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

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

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 Five[1] 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

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 Informationen[2] 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.

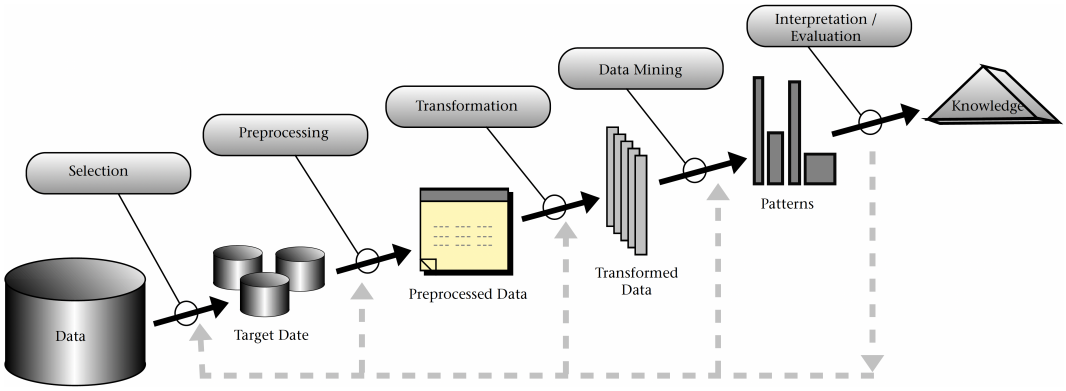

Abbildung 1.: Phasen des KDD-Prozesses. Original von Fayyad et al. (1996).

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-proprietäres Standard-Prozessmodell zu etablieren, das konkret als Blaupause dienen kann, um Datenbestände z. B. nach interessanten Mustern und Trends zu durchsuchen (Shearer, 2000).

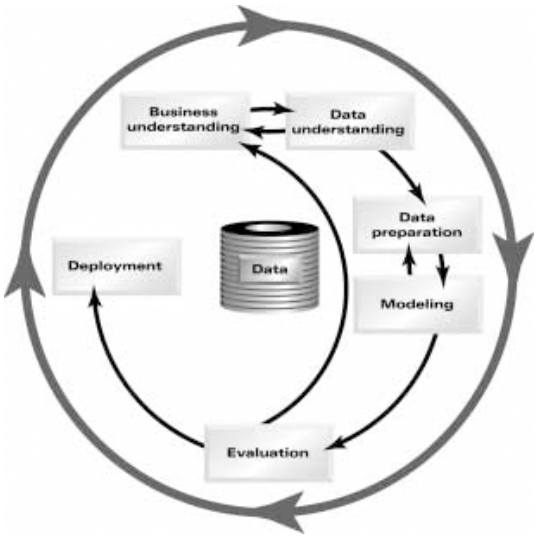

Abbildung 2.: Phasen des CRISP-DM. Original von Shearer (2000).

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

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 SEMMA[5] 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

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 allesamt 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 nachfoldende 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.

- 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 Darstellungen[7] 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.

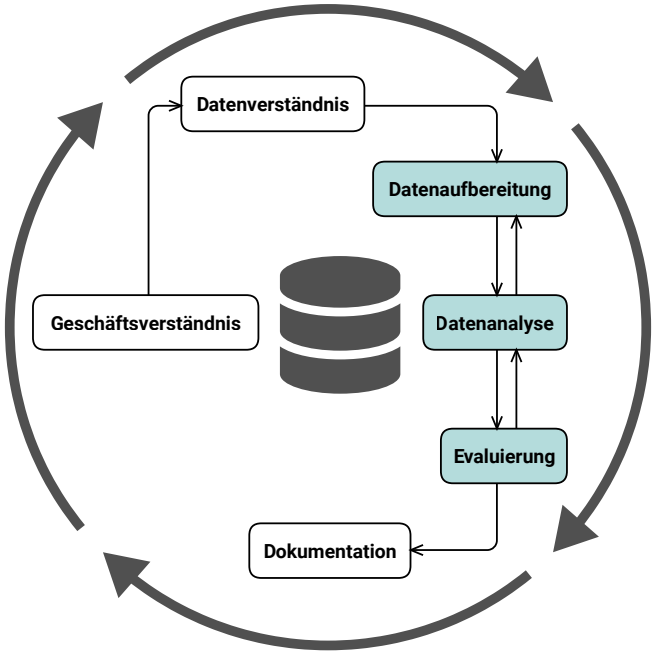

Abbildung 4.: Phasen des verwendeten Vorgehensmodells.

