576	1	49	1	0	8	47	13
66	1795	0	25	13	0	1	17	5
67	1788	4	29	2	0	7	29	31
68	3579	2	41	0	0	26	72	6
69	9756	1	63	0	0	16	78	15
70	1727	0	13	6	0	9	18	6
71	5985	1	68	0	0	16	72	21
72	2526	1	5	0	0	3	7	2
73	1929	0	3	0	0	0	3	0
76	5434	1	12	0	0	2	12	0
78	4300	1	35	2	0	3	23	17
80	2611	2	47	0	0	2	24	48
83	1084	1	11	0	0	0	10	1
87	1165	0	6	0	0	1	8	0
91	1641	4	19	0	0	0	23	2
93	3685	7	42	3	0	26	41	102
94	4561	3	27	2	0	49	78	25
96	1928	2	16	0	0	9	24	4
97	2861	0	25	0	0	36	60	89
98	3894	6	37	3	0	12	36	14
99	1883	1	22	0	0	11	26	10
102	112	0	0	0	0	0	1	0
104	4136	1	67	0	0	11	57	20
105	5887	2	85	2	0	16	70	31
107	8751	2	167	33	0	0	11	358
109	5774	4	193	144	0	0	10	387
111	3577	2	141	32	0	1	10	292
112	4486	4	145	31	0	1	11	351
113	6108	2	254	67	0	0	9	377
115	5488	4	166	29	0	1	11	300
116	5717	4	191	16	0	0	10	328
117	3466	3	162	37	0	0	9	329
119	3616	1	87	2	0	0	9	244
120	2902	1	175	10	0	0	5	284
122	3564	1	150	15	0	0	8	288
123	6896	3	134	56	0	0	9	358
124	5505	2	162	22	0	0	9	296
125	2669	1	108	8	0	0	6	219
126	4311	1	171	80	0	0	5	297
127	3250	2	78	17	0	0	5	264
128	4309	3	134	41	0	0	8	311
129	3803	0	114	57	0	0	8	303
131	3748	33	162	17	0	1	7	276
132	2973	2	112	19	0	0	6	279
134	4629	2	146	22	0	0	12	304
136	33	0	0	0	0	0	2	0
142	10	0	0	0	0	0	1	0
143	1387	0	11	0	0	0	4	2

Abbildung 20.: Identifikation von Studenten

Hierzu wurden zunächst die Ergebnisse aus der Abfrage zur Identifikation von Studenten in eine neue Tabelle *moodle_data_students* übernommen. Das vorherige SQL-Statement war hierfür nur um zwei Zeilen Code zu ergänzen:

Erstellen der neuen Tabelle moodle_data_students

```

1 CREATE TABLE moodle_data_students
2 AS
3 /*
4 SQL-Statement zur Identifikation von Studenten
5 */

```

Listing 20: Erstellen der neuen Tabelle moodle_data_students

Hiernach wurden in der Relation *moodle_data* die neuen Merkmale *userstatus* und *relateduserstatus* mit dem Default-Wert *other* eingefügt, und dieser in einem letzten Schritt entsprechend den folgenden Anweisungen angepasst (die letzte Anweisung dient einer einfacheren Selektion von Aktivitäten ohne Personenbezug):

Kennzeichnung von Studenten

```

1 UPDATE moodle_data SET userstatus = 'student'
2 WHERE userid IN (SELECT userid FROM moodle_data_students);
3
4 UPDATE moodle_data SET relateduserstatus = 'student'
5 WHERE relateduserid IN (SELECT userid FROM moodle_data_students);
6
7 UPDATE moodle_data SET relateduserstatus = 'none'
8 WHERE relateduserid = 0;

```

Listing 21: Kennzeichnung von Studenten

Abschließende Prüfungen der durchgeführten Änderungen ergaben das erwartete Resultat: Alle Datensätze mit einer userid eines zuvor erkannten Studenten wurden vollständig und richtig aktualisiert.

Überprüfung der Änderungen auf Vollständigkeit

```

1 SELECT DISTINCT userid FROM moodle_data
2 WHERE userstatus = 'student';

```

Listing 22: Überprüfung der Änderungen auf Vollständigkeit

```

+-----+
| userid |
+-----+
|      1 |
|     13 |
|     18 |
|     ... |
|    136 |
|    142 |
|    143 |
+-----+
72 rows in set (2,39 sec)

```

Abbildung 21.: Überprüfung der Änderungen auf Vollständigkeit

Überprüfung der Änderungen auf Richtigkeit

```

1 SELECT * FROM moodle_data
2 WHERE (userstatus != 'student')
3   AND (userid IN (SELECT userid FROM moodle_data_students));

```

Listing 23: Überprüfung der Änderungen auf Richtigkeit

Empty set (0,40 sec)

Abbildung 22.: Überprüfung der Änderungen auf Richtigkeit

3.1.3. Zusammenfassung

In diesem Teil der Arbeit wurde gezeigt, wie rein datenorientiert eine hinreichend gesicherte Identifikation von Studenten auf Basis der bereitgestellten Moodle-Daten möglich ist. Anhand der beschriebenen Überlegungen, der durchgeführten Betrachtungen und der Auswertungen wurde detailliert der Weg beschrieben, der schließlich zur Lösung des Problems geführt hat. Schritt für Schritt wurden dabei folgende Auswahlkriterien ermittelt:

1. Studenten werden nur als Einzelbenutzer betrachtet, d. h. sie müssen zuvor der Beobachtung ihres Verhaltens zugestimmt haben und dürfen außerdem nicht im Bachelor-Studiengang Medieninformatik Online aktiv gewesen sein.
2. Studenten verfügen im Vergleich zu Anderen über einen relativ hohen Anteil an viewed- und submitted-Actions.
3. Studenten besitzen im Vergleich zu Anderen einen relativ geringen Anteil an added-, created-, deleted-, evaluated-, graded- und updated-Actions.

In einem iterativen Prozess wurden die anteiligen Mengen der genannten Actions mithilfe bestimmter Faktoren wiederholt gewichtet und die daraus resultierenden Ergebnisse anhand von Einzelanalysen¹⁷ exemplarisch geprüft bis schließlich eine Menge von insgesamt 72 Studenten bestätigt werden konnte.

Abschließend wurden die identifizierten Studenten mit ihren charakteristischen Aktivitätsprofilen in einer eigenen Tabelle zusammengefasst und im Gesamtdaten-

¹⁷ Siehe auch die zu dieser Arbeit beigefügten Jupyter Notebook Dokumente zu Einzelanalysen.

bestand eindeutig gekennzeichnet, so dass sie zur Betrachtung ihres Verhaltens nun unmittelbar zur Verfügung standen.

3.2. Konkretisierung der zu untersuchenden Datenbasis

Nachdem zuvor die Identität von Studenten anhand ihrer Aktivitäten festgestellt werden konnte, sollte im weiteren Verlauf der Arbeit deren Verhalten in zeitlicher Hinsicht untersucht und ausgewertet werden.

Als Datenbasis für die anstehenden zeitbezogenen Analysen sollte erneut die *viewed*-Action dienen können, die im vorhergehenden Kapitel zur [Identifikation von Studenten](#) wegen ihres hohen Anteils schon als relevante Größe zur Bestimmung studentischen Verhaltens in Erwägung gezogen wurde.

3.2.1. Betrachtung von *viewed*-Action und *viewed*-Events

Der Gedanke lag zwar sehr nahe, dass sich die *viewed*-Action demnach auch sehr gut für die Analyse eines spezifischen Lernverhaltens als Datenbasis eignen müsste, dieses sollte aber dennoch noch einmal geprüft werden. Grund hierfür war nicht zuletzt die Überlegung, dass für die Untersuchung des Kommunikationsverhaltens ebenfalls eine solide Grundlage gebraucht wurde, und es schien fraglich, ob hierfür ebenfalls die *viewed*-Action geeignet sei.

Eine Antwort auf die Fragen gab das nachfolgende SQL-Statement, mit dessen Hilfe sich die zur *viewed*-Action korrespondierenden *viewed*-Events ermitteln ließen.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst eine *viewed*-Action initiiert haben, also auch solche die mit einer *viewed*-Action einer anderen Person in Beziehung standen (s. u. die WHERE-Klausel im SQL-Listing).

Datenanalyse: Ermittlung korrespondierender *viewed*-Events

```
1 SELECT eventname, COUNT(eventname) AS "total_number_records"
2 FROM moodle_data
3 WHERE (userstatus = 'student' OR relateduserstatus = 'student')
4 AND action = 'viewed'
5 GROUP BY eventname
6 ORDER BY total_number_records DESC;
```

Listing 24: Ermittlung korrespondierender viewed-Events

Die Entscheidung, für eine genauere Betrachtung des studentischen Verhaltens die Werte des Merkmals *eventname* heranzuziehen, ergab sich u. a. aus Beobachtungen in vorbereitenden Tests und zahlreichen Einzelanalysen. Hierbei wurde ersichtlich, dass die Eventnames nicht nur sprechender und präziser waren als die in Beziehung stehenden Werte der Merkmale *objecttable* und *course_module_type*. Sie gaben vielmehr auch einen praktischen Hinweis auf die im Moodle-System verwendeten URLs und ließen so erahnen, welche Moodle-Seiten mit den bezeichneten events sehr wahrscheinlich in Verbindung stehen.

eventname	total_number_records	
\core\event\course_viewed	69507	L
...	...	
\mod_resource\event\course_module_viewed	20113	L
\mod_assign\event\course_module_viewed	15088	L
\mod_forum\event\discussion_viewed	14237	K
\mod_quiz\event\attempt_viewed	13605	L
\mod_forum\event\course_module_viewed	12146	K
...	...	
\mod_url\event\course_module_viewed	6636	L
...	...	
\mod_quiz\event\course_module_viewed	3198	L
...	...	
\mod_page\event\course_module_viewed	2878	L
\core\event\message_viewed	2036	K
\mod_wiki\event\course_module_viewed	1790	L
\mod_wiki\event\page_viewed	1568	L
...	...	
\mod_choice\event\course_module_viewed	1336	L
...	...	
\mod_folder\event\course_module_viewed	1186	L
...	...	
\mod_glossary\event\course_module_viewed	920	L
...	...	
\mod_workshop\event\course_module_viewed	518	L
...	...	
\mod_bigbluebuttonbn\event\recording_viewed	388	L
...	...	
\mod_chat\event\course_module_viewed	119	K
...	...	
\mod_chat\event\sessions_viewed	66	K
...	...	

62 rows in set (0,39 sec)

Abbildung 23.: Ermittlung korrespondierender viewed-Events

Evaluierung

Die Ergebnistabelle zeigt mit 19 Eventnames nur einen Ausschnitt von etwa einem

Drittel aller Werte, die mit dem Wert *viewed* des Merkmals *action* korrespondierten. Nicht angezeigt und damit auch bei weiteren Untersuchungen nicht berücksichtigt wurden dagegen beispielsweise Eventnames wie `\core\event\dashboard_viewed`, `\mod_assign\event\submission_status_viewed` oder `\core\event\user_profile_viewed`, die nicht oder nur wenig konkret auf die für weitere Analysen relevanten Lern- und Kommunikationsaktivitäten hindeuteten.

Betrachtete man die Mengen der in der Tabelle mit *L* markierten Ergebnissezeilen, die mit sehr hoher Wahrscheinlichkeit ein studentisches Lernverhalten implizierten, ergaben diese in Summe einen Anteil von 59,67% an der Gesamtmenge aller Datensätze, die mit *viewed*-Events in Beziehung stehen. *Dies sollte als solide Basis für weitere Betrachtungen des Lernverhaltens ausreichend sein.*

3.2.2. Entscheidung für *viewed*-Events als Grundlage

Fraglich war jedoch, ob auch die in der Tabelle mit *K* gekennzeichneten *viewed*-Events allein diejenigen waren, die das studentische Kommunikationsverhalten im wesentlichen bestimmten, oder ob nicht die studentische Kommunikation maßgeblich auch durch *sent*-Actions bestimmt war, die, wie im vorherigen Kapitel bereits erwähnt, über das gesamte Semester hinweg ebenfalls sehr hohe Zahlen aufwiesen. Aufschluss hierüber sollte die nachfolgende SQL-Abfrage geben.

Datenaufbereitung

Die Datenauswahl wurde analog zu der vorhergehenden Analyse getroffen, nur wurden dieses mal Studenten betrachtet, die selbst eine *sent*-Action initiiert haben oder mit einer *sent*-Action einer anderen Person in Beziehung standen (s.u. die WHERE-Klausel im SQL-Listing).

Datenanalyse: Ermittlung korrespondierender sent-Events

```

1  SELECT eventname, COUNT(eventname) AS "total_number_records"
2  FROM moodle_data
3  WHERE (userstatus = 'student' OR relateduserstatus = 'student')
4      AND action = 'sent'
5  GROUP BY eventname
6  ORDER BY total_number_records DESC;

```

Listing 25: Ermittlung korrespondierender sent-Events

3. Analyse

eventname	total_number_records
\core\event\notification_sent	38762
\core\event\message_sent	1255
\core\event\group_message_sent	176
\mod_chat\event\message_sent	20

4 rows in set (0,34 sec)

Abbildung 24.: Ermittlung korrespondierender sent-Events

Im Ergebnis zeigte sich, dass neben den hohen Zahlen an *notification*-Events, die sich aber nur auf automatisch generierte Nachrichten des Moodle-Systems¹⁸ bezogen, nur relativ wenige andere *sent*-Events protokolliert wurden.

Dies bedeutete im Umkehrschluss, dass die vorab ermittelten viewed-Events den weitaus größten Teil des studentischen Kommunikationsverhaltens abbildeten und die damit in Beziehung stehenden Datensätze fortan als Grundlage weiterer Untersuchungen dienen konnten.

3.3. Lokalität des Lern- und Kommunikationsverhaltens

Unter Bezugnahme auf die im vorausgegangenen Abschnitt ermittelte Datenbasis sollte es nun darum gehen, die erste der drei eingangs formulierten Grundfragen zu beantworten: *Hiernach war zu analysieren, ob und wie sich die Gesamtmenge der identifizierten Studenten hinsichtlich des zeitlichen Auftretens ihrer Aktivitäten klassifizieren ließen.*

...

An dieser Stelle folgen noch Analysen zu möglichen Unterschieden von Lernverhalten und Kommunikationsverhalten sowie Studiengängen.

3.3.1. Vergleich des Lern- und Kommunikationsverhaltens

...

3.3.2. Vergleich der Studiengänge

...

¹⁸ Siehe auch die Moodle Documentation (Moodle, 2022): [Moodle Messaging Notifications, 06/2022](#)

3.4. Kontinuität des Lern- und Kommunikationsverhaltens

Auf Grundlage der im Abschnitt zur [Konkretisierung der Datenbasis](#) ermittelten viewed-Events sollten nun weitere Untersuchungen erfolgen: *Insbesondere sollte es dabei um die Beantwortung der Frage gehen, wie man anhand einer bestimmten Kontinuität der Lern- und Kommunikationsaktivitäten die Menge der Studenten in verschiedene Gruppen einteilen konnte.*

Diese eine Frage zog sogleich zwei weitere Fragen nach sich, deren Beantwortung erforderlich war, um die Studenten schließlich typisieren zu können:

- Nach welchem zeitlichen Maßstab war die Kontinuität zu untersuchen?
- Nach welchen Vergleichsgrößen war die Kontinuität zu messen?

3.4.1. Bestimmung des zeitlichen Maßstabs

Betrachtungen auf verschiedenen zeitlichen Ebenen, insbesondere auf Wochen- und Tagessicht wie bei der folgenden Analyse, haben gezeigt, dass sich das studentische Verhalten über das Semester hinweg keineswegs konstant präsentierte.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst ein viewed-Event initiiert haben, also auch solche die mit einem viewed-Event einer anderen Person in Beziehung standen (s. das nachfolgende Listing zur Ergänzung des Merkmals *behaviour* und die Auswahl spezifischer Werte dieses Merkmals im anschließenden Listing zur Visualisierung).

```

1  md['behaviour'] = 'other'
2  md.loc[(md.eventname == r'\core\event\course_viewed'),
3      ['behaviour']] = 'learning'
4  md.loc[(md.eventname == r'\mod_resource\event\course_module_viewed'),
5      ['behaviour']] = 'learning'
6  md.loc[(md.eventname == r'\mod_assign\event\course_module_viewed'),
7      ['behaviour']] = 'learning'
8  md.loc[(md.eventname == r'\mod_quiz\event\attempt_viewed'),
9      ['behaviour']] = 'learning'
10 md.loc[(md.eventname == r'\mod_url\event\course_module_viewed'),
11      ['behaviour']] = 'learning'
12 md.loc[(md.eventname == r'\mod_quiz\event\course_module_viewed'),
13      ['behaviour']] = 'learning'
14 md.loc[(md.eventname == r'\mod_page\event\course_module_viewed'),
15      ['behaviour']] = 'learning'
16 md.loc[(md.eventname == r'\mod_wiki\event\course_module_viewed'),
17      ['behaviour']] = 'learning'
```

3. Analyse

```
18 md.loc[(md.eventname == r'\mod_wiki\event\page_viewed'),  
19     ['behaviour']] = 'learning'  
20 md.loc[(md.eventname == r'\mod_choice\event\course_module_viewed'),  
21     ['behaviour']] = 'learning'  
22 md.loc[(md.eventname == r'\mod_folder\event\course_module_viewed'),  
23     ['behaviour']] = 'learning'  
24 md.loc[(md.eventname == r'\mod_glossary\event\course_module_viewed'),  
25     ['behaviour']] = 'learning'  
26 md.loc[(md.eventname == r'\mod_workshop\event\course_module_viewed'),  
27     ['behaviour']] = 'learning'  
28 md.loc[(md.eventname == r'\mod_bigbluebuttonbn\event\recording_viewed'),  
29     ['behaviour']] = 'learning'  
30 md.loc[(md.eventname == r'\mod_forum\event\course_module_viewed'),  
31     ['behaviour']] = 'communication'  
32 md.loc[(md.eventname == r'\mod_forum\event\discussion_viewed'),  
33     ['behaviour']] = 'communication'  
34 md.loc[(md.eventname == r'\core\event\message_viewed'),  
35     ['behaviour']] = 'communication'  
36 md.loc[(md.eventname == r'\mod_chat\event\course_module_viewed'),  
37     ['behaviour']] = 'communication'  
38 md.loc[(md.eventname == r'\mod_chat\event\sessions_viewed'),  
39     ['behaviour']] = 'communication'
```

Listing 26: Ergänzung des Merkmals *behaviour*

Durchgeführt wurde die Analyse hier also mit Bezug auf das gesamte studentische Lern- und Kommunikationsverhalten in Form einer Gesamtbetrachtung über den ganzen Zeitraum von 235 Tagen (s. Anzahl der *bins* im folgenden Listing).

Datenanalyse: Verteilung der Log-Einträge im Gesamtzeitraum pro Tag

```
1 # Visualisierung der Menge der Log-Einträge über den Gesamtzeitraum  
2 chart =  
3     sns.histplot(data=  
4         md.timecreated[(md['behaviour'] == 'learning') |  
5             (md['behaviour'] == 'communication')],  
6         bins=235, color=colors_general[4], alpha=1)
```

Listing 27: Verteilung der Log-Einträge im Gesamtzeitraum pro Tag

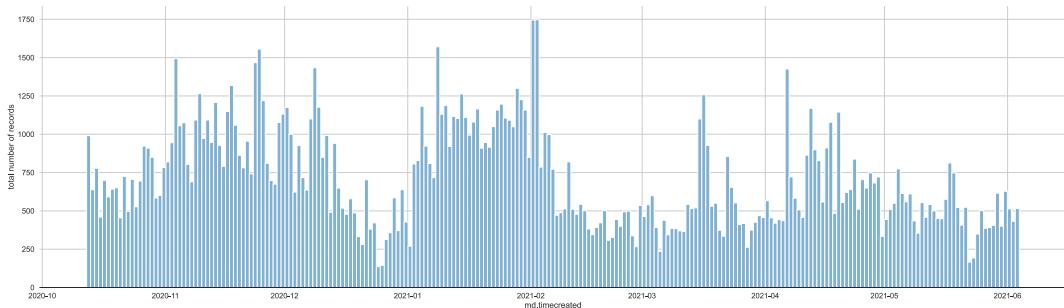


Abbildung 25.: Verteilung der Log-Einträge pro Tag ([s. Anhang](#))

Evaluierung

Auf den ersten Blick ließ das Histogramm in der Gesamtübersicht keine Kontinuität i. S. eines konstanten Verlaufs erkennen. Aufallend waren primär die zu erwarten-

3. Analyse

den starken Rückgänge an den Feiertagen oder zwischen den Prüfungszeiträumen sowie die höheren Mengen an Log-Einträgen vor und während der Prüfungen. Des weiteren deutete das stetige Auf und Ab der Aktivitäten eher auf ein sprunghaftes Verhalten hin.

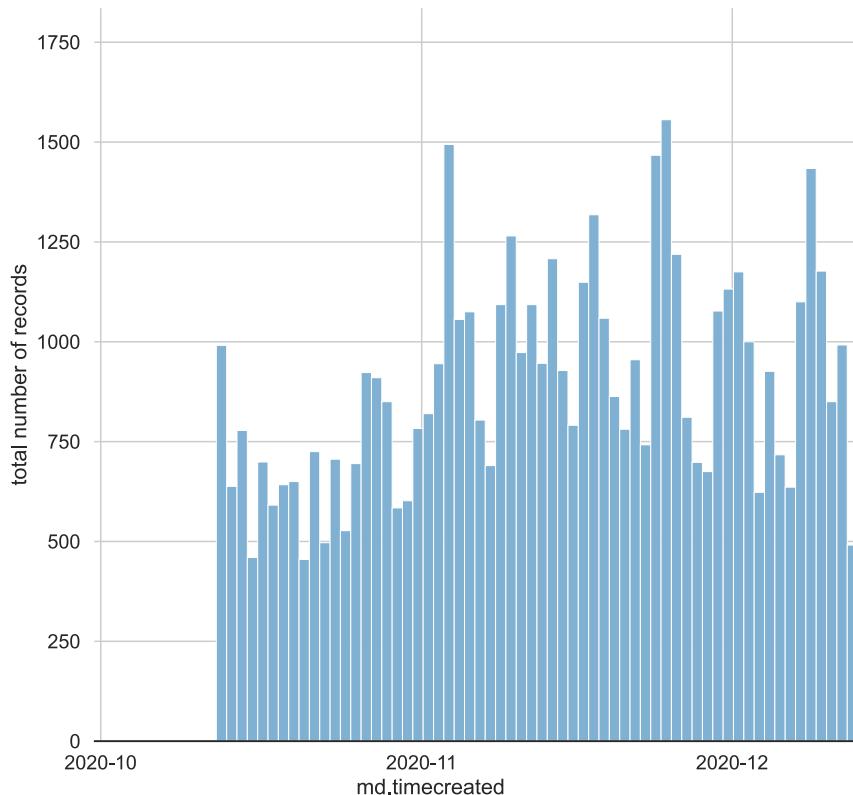


Abbildung 26.: Verteilung der Log-Einträge pro Tag (Ausschnitt)

Betrachtete man die Visualisierung jedoch etwas genauer, so ließ sich vor allem im Zeitraum bis Mitte Dezember beobachten, dass die Änderungen in der Verteilung doch nicht willkürlich waren. Vielmehr folgten sie einem etwas unregelmäßigen Muster, wonach sich steile Anstiege und mehrere Tage hoher Aktivität mit einem kürzeren starken Rückgang stetig abwechselten. Der Gedanke lag nahe, dass sich darin ein periodischer Ablauf studentischer Aktivitäten zeigte, der auf Wochensicht durchaus eine gewisse Kontinuität besaß und der sich beiläufig bemerkte auch mit dem allgemein üblichen Kursgeschehen deckte.

Bei Rücksichtnahme auf die Abläufe in den Fachkursen mit den oft wöchentlich organisierten Lehrveranstaltungen und Arbeitsaufträgen, erschien es also nicht nur

ratsam, sondern geradezu folgerichtig, die Kontinuität studentischer Aktivitäten ebenfalls nach einem wöchentlichen Maßstab zu untersuchen.

Schien an diesem Punkt bereits geklärt, in welchem Zeitrahmen die Kontinuität des Verhaltens zu untersuchen sein sollte, stellte sich umgehend die Frage, nach welchen Kriterien diese zu bemessen sei.

Da sich aus diesbezüglichen Beobachtungen nur vage Anhaltspunkte ergaben, und auch nicht die Gefahr bestand, aufgrund eines schlecht gewählten Ansatzes gleich ein falsche Aussage abzuleiten, sollte es genügen die Untersuchungen nach allgemeinen Überlegungen zu beginnen.

Demzufolge sollten in einer ersten Phase aus den zur Verfügung gestellten Daten zunächst ein paar weitere Informationen ermittelt werden, die Aufschluss gaben über die grundlegende Art und Weise des studentischen Verhaltens. Insbesondere sollte herausgefunden werden, wieviele Log-Einträge einem Studenten überhaupt zugeordnet waren und in wie vielen Wochen dieser aktiv war.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst ein *viewed*-Event initiiert haben, also auch solche die mit einem *viewed*-Event einer anderen Person in Beziehung standen (s. das nachfolgende Listing).

```
1 # Ermittlung der Menge der Log-Einträge pro Student
2 loggings_user = pd.Series(md.userid[
3     (md.userstatus == 'student') & ((md['behaviour'] == 'learning') |
4     (md['behaviour'] == 'communication'))].groupby(md.userid).count(),
5     name='loggings')
```

Listing 28: Menge der Log-Einträge pro Student

Das nachfolgende Diagramm visualisiert anschaulich die Menge der Log-Einträge pro Student über den gesamten Untersuchungszeitraum.