[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 2020/2021[9]. 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

(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 Studiengängen[10] 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

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

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 sogenannen 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 Kontext-betrachtungen zu berücksichtigen.

```
+--------+----------------------+
| 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

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 gegeben 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 unterteit 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.

```
# Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(64, 36)) # Größe der Visualisierung (in inch)

# Visualisierung der Menge der Log-Einträge pro Benutzer
chart = sns.countplot(x=moodle_data.userid)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_axisbelow(True)
chart.set_xlabel('moodle_data.userid')
chart.set_ylabel('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.show()
```

Listing 9: Menge der Log-Einträge pro Benutzer

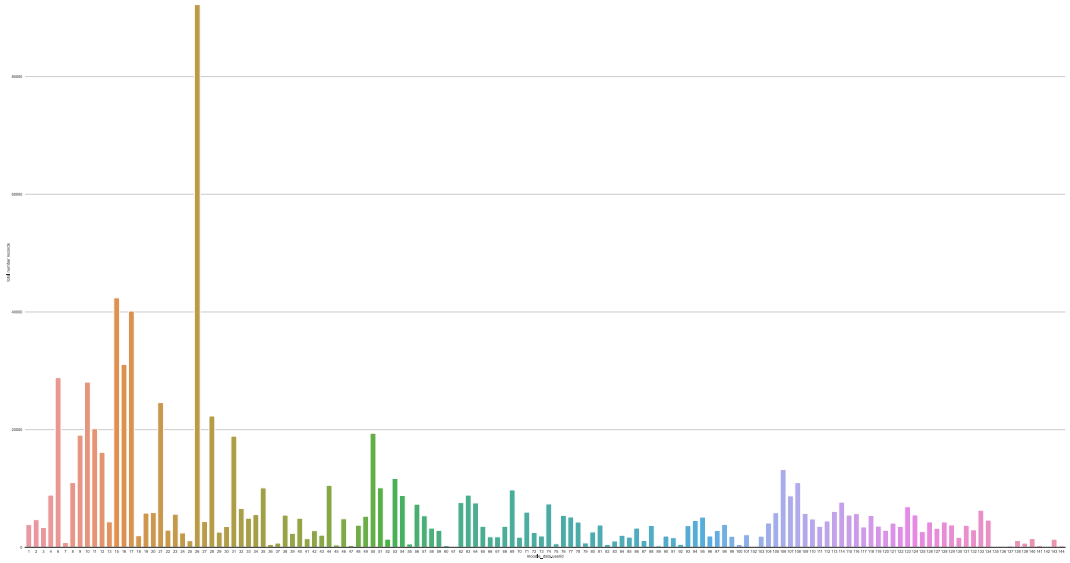