Datenanalyse: Menge der Log-Einträge pro Student

```
1 # Visualisierung der absoluten Menge der Log-Einträge pro Student
2 chart = sns.barplot(x=loggings_user.index.astype(int),
3                      y=loggings_user, color=colors_general[4])
```

Listing 29: Menge der Log-Einträge pro Student

3. Analyse

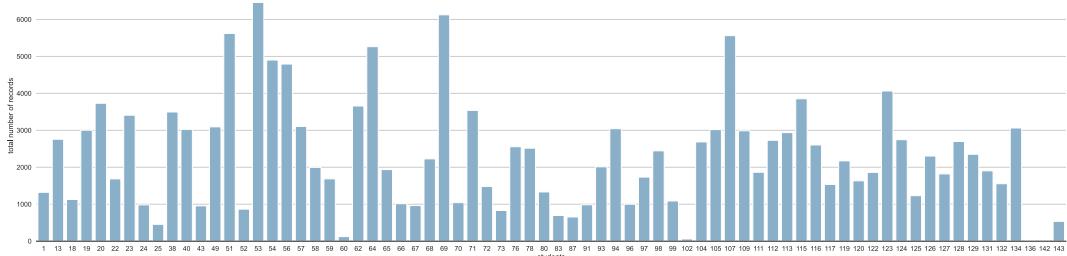


Abbildung 27.: Menge der Log-Einträge pro Student (s. Anhang)

Evaluierung

Wie selbst in der verkleinerten Darstellung der Grafik zu erkennen ist, gab es sehr auffällige Unterschiede im Umfang der studentischen Aktivitäten. Es war bereits an dieser Stelle zu vermuten, dass diejenigen Studenten mit nur geringen Mengen an Log-Einträgen auch nur wenig kontinuierlich gearbeitet haben und wahrscheinlich auch nur an wenigen Wochen und Tagen und überhaupt aktiv waren. Aufschluss hierüber sollten die beiden nachfolgenden Analysen geben.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst ein *viewed*-Event initiiert haben, also auch solche die mit einem *viewed*-Event einer anderen Person in Beziehung standen (s. das nachfolgende Listing).

```
1 # Ermittlung der Menge der Arbeitswochen pro Student
2 weeks_user = pd.Series(md.year_week[
3     (md.userstatus == 'student') & ((md['behaviour'] == 'learning') |
4     (md['behaviour'] == 'communication'))].groupby(md.userid).nunique(),
5     name='weeks')
```

Listing 30: Menge der Arbeitswochen pro Student

Die Grafik unten zeigt die Menge der Arbeitswochen im Gesamtzeitraum für jeden einzelnen Studenten.

Datenanalyse: Menge der Arbeitswochen pro Student

```
1 # Visualisierung der absoluten Menge der Arbeitswochen pro Student
2 chart = sns.barplot(x=weeks_user.index.astype(int),
3                      y=weeks_user, color=colors_general[3])
```

Listing 31: Menge der Arbeitswochen pro Student

Evaluierung

3. Analyse

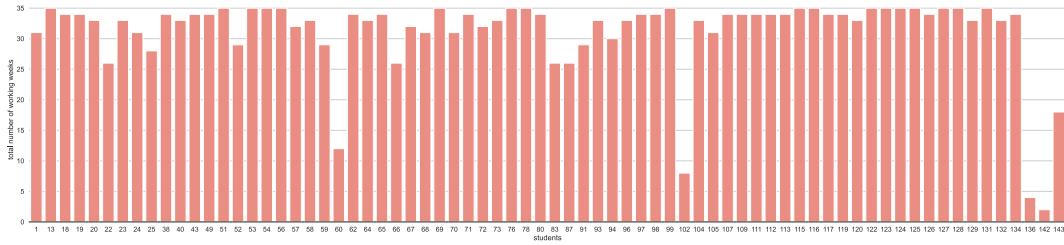


Abbildung 28.: Menge der Arbeitswochen pro Student ([s. Anhang](#))

Die obige Visualisierung bestätigte die vorherige Vermutung und zeigte bei den Studenten mit allgemein wenigen Log-Einträgen auch nur wenige aktive Wochen. Insofern war in diesen Fällen von einem sporadischen Lernverhalten auszugehen.

Fraglich waren an dieser Stelle jedoch die durchgängig hohen Wochenzahlen der anderen Studenten. Ließen diese unmittelbar auf eine ebenfalls hohe Kontinuität des Studierens schließen?

Unter Berücksichtigung dessen, dass die Untersuchung nur darauf abstellte, die Anzahl der Wochen zu ermitteln an denen ein Student überhaupt aktiv war, nicht aber zum Ziel hatte auch den Umfang der studentischen Aktivitäten innerhalb der Wochen aufzuzeigen, war diese Frage klar zu verneinen.

Hier war nun also doch geboten, neben den Wochen auch die nächstkleinere Zeit-einheit also die Tage an denen ein Student aktiv war, genauer zu untersuchen. Auch hierzu mussten zunächst die jeweiligen Zahlen pro Student aus den bereitgestellten Daten ermittelt werden.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst ein *viewed*-Event initiiert haben, also auch solche die mit einem *viewed*-Event einer anderen Person in Beziehung standen (s. das nachfolgende Listing).

```
1 # Ermittlung der Menge der Arbeitstage pro Student
2 days_user = pd.Series(md.year_day[
3     (md.userstatus == 'student') & ((md['behaviour'] == 'learning') |
4     (md['behaviour'] == 'communication'))].groupby(md.userid).nunique(),
5     name='days')
```

Listing 32: Menge der Arbeitstage pro Student

Die nachfolgende Visualisierung präsentiert in einer Gesamtübersicht die Menge

3. Analyse

der Arbeitstage im gesamten Zeitraum pro Student.

Datenanalyse: Menge der Arbeitstage pro Student

```
1 # Visualisierung der absoluten Menge der Arbeitstage pro Student
2 chart = sns.barplot(x=days_user.index.astype(int),
3                      y=days_user, color=colors_general[6])
```

Listing 33: Menge der Arbeitstage pro Student

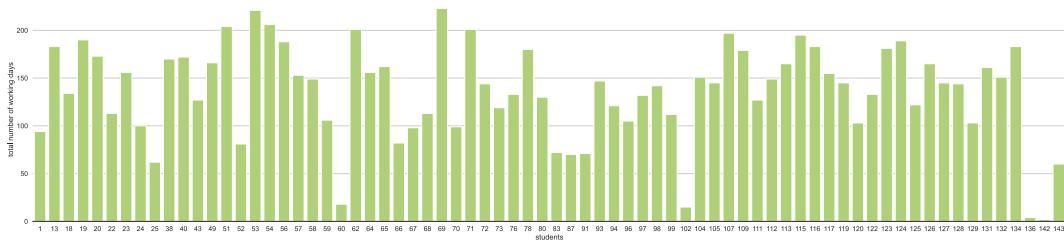


Abbildung 29.: Menge der Arbeitstage pro Student ([s. Anhang](#))

Evaluierung

Betrachtete man das obige Diagramm zu den Arbeitstagen im direkten Vergleich mit dem vorherigen zu den Mengen der Arbeitswochen, so wurde schnell deutlich, dass einige Studenten auf Tagessicht eine deutlich schlechtere Performance zeigten als auf Wochensicht.

Dies mochte sicher verschiedene Gründe haben, es ließ aber erkennen, *dass sich die studentischen Aktivitäten über die Tage und Wochen in unterschiedlichen zeitlichen Abständen, d.h. in mehr oder minder kontinuierlicher Form, verteilten*.

Damit wurde hier deutlich, dass eine rein wochenbasierte Sicht für die Untersuchung auf Kontinuität nicht ausreichte, sondern die Frage, wie diese zu bemessen sei, *sich nur in der gemeinsamen Betrachtung der Wochen- und Tagesaktivitäten beantworten ließ*.

3.4.2. Ermittlung der Vergleichsgöße

Waren die zeitlichen Maßstäbe nun definiert, ging es um die Frage: Woran sollte die Kontinuität denn bemessen werden? Dies bedeutete, es musste vorab noch geklärt werden, worauf sich die Untersuchung beziehen sollte. Gab es eine für Studenten allgemein gültige Größe, an der man die Kontinuität des individuellen Verhaltens

3. Analyse

objektiv messen konnte?

Nach Betrachtung von Einzelfällen und reiflicher Überlegung war diese Frage zu verneinen, denn man musste stets auch jene Studenten im Blick haben, die z. B. aufgrund einer erhöhten beruflichen Belastung nur sehr wenige Kursmodule belegt hatten, diese aber konsequent bearbeiteten. Hätte man vielleicht einfach auf Basis der Gesamtaktivitäten, die wie bereits gezeigt sehr unterschiedlich war, und des Gesamtzeitraums eine fixe Größe zur Unterscheidung ermittelt, wäre man diesen Fällen sicher nicht gerecht geworden.

Zur Beantwortung der Frage musste folglich stets die individuelle Situation eines Studenten betrachtet werden, nur daran konnte man das Maß für die Kontinuität der Aktivitäten messen.

Für das weitere Vorgehen war noch notwendig, ein paar weitere zeitliche Größen zu ermitteln. So wurde das bisherige Datenset u. a. um zusätzliche Informationen zur Kalenderwoche und zum Kalendertag ergänzt, um diese bei Bedarf unmittelbar identifizieren zu können.

Im Anschluss daran wurde auf Basis der schon ermittelten Ergebnisse zu den Log-Einträgen, den Arbeitswochen sowie den Arbeitstagen pro Student ein neues Datenset *time_relation* erstellt, dem nun in weiteren Schritten für jeden Studenten wichtige individuelle Kennziffern hinzugefügt werden sollte.¹⁹

Im wesentlichen ging es hierbei zunächst darum, die Gesamtzahl der Log-Einträge pro Student zu den jeweiligen Wochen und Tagen ins Verhältnis zu setzen, um so einen individuellen Durchschnitt zu erhalten. Dieser sollte jedoch nicht selbst als Größe in die Analyse einbezogen werden, sondern nur als Anhaltspunkt dienen, von dem aus für jeden Studenten ein eigenes unteres bzw. oberes Limit berechnet werden konnte.

Mithilfe des durch diese beiden Limits begrenzten Toleranzbereichs, der sich in seiner Ausdehnung auch an der individuellen Gesamtaktivität bemisst, sollte es also möglich sein, *auch natürlich auftretende moderate Schwankungen studentischer Aktivitäten angemessener zu berücksichtigen*, als dieses durch Verwendung eines

¹⁹ Siehe auch das dieser Arbeit beigelegte Jupyter Notebook Dokument zur zeitbezogenen Analyse.

einfachen Schwellwerts möglich gewesen wäre.

Datenaufbereitung

Bei der Auswahl der Daten berücksichtigt wurden sowohl die Studenten, die selbst ein *viewed*-Event initiiert haben, also auch solche die mit einem *viewed*-Event einer anderen Person in Beziehung standen.

```

1 # Erstellung des neuen Datensets
2 time_relation =
3     pd.concat([loggings_user, weeks_user, days_user], axis=1)
4
5 # Erstellung neuer Spalten für die individuellen Kennziffern pro Woche
6 time_relation['avg_count_per_week'] = 0
7 time_relation.loc[(time_relation['avg_count_per_week'] == 0),
8     ['avg_count_per_week']] =
9     (loggings_user / weeks_user).astype(int)
10 time_relation['lower_count_per_week'] = 0
11 time_relation.loc[(time_relation['lower_count_per_week'] == 0),
12     ['lower_count_per_week']] =
13     ((loggings_user / weeks_user) * 0.5).astype(int)
14 time_relation['upper_count_per_week'] = 0
15 time_relation.loc[(time_relation['upper_count_per_week'] == 0),
16     ['upper_count_per_week']] =
17     ((loggings_user / weeks_user) * 1.5).astype(int)
18
19 # Erstellung neuer Spalten für die individuellen Kennziffern pro Tag
20 time_relation['avg_count_per_day'] = 0
21 time_relation.loc[(time_relation['avg_count_per_day'] == 0),
22     ['avg_count_per_day']] =
23     (loggings_user / days_user).astype(int)
24 time_relation['lower_count_per_day'] = 0
25 time_relation.loc[(time_relation['lower_count_per_day'] == 0),
26     ['lower_count_per_day']] =
27     ((loggings_user / days_user) * 0.5).astype(int)
28 time_relation['upper_count_per_day'] = 0
29 time_relation.loc[(time_relation['upper_count_per_day'] == 0),
30     ['upper_count_per_day']] =
31     ((loggings_user / days_user) * 1.5).astype(int)

```

Listing 34: Erstellung des Datensets zur Untersuchung zeitlicher Beziehungen

Unten dargestellt ist das neue Datenset, bei dem aus Platzgründen die Spalten für die Log-Einträge, die Arbeitswochen und die Arbeitstage pro Student ebenso wie verschiedene Zeilen nur durch Punkte (...) angezeigt werden können.

	userid	...	avg_count	lower_count	upper_count	avg_count	lower_count	upper_count
	userid	...	per_week	per_week	per_week	per_day	per_day	per_day
0	1	...	42	21	64	14	7	21
1	13	...	78	39	118	15	7	22
2	18	...	33	16	49	8	4	12
3	19	...	88	44	132	15	7	23
4	20	...	113	56	169	21	10	32
5
67	132	...	47	23	70	10	5	15
68	134	...	90	45	135	16	8	25
69	136	...	4	2	6	4	2	6
70	142	...	1	0	2	1	0	2
71	143	...	29	14	44	8	4	13

Abbildung 30.: Datenset zur Untersuchung zeitlicher Beziehungen (s. Anhang)

Mit diesem neuen Datenset waren die Voraussetzungen geschaffen, um auch die Mengen der Wochen und Tage zu ermitteln, an denen ein Student innerhalb seines persönlichen Toleranzbereichs agiert hat. Hierzu wurden nun ebenfalls die Werte zu den Kalenderwochen und Kalendertagen benötigt, die eingangs dem ursprünglichen Datenbestand hinzugefügt wurden.