Abbildung 12.: Menge der Log-Einträge pro Benutzer (s. Anhang)

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

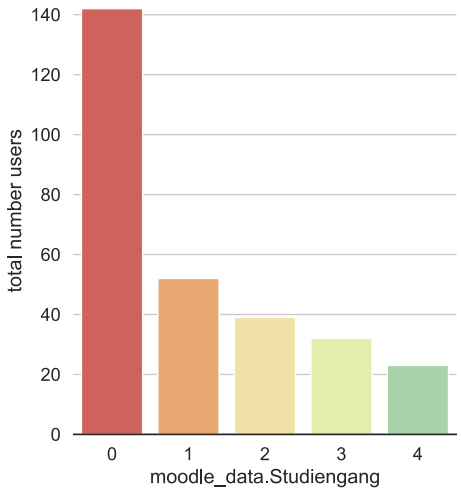

Abbildung 13.: Menge der Benutzer pro Studiengang

Evaluierung

Die Abbildung zur Menge der Benutzer pro Studiengang präsentiert 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

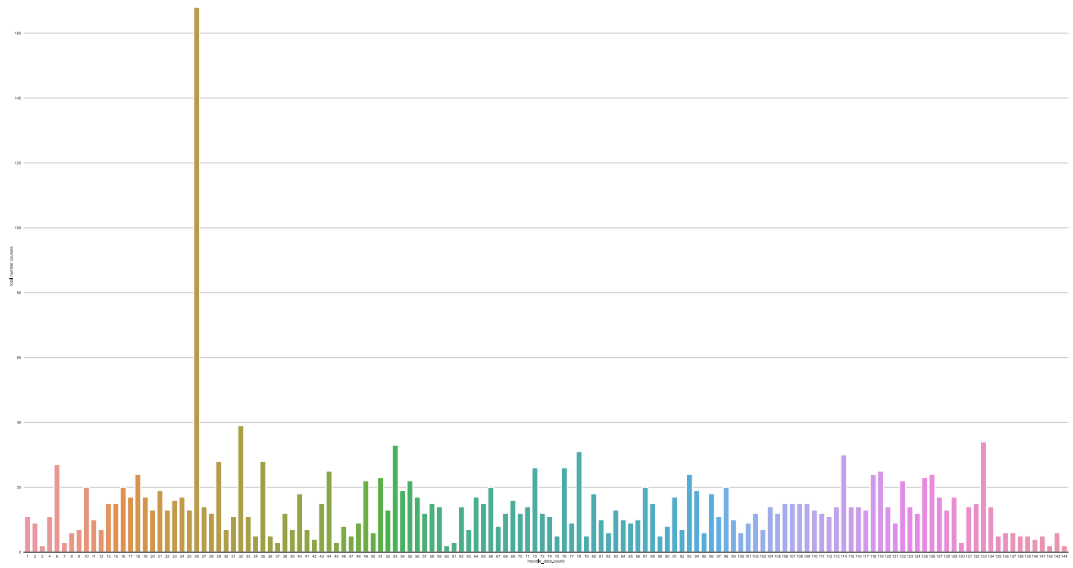

Abbildung 14.: Menge der Kurse pro Benutzer (s. Anhang)

Evaluierung

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 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 herzustellen.

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.

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 fesgelegt. 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 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

[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.

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*:

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

Listing 12: Mengen aller Actions in der Gesamtbetrachtung

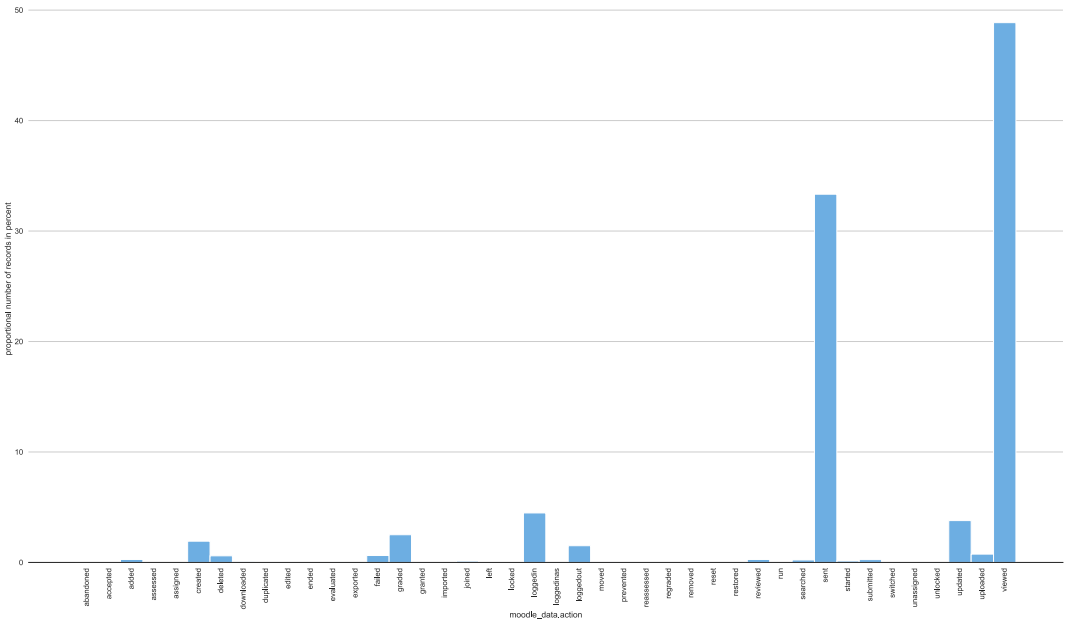

Abbildung 15.: Mengen aller Actions in der Gesamtbetrachtung (s. Anhang)

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.

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