Datenaufbereitung

Die Datenaufbereitung erfolgte an dieser Stelle in zwei Phasen: Zunächst waren die persönlichen Toleranzwerte mit den Kalenderwochen und -tagen abzugleichen, um die entsprechenden Mengen zu ermitteln. Anschließend waren diese in Listen gespeicherten Mengen als neue Kennziffern dem neuen Datenset zu ergänzen.

```

1 # Ermittlung der Menge der Arbeitswochen, an denen die Menge der
2 # Log-Einträge pro Arbeitswoche innerhalb des Toleranzbereichs lag
3 list_weeks = list()
4
5 def get_weeks_in_range(i, list_weeks):
6     count = 0
7     for row in md.year_week[
8         (md.userid == time_relation.iloc[i]['userid']) &
9         ((md['behaviour'] == 'learning') |
10          (md['behaviour'] == 'communication'))]
11         .groupby(md.year_week).count():
12             if row > time_relation.iloc[i]['lower_count_per_week'] &
13                 row < time_relation.iloc[i]['upper_count_per_week']:
14                 count += 1
15     list_weeks.append(count)
16
17 for i in time_relation.index:
18     get_weeks_in_range(i, list_weeks)
19
20 weeks_in_range =
21     pd.Series(list_weeks, name='weeks_in_range', dtype='int32')
22
23 # Ergänzung des Datensets mit den Arbeitswochen im Toleranzbereich
24 time_relation = pd.concat([time_relation, weeks_in_range], axis=1)

```

Listing 35: Ermittlung der Arbeitswochen im Toleranzbereich

Die Ermittlung der Mengen der Arbeitstage, an denen die Menge der Log-Einträge pro Arbeitstag innerhalb des Toleranzbereichs lag, verlief analog:

```

1 # Ermittlung der Menge der Arbeitstage, an denen die Menge der
2 # Log-Einträge pro Arbeitstag innerhalb des Toleranzbereichs lag
3 list_days = list()
4
5 def get_days_in_range(i, list_days):
6     count = 0
7     for row in md.year_day[
8         (md.userid == time_relation.iloc[i]['userid']) &
9         ((md['behaviour'] == 'learning') |
10          (md['behaviour'] == 'communication'))]
10         .groupby(md.year_day).count():
11             if row > time_relation.iloc[i]['lower_count_per_day'] &
12                 row < time_relation.iloc[i]['upper_count_per_day']:
13                 count += 1
14     list_days.append(count)
15
16 for i in time_relation.index:
17     get_days_in_range(i, list_days)
18
19 days_in_range =
20     pd.Series(list_days, name='days_in_range', dtype='int32')
21
22 # Ergänzung des Datensets mit den Arbeitstagen im Toleranzbereich
23 time_relation = pd.concat([time_relation, days_in_range], axis=1)

```

3. Analyse

```

11     ].groupby(md.year_day).count():
12     if row > time_relation.iloc[i]['lower_count_per_day'] &
13         row < time_relation.iloc[i]['upper_count_per_day']:
14         count += 1
15     list_days.append(count)
16
17 for i in time_relation.index:
18     get_days_in_range(i, list_days)
19
20 days_in_range =
21     pd.Series(list_days, name='days_in_range', dtype='int32')
22
23 # Ergänzung des Datensets mit den Arbeitswochen im Toleranzbereich
24 time_relation = pd.concat([time_relation, days_in_range], axis=1)

```

Listing 36: Ermittlung der Arbeitstage im Toleranzbereich

Waren an diesem Punkt die Vorbereitungen gewissermaßen abgeschlossen, *konnte endlich der Versuch unternommen werden, die notwendigen Größen zu bestimmen, nach welchen die Kontinuität studentischer Aktivitäten zu messen war.*

Ein paar logische Vorüberlegungen waren aber dennoch noch zu leisten, bevor es an die konkrete mathematische Formulierung gehen konnte. Die gesuchten Größen mussten folgende Eigenschaften erfüllen:

1. Gemäß den bisherigen Ergebnissen sollten die ermittelten Mengen an Wochen und Tagen im individuellen Toleranzbereich berücksichtigt werden.
2. Als feste Bezugsgröße war der Zeitraum miteinzubeziehen, innerhalb dessen die Kontinuität festgestellt werden sollte.
3. War ein Student nie aktiv, musste der mathematische Ausdruck im Ergebnis den Wert 0 ergeben.
4. War ein Student an jedem Tag und somit dann auch in jeder Woche aktiv, sollte der mathematische Ausdruck den maximalen Wert liefern.

Entsprechend dieser Vorgaben konnte schließlich für das Verhältnis der genannten Mengen die nachfolgende Gesetzmäßigkeit definiert werden:

$$\begin{array}{ccc}
 \frac{\text{Menge der Arbeitswochen im Toleranzbereich} \longrightarrow 0}{\text{Gesamtmenge der Wochen im Untersuchungszeitraum}} & \times & \frac{\text{Menge der Arbeitstage im Toleranzbereich} \longrightarrow 0}{\text{Gesamtmenge der Tage im Untersuchungszeitraum}} \longrightarrow 0 \\
 \longrightarrow 0 & & \longrightarrow 0
 \end{array}$$

$$\begin{array}{ccc}
 \frac{\text{Menge der Arbeitswochen im Toleranzbereich} \longrightarrow \max}{\text{Gesamtmenge der Wochen im Untersuchungszeitraum}} & \times & \frac{\text{Menge der Arbeitstage im Toleranzbereich} \longrightarrow \max}{\text{Gesamtmenge der Tage im Untersuchungszeitraum}} \longrightarrow 1 \\
 \longrightarrow 1 & & \longrightarrow 1
 \end{array}$$

Abbildung 31.: Individueller Kontinuitätskoeffizient ([s. Anhang](#))

3. Analyse

Dargestellt in der obigen Abbildung sind die beiden maximalen Ausprägungen des individuellen Kontinuitätskoeffizienten (s. o. die beiden letztgenannten Überlegungen 3 und 4).

In allen anderen Situationen, repräsentiert durch beliebige Werte zwischen 0 und den maximalen Werten für die Arbeitswochen bzw. Arbeitstage im Toleranzbereich eines Studenten könnte das Ergebnis des Gesamtausdrucks dann immer nur in Richtung des Werts 0 oder in Richtung des Werts 1 tendieren, dieses Intervall aber nicht verlassen. *Mit dem individuellen Kontinuitätskoeffizienten (IKK) ist also die gesuchte Größe gefunden, mit der sich die Kontinuität studentischen Verhaltens messen lässt.*

Hier muss noch ein Absatz zur programmtechnischen Ermittlung und das Listing zur Ermittlung des IKK selbst ergänzt werden.

3.4.3. Kategorisierung nach IKK

Der IKK sollte nun ebenfalls noch als Kennziffer in das Datenset *time_relation* aufgenommen werden, *um für die abschließende Typisierung der Studenten nach ihren individuellen Werten* zur Verfügung zu stehen.

Datenaufbereitung

```
1 # Ergänzung des individuellen Kontinuitätskoeffizienten (IKK)
2 time_relation['ikk'] = 0
3 time_relation.loc[(time_relation['ikk'] == 0), ['ikk']] =
4     ((time_relation.weeks_in_range / time_relation.weeks.max()) *
5      (time_relation.days_in_range / time_relation.days.max()))
```

Listing 37: Ergänzung des IKK im Datenset *time_relation*

Danach konnte in einem letzten vorbereitenden Schritt das Datenset *time_relation* durch das Merkmal *continuity* und entsprechender Kategorien auf Basis des IKK komplettiert werden.

```
1 # Erstellung einer neuen Spalte zur Typisierung
2 time_relation['continuity'] = 'standard'
3
4 # Einordnung der Studenten nach der Kontinuität ihrer Aktivitäten
5 time_relation.loc[(time_relation['ikk'] > 0.6), ['continuity']] = 'high'
6 time_relation.loc[(time_relation['ikk'] < 0.3), ['continuity']] = 'low'
```

Listing 38: Typisierung der Studenten nach Kontinuität ihrer Aktivitäten

3. Analyse

Die folgende Analyse sollte schließlich Aufschluss geben über die einleitende Frage, wie sich die Gesamtmenge der identifizierten Studenten aufgrund einer gewissen Kontinuität der Lern- und Kommunikationsaktivitäten in Gruppen einteilen lassen.

Datenanalyse: Typisierung der Studenten nach IKK

```
1 # Visualisierung der Typisierung der Studenten nach IKK
2 chart = sns.barplot(x=time_relation.userid.astype(int),
3                      y=time_relation.ikk, hue=time_relation.continuity,
4                      hue_order=['high', 'standard', 'low'],
5                      dodge=False, palette='Set2')
```

Listing 39: Typisierung der Studenten nach IKK

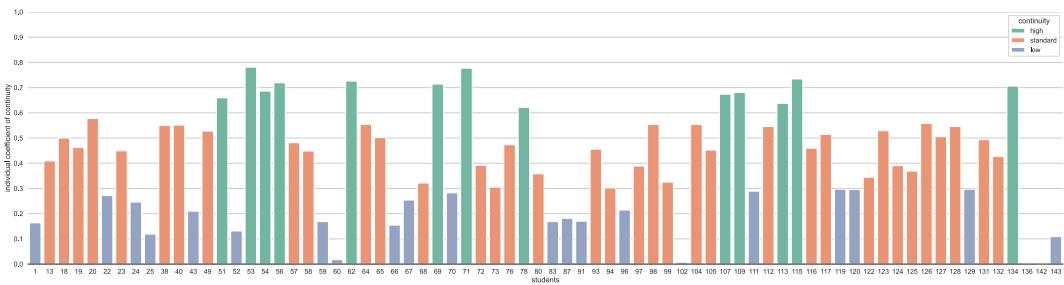


Abbildung 32.: Typisierung der Studenten nach IKK ([s. Anhang](#))

Evaluierung

Wie die Visualisierung des finalen Resultats der Analyse aufzeigt, *ließen sich die Studenten nach den bei der Datenaufbereitung gewählten IKK-Schwellwerten in die drei Gruppen High-, Low- und Standard-Performer einteilen.*

Im konkreten Fall sollte zwar eine gleichmäßige Aufteilung des IKK-Intervalls genügen, es wären jedoch je nach gewählter Höhe und Anzahl der Schwellwerte auch andere Ergebnisse möglich gewesen. *Dies weist darauf hin, dass mithilfe des IKK eine Typisierung des studentischen Verhaltens ferner sehr einfach und flexibel möglich ist.*

3.4.4. Prüfung des Untersuchungsergebnisses

An dieser Stelle durfte die Untersuchung aber noch nicht enden, *denn die Richtigkeit der Kategorisierung war mit dem obigen Ergebnis tatsächlich nicht bewiesen.* Festgestellt wurde hier, dass auf Grundlage vorab gewonnener Erkenntnisse eine Einordnung nach willkürlich definierten IKK-Schwellwerten erfolgen konnte. Dass

3. Analyse

also z. B. alle Studenten mit einem IKK größer als 0.6 als High-Performer galten, konnte also schlichtweg nur an dieser Festlegung liegen und daran, dass der IKK selbst als Maßstab der Betrachtung (s. y-Achse) diente.

Um nun also zu prüfen, ob die Einordnung der Studenten anhand der beliebigen Schwellwerte korrekt war, bedurfte es vielmehr eines Blicks auf eine unabhängige Größe. Dementsprechend empfahl es sich, hier erneut zum Anfang der Analysen zurückzukehren und zu betrachten, wie sich die Einordnung anhand der Schwellwerte bei Untersuchung der Log-Einträge über den Gesamtzeitraum darstellte.

Datenaufbereitung

Um den direkten Vergleich mit der vorherigen Analyse zu ermöglichen, erfolgte die Auswahl der Daten und die Definition der IKK-Schwellwerte vollkommen analog (s. [Listings zur Datenaufbereitung](#)).

Datenanalyse: Typisierung der Studenten nach IKK mit Bezug auf Log-Einträge

```
1 # Visualisierung der Typisierung der Studenten nach IKK
2 # mit Bezug auf die absolute Menge der Log-Einträge pro Student
3 chart = sns.barplot(x=time_relation.userid.astype(int),
4                      y=time_relation.loggings,
5                      hue=time_relation.continuity,
6                      hue_order=['high', 'standard', 'low'],
7                      dodge=False, palette='Set2')
```

Listing 40: Typisierung der Studenten nach IKK

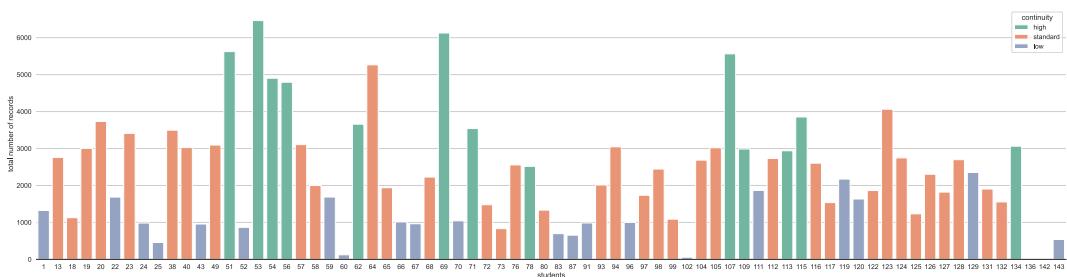


Abbildung 33.: Typisierung nach IKK mit Bezug auf Log-Einträge ([s. Anhang](#))

Evaluierung

Wie im Vergleich der beiden letzten Visualisierungen deutlich wurde, machte es einen bedeutenden Unterschied, ob die Kontinuität als solche betrachtet wurde oder ob diese nur als Zusatzinformation bei der Analyse einer realen Größe diente.