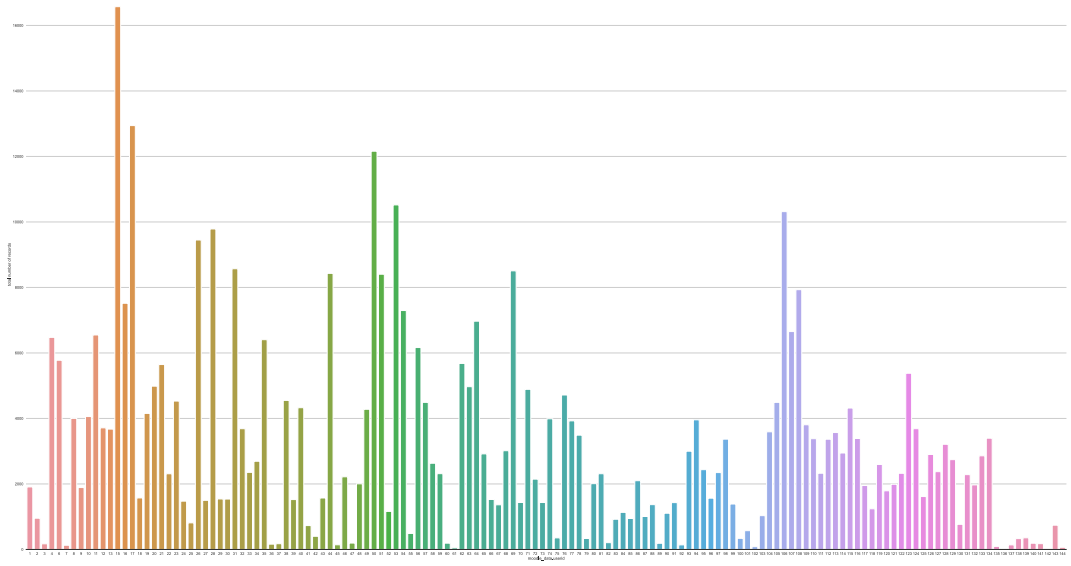

Abbildung 16.: Menge der viewed-Actions pro Benutzer (s. Anhang)

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 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

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

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

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 sorg-***

fältig 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 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.

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',
4                     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

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

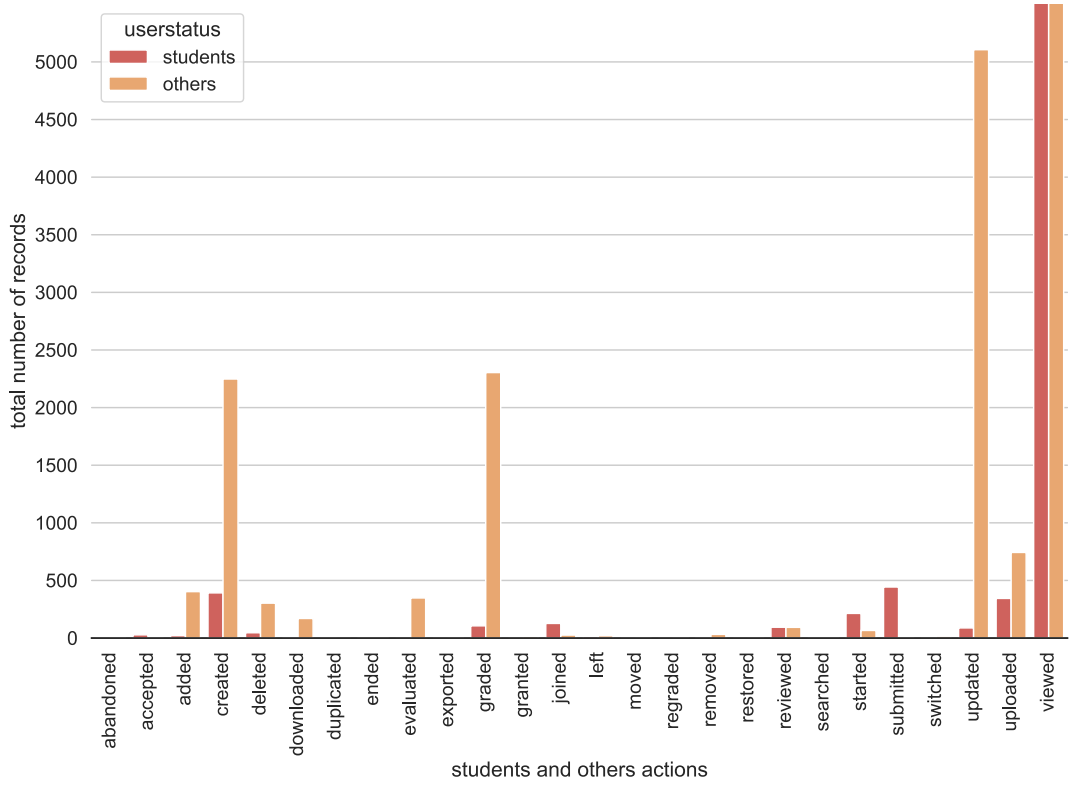

Abbildung 19.: Menge der Log-Einträge pro Aktivität und Benutzergruppe

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 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.***

Einzeln 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 herauszufiltern. 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 einesteils 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

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

72 rows in set (10,24 sec)

Abbildung 20.: Identifikation von Studenten

Studenten erfasst wurden, die eventuell erst später auf Anfrage[16] der Beobachtung ihres Verhalten zugestimmt haben, wäre bei Bedarf noch zu prüfen, soll hier für die

[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.

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.

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 der 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 Einzelanalysen[17] exemplarisch geprüft bis schließlich eine Menge von insgesamt 72 Studenten bestätigt werden konnte.

Abschließend wurden die identifizierten Studenten mit ihren charakteristischen Akitivitätsprofilen in einer eigenen Tabelle zusammengefasst und im Gesamtdatenbestand eindeutig gekennzeichnet, so dass sie zur Betrachtung ihres Verhaltens nun unmittelbar zur Verfügung standen.

3.2. Zeitbezogene Untersuchungen

Nachdem im vorausgegangenen Kapitel die Identität von Studenten festgestellt werden konnte, soll nun in diesem Kapitel deren Lern- und Kommunikationsverhalten in zeitlicher Hinsicht untersucht werden.

3.3. Aktivitätsbezogene Untersuchungen

...

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

4. Ergebnisse

...

5. Fazit

...

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/`

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