So konnte u.a. bei den Studenten 59 und 80 und mehr noch bei den Studenten 62

3. Analyse

und 64 beobachtet werden, dass eine geringere absolute Menge an Log-Einträgen durchaus mit einer höheren Kontinuität einhergehen kann bzw. umgekehrt. Wenn dem so war, dann musste sich dies auch im großen Bild bei der Visualisierung der Log-Einträge über den Gesamtzeitraum zeigen.

Datenaufbereitung

Um den direkten Vergleich zu ermöglichen, erfolgte die Auswahl der Daten und die Festlegung der IKK-Schwellwerte erneut analog zu den vorherigen Analysen ([s. Listings zur Datenaufbereitung](#)).

Datenanalyse: Menge der Log-Einträge im Gesamtzeitraum nach Tagen

```
1 # Visualisierung der Menge der Log-Einträge
2 # pro Student über 235 Tage (Gesamtzeitraum)
3 chart =
4     sns.histplot(x=md.timecreated,
5                 y=md.userid[md.userstatus == 'student'].astype(str),
6                 bins=235, hue=md.userid,
7                 hue_order=np.sort(md.userid.unique()),
8                 palette='rocket', edgecolor='white', legend=False)
```

Listing 41: Menge der Log-Einträge im Gesamtzeitraum nach Tagen

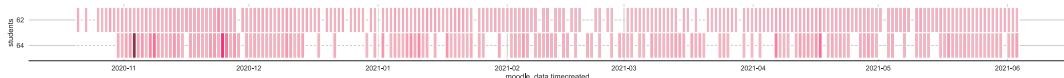


Abbildung 34.: Menge der Log-Einträge für Student 62 und 64 ([s. Anhang](#))

Evaluierung

Der obige schmale Ausschnitt aus der [Gesamtübersicht](#) zu den Verteilungen der Log-Einträge bestätigte das vermutete Ergebnis in vollem Umfang.

Anhand der beiden mittleren Zeilen war zu erkennen, dass Student 62 an mehr Wochen und Tagen aktiv war als Student 64, dennoch hatte er insgesamt deutlich weniger Log-Einträge zu verzeichnen als Student 64 ([s. Datenset zur Untersuchung zeitlicher Beziehungen](#)). Dies musste notwendigerweise bedeuten, dass die Log-Einträge von Student 62 über den Untersuchungszeitraum hinweg kontinuierlicher verteilt waren als bei seinem Komilitonen.

Letzterer war offenbar etwas später ins Semester gestartet, wie man der Lücke am Anfang seiner Zeile entnehmen konnte und wies im Semesterverlauf außerdem mehrere Tage mit deutlich höheren Zahlen auf als Student 62, was man an der

3. Analyse

stärkeren Färbung der Log-Einträge sah. Auch diese beiden letzten Beobachtungen sprachen eindeutig für ein im Vergleich mit Student 62 weniger kontinuierliches Arbeitsverhalten.

Mit Blick auf das Ergebnis der obigen Evaluierung war mithin die Typisierung auf Basis des ermittelten individuellen Kontinuitätskoeffizienten (IKK) abschließend geprüft, und die Korrektheit der Untersuchungsergebnisse konnte bestätigt werden.

An dieser Stelle folgen noch Analysen zu möglichen Unterschieden von Lernverhalten und Kommunikationsverhalten sowie Studiengängen.

3.4.5. Vergleich des Lern- und Kommunikationsverhaltens

...

3.4.6. Vergleich der Studiengänge

...

3.5. Dynamik des Lern- und Kommunikationsverhaltens

Die letzte der genannten *Grundfragen zur Analyse des studentischen Verhaltens* sollte im Folgenden den Kern dieses abschließenden Kapitels bilden: *Es war demnach zu untersuchen, inwiefern man Studenten nach der Dynamik ihres Verhaltens bewerten und kategorisieren konnte.*

Das große Bild mit dem das letzte Kapitel geendet hat, soll zum Einstieg kurz betrachtet werden. Hinweise auf erhöhte Mengen an Log-Einträgen insb. kurz vor den Prüfungszeiträumen können als erste Anhaltspunkte für eine erhöhte Dynamik dienen – manche Studenten habe hier auch weniger Loggings, was auf mangelnde Aktivität schließen lässt. Weitere einleitende Vermutungen könnten hier ebenfalls formuliert werden.

Ferner sollte man hier auch noch die Frage ansprechen, wie die Dynamik insgesamt zu betrachten sein sollte. Genauso wie bei der Kontinuität, also pauschal über den gesamten Zeitraum? Oder doch eher im Hinblick auf besondere Zeiträu-

3. Analyse

me und kritische Zeitpunkte wie z.B. Prüfungen? Bei der letzteren Betrachtungsweise, die wohl mehr Aussagekraft über das dynamische Verhalten hätte, würden die Lücken um die Weihnachtszeit oder zwischen den Prüfungen das Ergebnis aber wahrscheinlich stören.

Sollte man also vielleicht die Wochen um Weihnachten herum aus der Betrachtung herausnehmen? Auch wenn es letztlich nicht darum gehen sollte, die Dynamik per se absolut festzustellen, sondern nur eine Einordnung der Studenten vorzunehmen, es also letztlich nur um eine vergleichende Betrachtung gehen sollte i.S. von wer war wenig oder sehr dynamisch, wäre das die einfachste Lösung.

Könnte man das aber auch mit den Wochen zwischen den Prüfungszeiträumen so handhaben? Soll man die auch aus der Analyse herausnehmen? Diese Frage ist schwieriger zu beantworten, da man ja gerade auch betrachten will, wie sich das Verhalten im Hinblick auf die Prüfungen entwickelt ... und es gibt ja immer auch Studenten, die sich bewusst gegen einen Termin im Prüfungszeitraum 1 entscheiden und lieber im März zur Prüfung gehen. Bei diesen Studenten dürfte man dann nicht das Verhalten vor PZ1 betrachten, sondern müsste die Wochen vor PZ2 ansehen. Aber wie soll das gehen? Es gibt keine Infos, wer wann zur Prüfung gehen wollte oder tatsächlich gegangen ist.

Lösung: Man betrachtet nur die Wochen bis zum Prüfungszeitraum 1 (ohne die Woche vor und nach Weihnachten), da bis dahin in jedem Fall eine erhöhte Aktivität zur Erbringung der Prüfungsvorleistungen zu messen sein müsste – auch von denjenigen, die erst im Prüfungszeitraum 2 antreten.)

Nächster Schritt könnte sein, hier das ursprüngliche Datenset um ein Merkmal für die zu betrachenden Tage und Wochen zu ergänzen (kann aber auch später kommen direkt bei der konkreten Datenaufbereitung).

Wichtige Frage: Wie soll die Dynamik betrachtet werden? Gibt es einen objektiven Maßstab oder doch individuell wie bei der Kontinuität? Diese Frage müsste ähnlich wie bei der Kontinuität zu beantworten sein. Das Verhalten eines Studenten mit einem grundsätzlich geringeren Semesterworkload zeigt vermutlich auch eine geringere Dynamik vor den Prüfungswochen und würde bei einem allgemein-

3. Analyse

gültigen Maßstab eventuell falsch eingeordnet. Die Lösung ist also auch hier eine individuelle Betrachtung.

Im weiteren geht es an die praktische Umsetzung, sprich erste Analysen. Folgende Überlegungen sind dazu wichtig und bilden die Basis für die Ermittlung der individuellen Dynamik:

- Bei der Betrachtung der Dynamik in diesem Untersuchungskontext soll es um die Feststellung von Änderungen des Aktivitätsumfangs über einen bestimmten Zeitraum gehen. Im konkreten Fall bemisst sich der Aktivitätsumfang an den individuellen Mengen der Log-Einträge.
- Um eine Änderung oder Abweichung feststellen zu können, benötigt man eine feste Bezugsgröße. Hierbei bietet sich der Gesamtdurchschnitt an, d.h. das Verhältnis der individuellen Gesamtmenge an Log-Einträgen pro Gesamtzeitraum. Der resultierende Wert kann als konstanter Wert über die gesamte Zeit gelten wie über jeden beliebigen Teilabschnitt. Im konkreten Fall soll eine Tagessbetrachtung erfolgen, da diese präziser erscheint.
- Eine positive Dynamik resultiert aus einem Wert über dem Durchschnitt.
- Eine negative Dynamik resultiert aus einem Wert unter dem Durchschnitt.
- Alle Abweichungen vom arithmetischen Mittel der Log-Einträge pro Tag, gewichtet mit einem bestimmten Faktor, der zum Ende des Betrachtungszeitraums hin ansteigen sollte (um die Aktivität in den kritischen Wochen vor den Prüfungen stärker zu berücksichtigen), ergeben in der Summe schließlich den gesuchten Wert für die individuelle Dynamik.
- Eine linear ansteigende Gewichtung wie im konkreten Fall ist dabei nur eine Möglichkeit, je nach Anforderungen könnte z.B. auch eine quadratische oder eine andere Gewichtung in Frage kommen um evtl. die Prüfungsvorbereitungszeit noch stärker zu berücksichtigen.

Hier folgt die Formulierung des individuellen Dynamikkoeffizienten (IDK):

An die Darstellung des mathematischen Ausdrucks schließen sich noch ein paar ergänzende Erläuterungen an.

Im Anschluss erfolgt die Ermittlung des IDK und danach die Aufnahme des IDK

3. Analyse

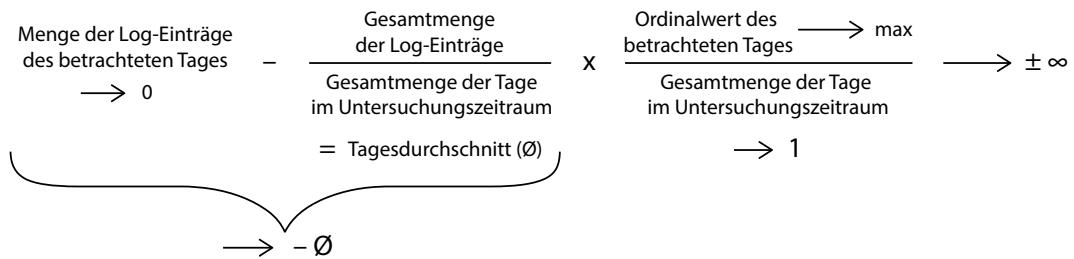


Abbildung 35.: Individueller Dynamikkoeffizient ([s. Anhang](#))

in ein neues Datenset (jeweils mit Code-Listings).

Anhand des ermittelten Werts für die Dynamik kann anschließend die Typisierung vorgenommen werden (Vorgehen analog zur Kontinuitätsanalyse).

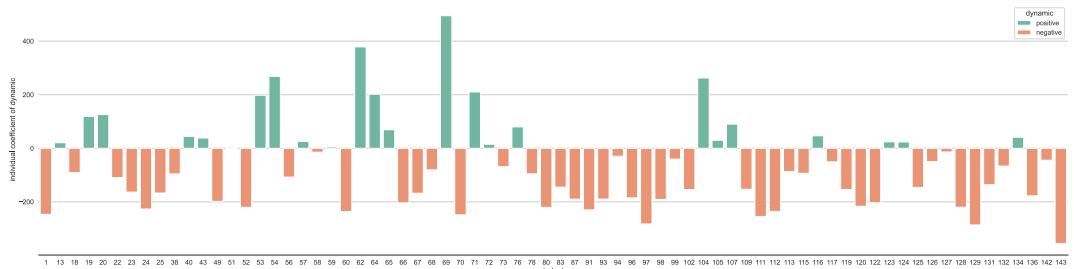


Abbildung 36.: Typisierung der Studenten nach IDK ([s. Anhang](#))

Vergleichende Analysen und/oder Einzelbetrachtungen sollen zur Verifikation ebenfalls noch durchgeführt werden.

An dieser Stelle folgen noch Analysen zu möglichen Unterschieden von Lernverhalten und Kommunikationsverhalten sowie Studiengängen.

3.5.1. Vergleich des Lern- und Kommunikationsverhaltens

...

3.5.2. Vergleich der Studiengänge

...

4. Ergebnisse

4. Ergebnisse

...

5. Fazit

...

6. Ausblick

6. Ausblick

...

Literaturverzeichnis

- Azevedo, A. & Santos, M. (2008, 01). KDD, SEMMA and CRISP-DM: A parallel overview. In (S. 182-185).
- Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27 (3), 326–327.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996, Mar.). From data mining to knowledge discovery in databases. *AI Magazine*, 17 (3), 37. Zugriff auf <https://ojs.aaai.org/index.php/aimagazine/article/view/1230> doi: 10.1609/aimag.v17i3.1230
- Green, M. (2022). *The Moodle Database. Table and relationship documentation generated from moodle source code*. Zugriff am 2022-04-08 auf <https://www.examulator.com/er/>
- Moodle. (2022). *The Moodle Documentation. Version 4.0*. Zugriff am 2022-06-30 auf https://docs.moodle.org/400/en/Main_page
- Runkler, T. A. (2020). Introduction. In *Data analytics: Models and algorithms for intelligent data analysis* (S. 1–4). Wiesbaden: Springer Fachmedien. Zugriff auf https://doi.org/10.1007/978-3-658-29779-4_1 doi: 10.1007/978-3-658-29779-4_1
- Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5 (4).
- Wang, R. Y. & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *J. Manag. Inf. Syst.*, 12, 5-33.

A. Anhang

A.2. Grundlagen

Datenbeschreibung / Visualisierung der Daten

Die folgenden Listings zeigen u. a. die erforderlichen Anweisungen zur Einrichtung der Arbeitsumgebung oder dem Import der Arbeitsdaten. Bei den Untersuchungen in dieser Arbeit wurden diese stets vorausgesetzt bzw. in besonderen Fällen entsprechend angepasst.

Prolog

```
1 from sqlalchemy import create_engine
2 import numpy as np
3 import pandas as pd
4 from matplotlib import pyplot as plt
5 import seaborn as sns
6 from IPython.core.display_functions import display
```

Listing 42: Import von Bibliotheken und anderen Erweiterungen

```
1 sns.set_theme(style='white', font_scale=1.2, palette='Spectral')
```

Listing 43: Definitionen zur Darstellung der Visualisierungen

```
1 user = "***"
2 password = "*****"
3 host = "localhost"
4 database = "vfh_moodle_ws20"
5 port = 3306
6
7 engine = create_engine(f'mysql+pymysql://{{user}}:{password}@{{host}}/{{database}}',
8                         pool_recycle=port)
9
10 connection = engine.connect()
```

Listing 44: Herstellung der Verbindung zur MySQL-Datenbank

```
1 query = """SELECT * FROM moodle_data"""
2 # Definition der Arbeitsdaten
3 moodle_data = pd.read_sql(query, connection)
```

Listing 45: Import der Arbeitsdaten aus der MySQL-Datenbank

A. Anhang

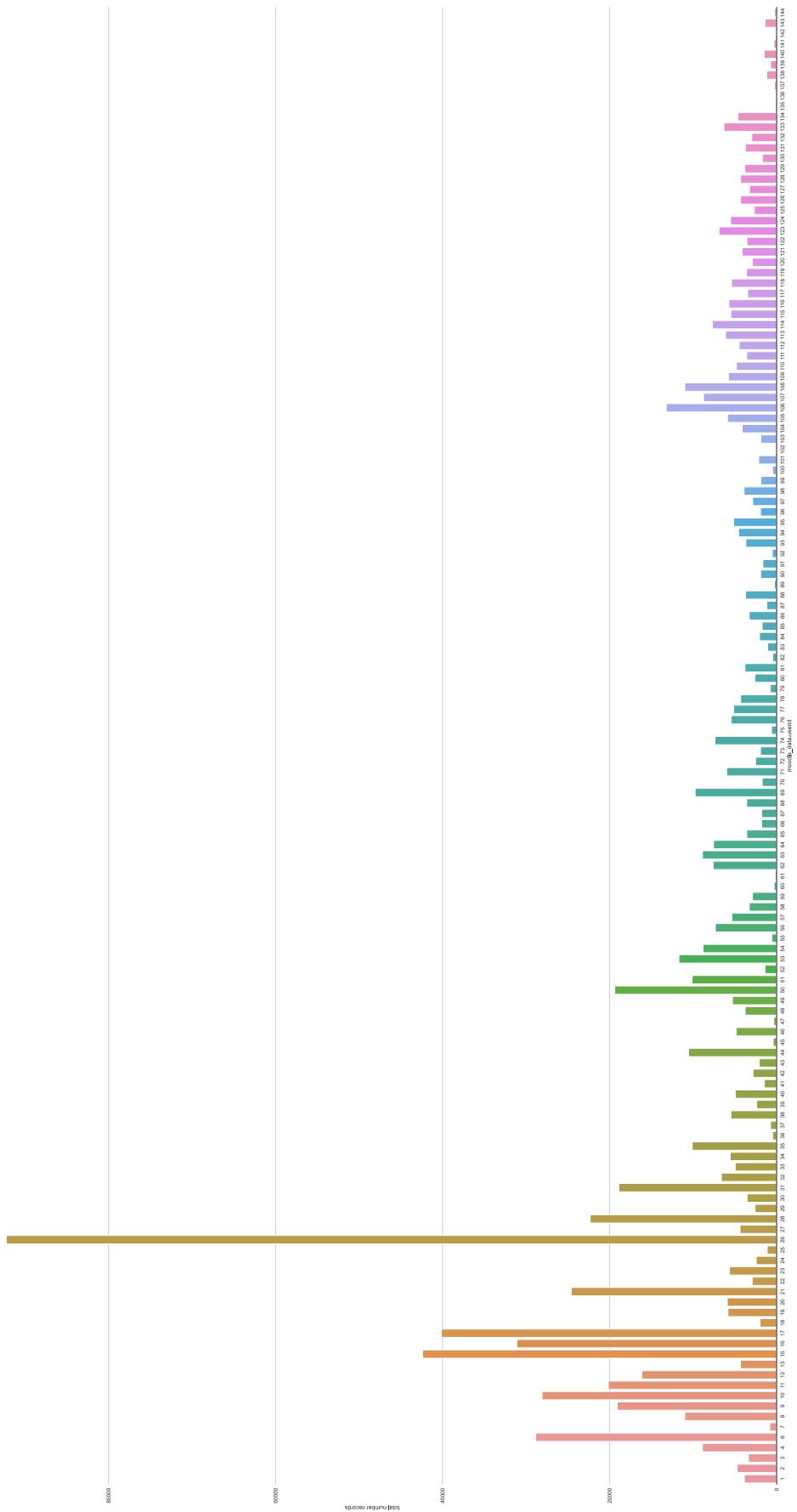


Abbildung 37.: Menge der Log-Einträge pro Benutzer

A. Anhang

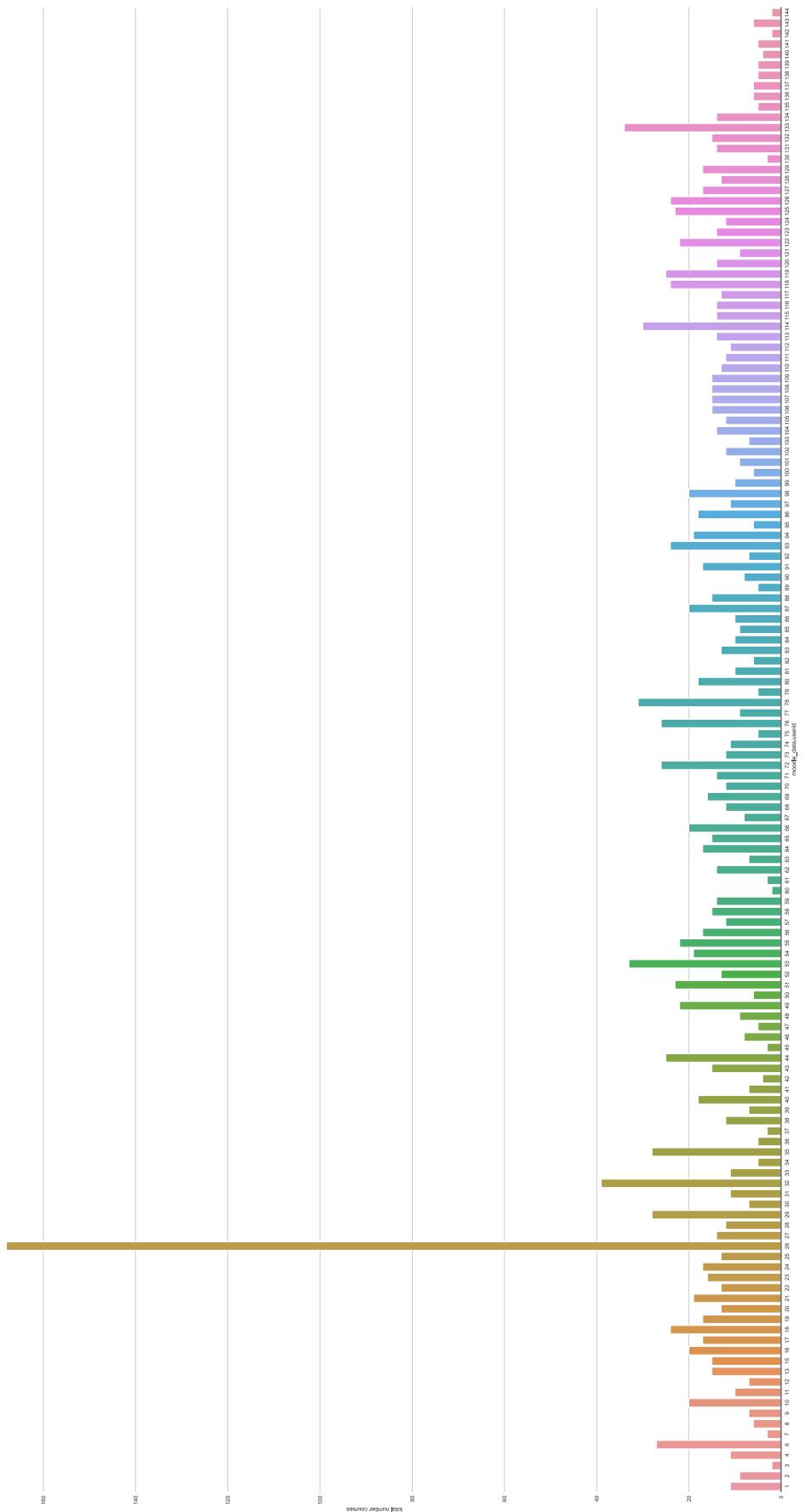


Abbildung 38.: Menge der Kurse pro Benutzer

A. Anhang

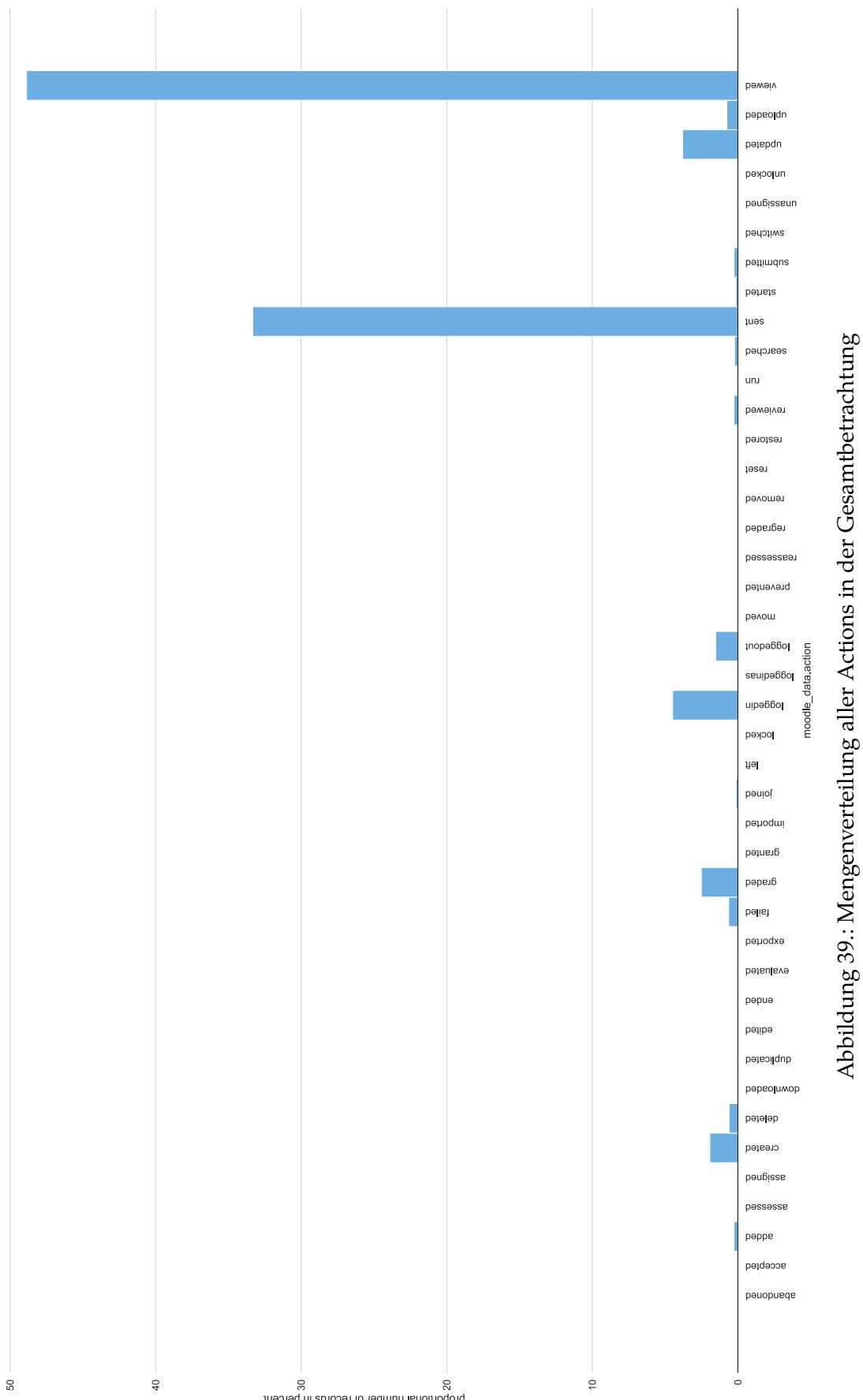


Abbildung 39: Mengenverteilung aller Actions in der Gesamtbeobachtung

A. Anhang

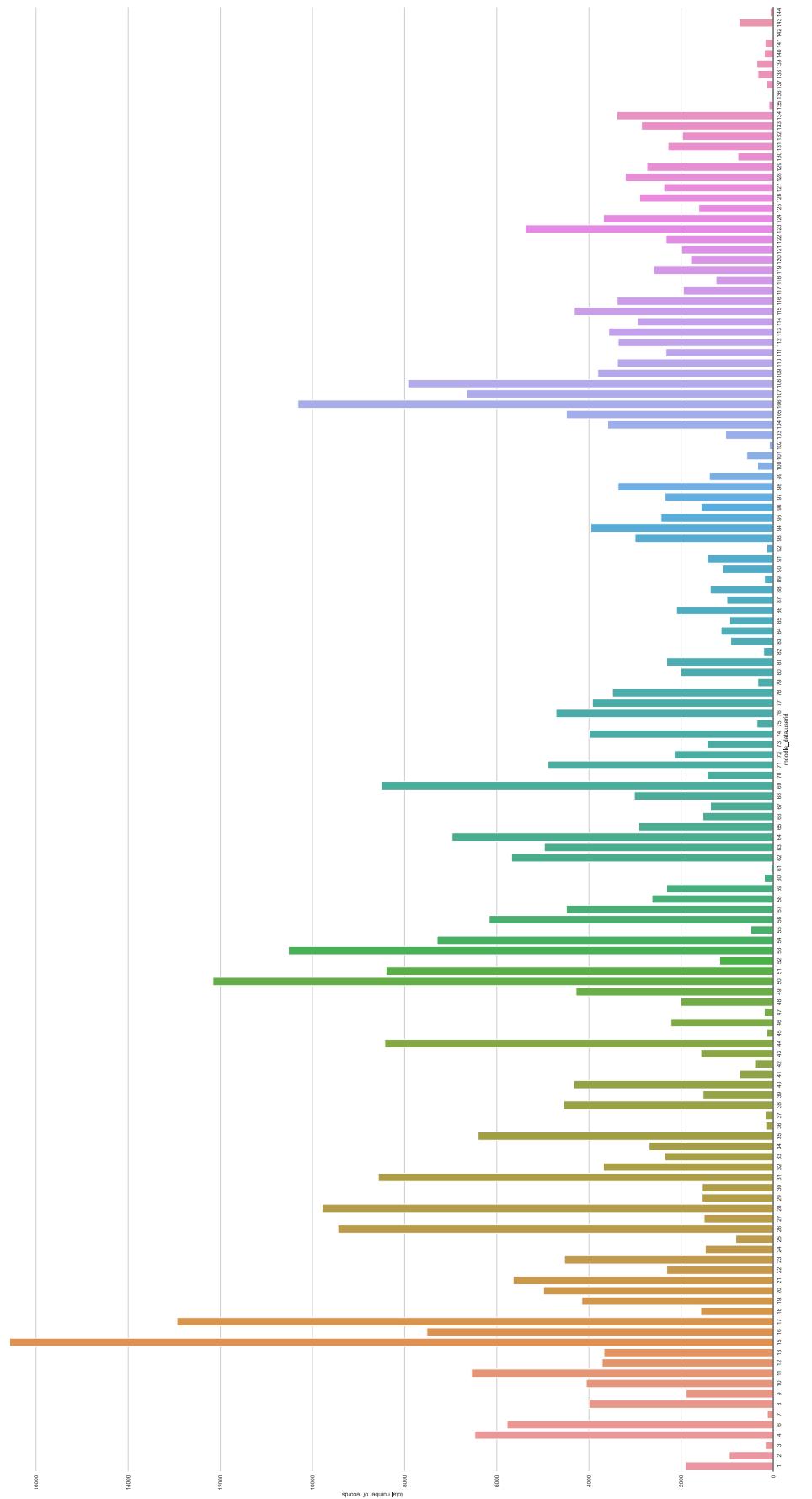


Abbildung 40.: Menge der viewed-Actions pro Benutzer

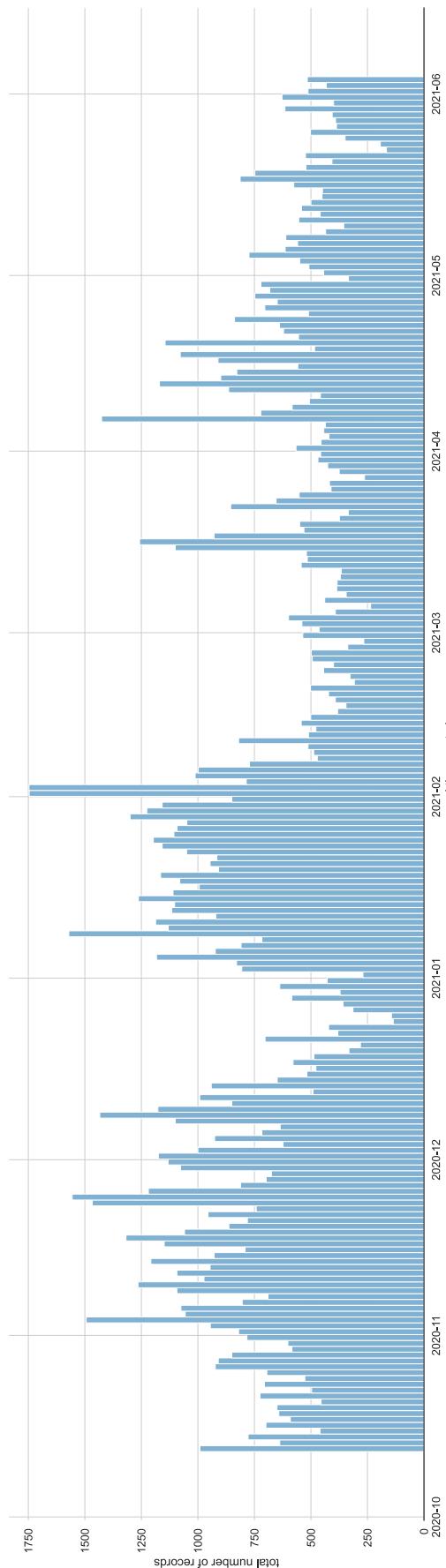


Abbildung 41.: Verteilung der Log-Einträge im Gesamtzeitraum pro Tag

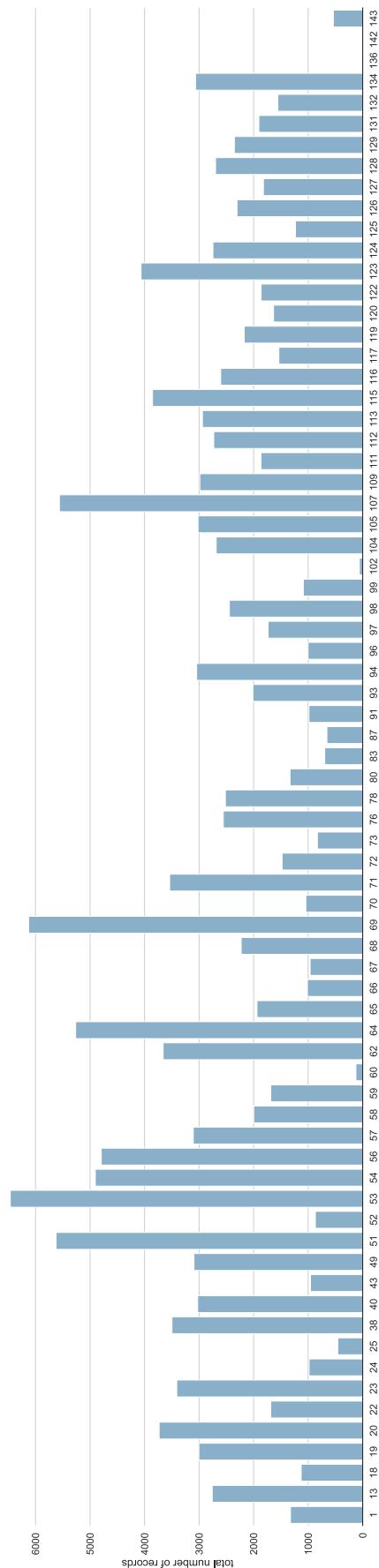


Abbildung 42: Menge der Log-Einträge im Gesamtzeitraum pro Student

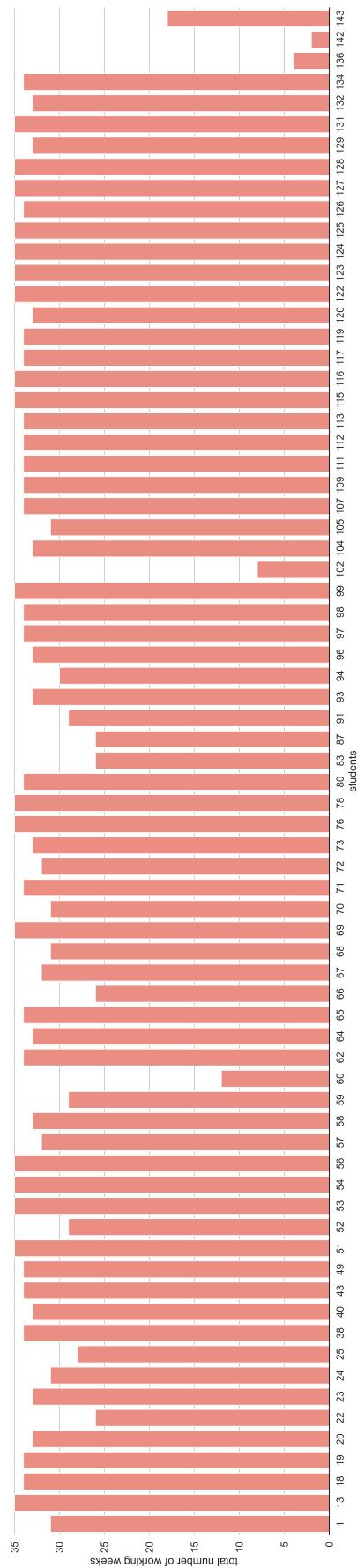


Abbildung 43: Menge der Arbeitswochen im Gesamtzeitraum pro Student

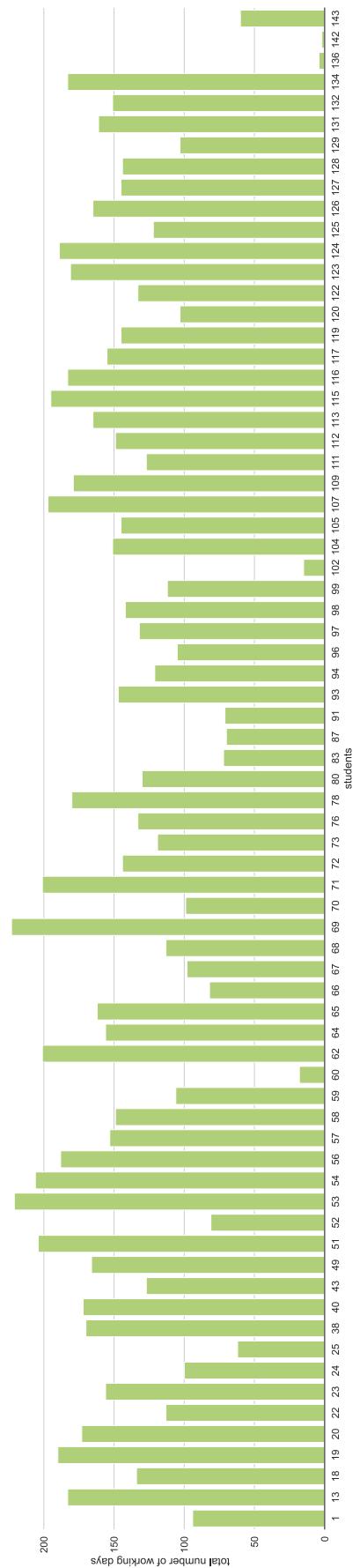


Abbildung 44: Menge der Arbeitstage im Gesamtzeitraum pro Student

A. Anhang

userid	loggings	days	weeks	avg_count_per_week	lower_count_per_week	upper_count_per_week	avg_count_per_day	lower_count_per_day	upper_count_per_day	avg_count_per_day	lower_count_per_day	upper_count_per_day
0	1	1324	31	94	42	21	64	14	15	7	7	21
1	2	2759	35	183	39	118	118	49	49	4	4	22
2	3	1128	34	134	33	16	16	44	44	12	12	22
3	4	3004	34	190	88	44	44	132	132	23	23	23
4	5	3733	33	173	113	56	56	169	169	10	10	32
5	6	1686	26	113	64	32	32	97	97	7	7	22
6	7	3410	33	156	51	103	103	155	155	10	10	32
7	8	983	31	100	31	15	15	47	47	4	4	14
8	9	457	25	457	28	8	8	24	24	3	3	11
9	10	3497	34	170	102	51	51	154	154	10	10	30
10	11	3026	33	172	91	45	45	137	137	8	8	26
11	12	957	34	127	28	14	14	42	42	3	3	11
12	13	1686	32	153	97	45	45	136	136	9	9	27
13	14	3094	34	166	91	45	45	145	145	10	10	41
14	15	5623	33	149	60	30	30	90	90	13	13	16
15	16	6462	35	204	160	80	80	240	240	10	10	43
16	17	4903	35	221	184	92	92	276	276	11	11	35
17	18	4794	35	188	130	70	70	210	210	12	12	38
18	19	3106	35	127	88	68	68	205	205	10	10	30
19	20	1998	33	149	97	48	48	145	145	6	6	20
20	21	1687	33	149	60	30	30	90	90	13	13	23
21	22	124	12	18	106	58	58	87	87	15	15	10
22	23	3659	34	201	107	53	53	161	161	6	6	27
23	24	5264	33	156	156	79	79	239	239	16	16	50
24	25	1938	34	162	57	28	28	85	85	11	11	18
25	26	1014	32	82	39	19	19	58	58	12	12	14
26	27	964	32	98	30	15	15	45	45	9	9	14
27	28	2228	31	223	71	35	35	107	107	19	19	29
28	29	6126	35	223	175	87	87	262	262	27	27	41
29	30	1043	31	99	33	16	16	50	50	10	10	15
30	31	3540	34	201	104	52	52	156	156	17	17	26
31	32	1479	32	144	46	23	23	69	69	10	10	15
32	33	8311	33	119	119	32	32	37	37	6	6	10
33	34	25556	35	133	73	36	36	109	109	19	19	28
34	35	2518	35	133	73	36	36	107	107	13	13	20
35	36	1333	34	130	39	19	19	58	58	10	10	15
36	37	696	26	72	26	16	16	37	37	9	9	14
37	38	656	26	70	26	12	12	37	37	6	6	20
38	39	985	29	71	33	16	16	50	50	13	13	20
39	40	2013	33	147	61	30	30	91	91	25	25	37
40	41	3045	30	121	101	50	50	152	152	13	13	31
41	42	1001	33	121	101	50	50	45	45	4	4	14
42	43	1735	34	132	51	25	25	76	76	13	13	19
43	44	2443	34	142	71	35	35	107	107	17	17	25
44	45	98	34	112	31	15	15	46	46	9	9	14
45	46	1089	35	112	31	15	15	33	33	11	11	6
46	47	102	60	88	35	15	15	7	7	4	4	6
47	48	2686	33	151	81	40	40	122	122	17	17	26
48	49	105	3020	31	145	97	48	146	146	20	20	31
49	50	5562	34	179	87	163	163	245	245	28	28	42
50	51	109	2987	34	179	87	43	131	131	16	16	25
51	52	49	109	2866	34	127	54	76	76	13	13	22
52	53	42	97	2445	34	149	80	27	27	14	14	22
53	54	98	43	1089	34	112	35	82	82	18	18	27
54	55	102	60	88	35	15	15	120	120	17	17	26
55	56	116	2686	33	151	81	40	129	129	17	17	29
56	57	119	1735	34	145	63	31	67	67	14	14	21
57	58	112	2931	34	149	80	40	95	95	14	14	21
58	59	113	2939	34	165	86	43	74	74	15	15	21
59	60	123	4064	34	165	86	43	79	79	14	14	21
60	61	124	2746	33	183	78	39	174	174	22	22	21
61	62	125	1246	33	122	35	17	117	117	10	10	15
62	63	126	2305	34	165	67	33	52	52	13	13	20
63	64	127	1821	35	145	52	26	78	78	12	12	18
64	65	128	2698	34	145	52	26	77	77	15	15	28
65	66	129	2353	33	133	53	38	115	115	22	22	34
66	67	131	4064	35	103	71	35	106	106	11	11	34
67	68	132	2746	33	161	54	27	81	81	11	11	34
68	69	134	1555	33	155	47	23	70	70	10	10	34
69	70	136	3063	34	183	90	45	135	135	8	8	25
70	71	142	33	2	2	0	0	2	2	0	0	2
71	72	143	538	18	60	29	14	44	44	1	1	13

Abbildung 45.: Datenset zur Untersuchung zeitlicher Beziehungen



Abbildung 46.: Individueller Kontinuitätskoeffizient (IKK)

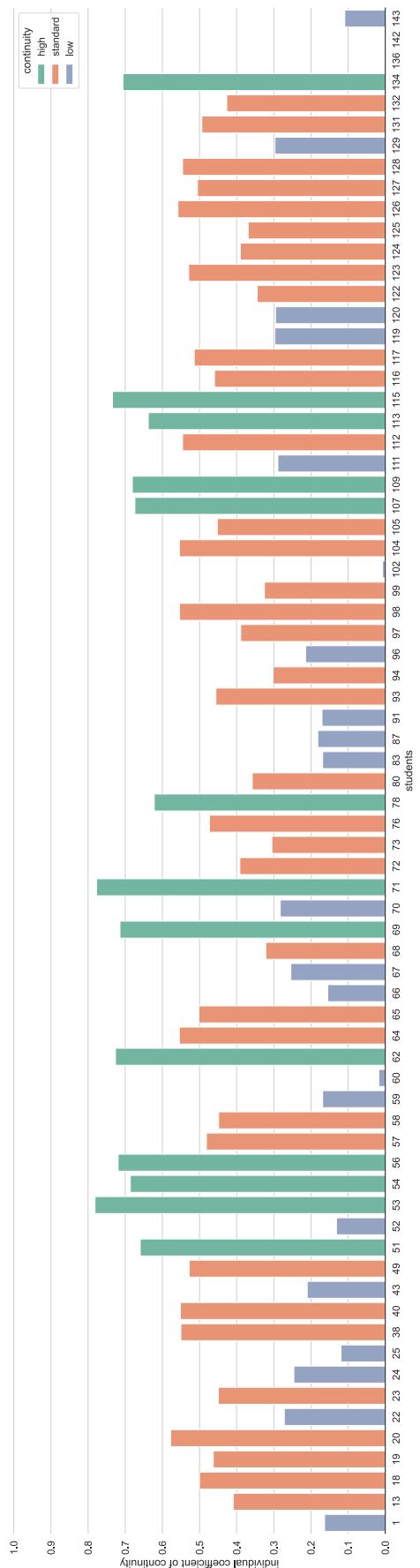


Abbildung 47.: Typisierung der Studenten nach IKK mit Bezug auf IKK

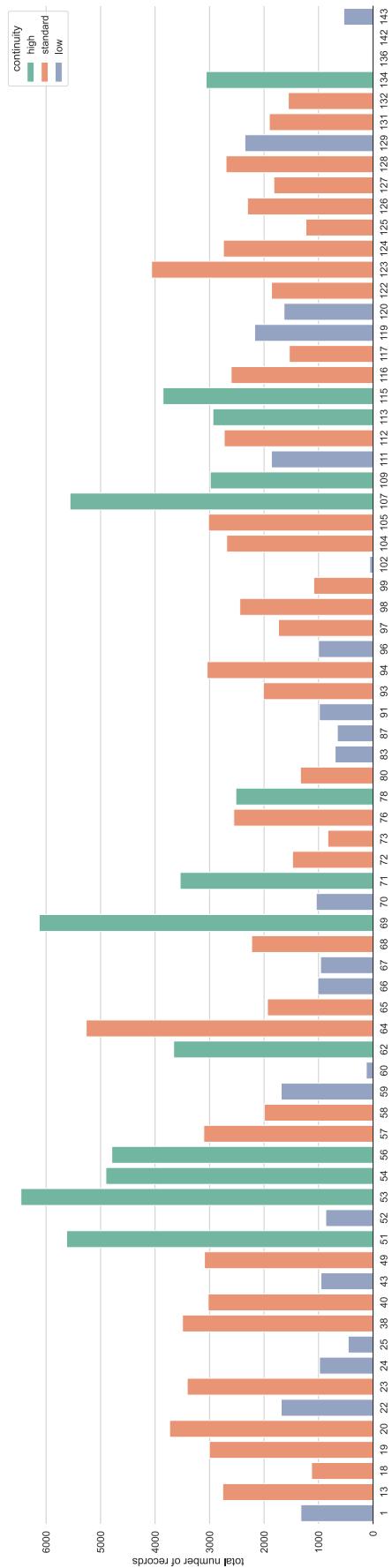
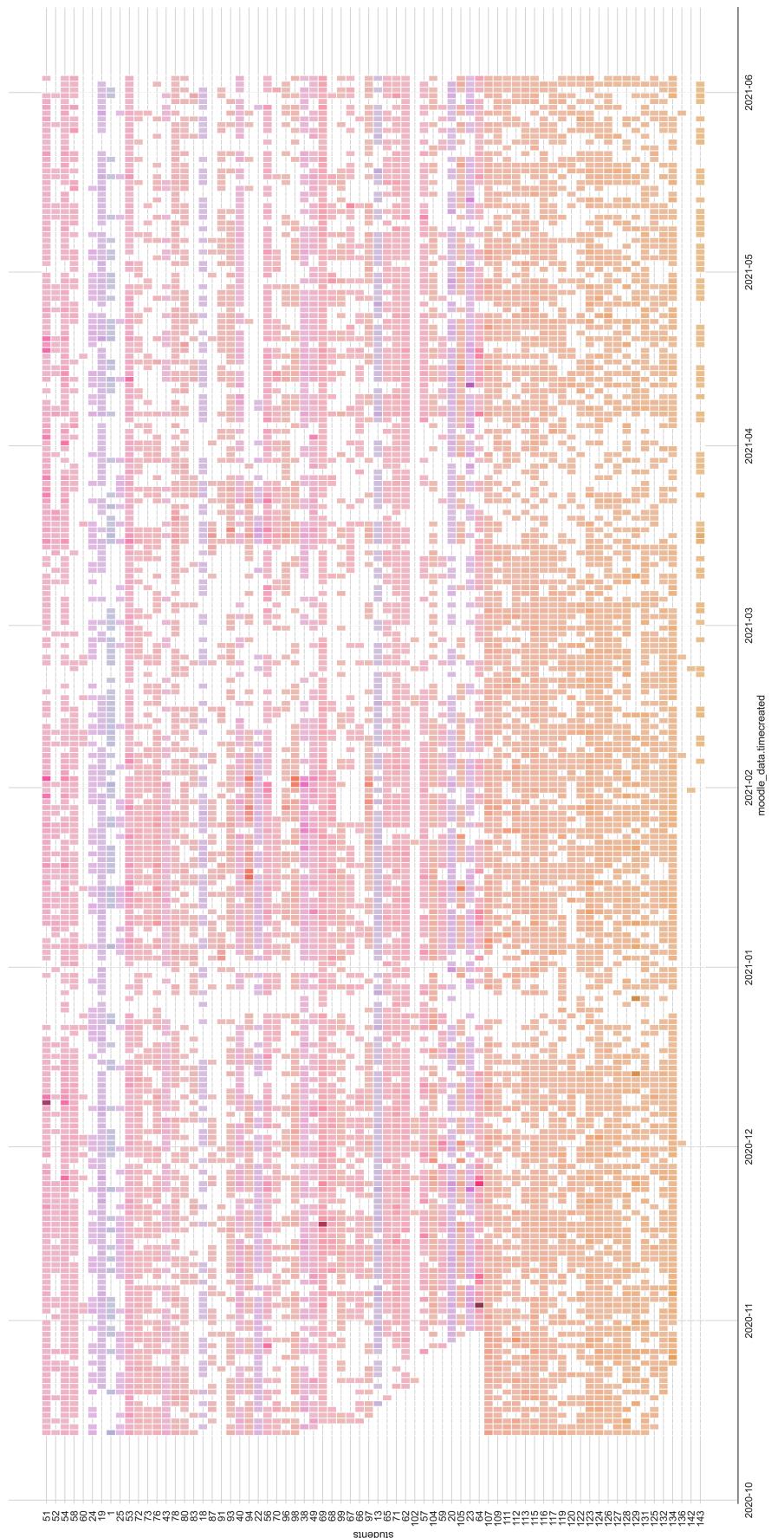


Abbildung 48.: Typisierung der Studenten nach IKK mit Bezug auf Log-Einträge

A. Anhang



89

Abbildung 49.: Menge der Log-Einträge pro Student im Gesamtzeitraum nach Tagen

Erklärung zur Urheberschaft

Erklärung zur Urheberschaft

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle Zitate und Übernahmen von fremden Aussagen kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version dieser Arbeit sind vollkommen identisch.

Heidelberg, dd.mm.2022

Unterschrift

Inhalt des beigefügten Datenträgers

Inhalt des beigefügten Datenträgers

Verzeichnis / Beschreibung

/1_ ...

/2_ ...

/3_ ...