```
from sqlalchemy import create_engine
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from IPython.core.display_functions import display
```

Listing 24: Import von Bibliotheken und anderen Erweiterungen

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

Listing 25: Definitionen zur Darstellung der Visualisierungen

```
user = "****"
password = "****"
host = "localhost"
database = "vfh_moodle_ws20"
port = 3306

engine = create_engine(f'mysql+pymysql://{user}:'
                       f'{password}@{host}/{database}',
                       pool_recycle=port)
connection = engine.connect()
```

Listing 26: Herstellung der Verbindung zur MySQL-Datenbank

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

Listing 27: Import der Arbeitsdaten aus der MySQL-Datenbank

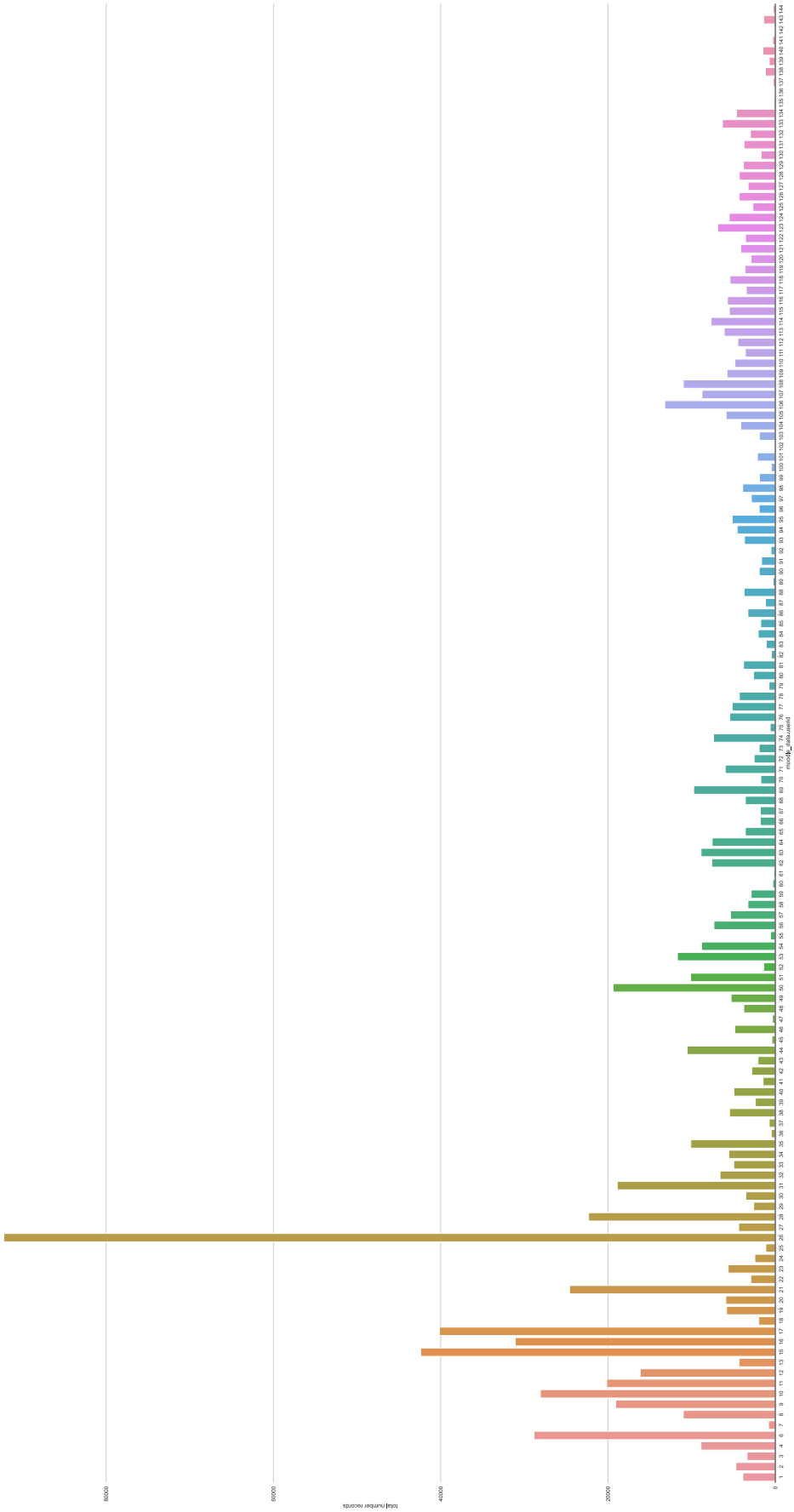

Abbildung 23.: Menge der Log-Einträge pro Benutzer

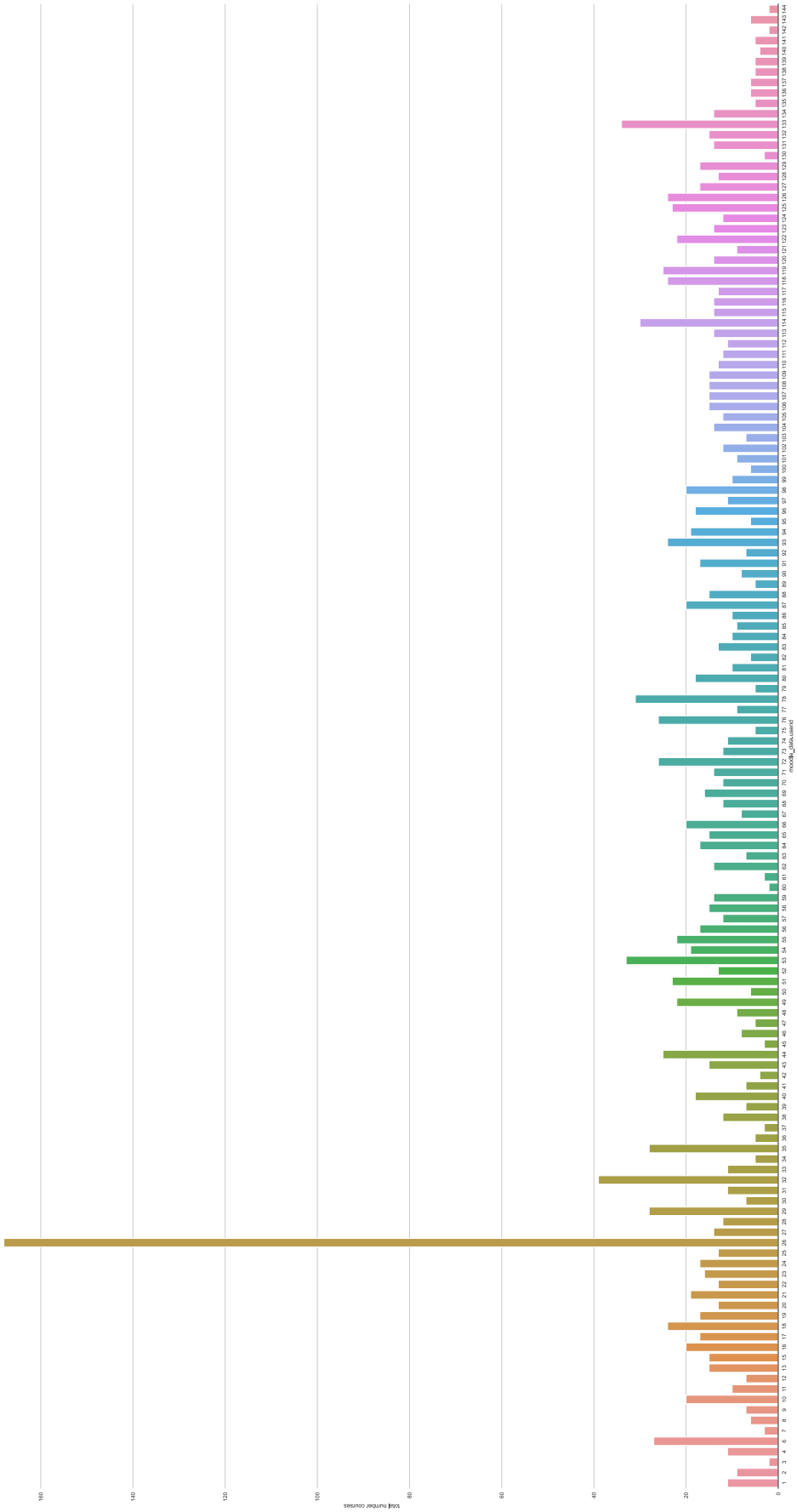

Abbildung 24.: Menge der Kurse pro Benutzer

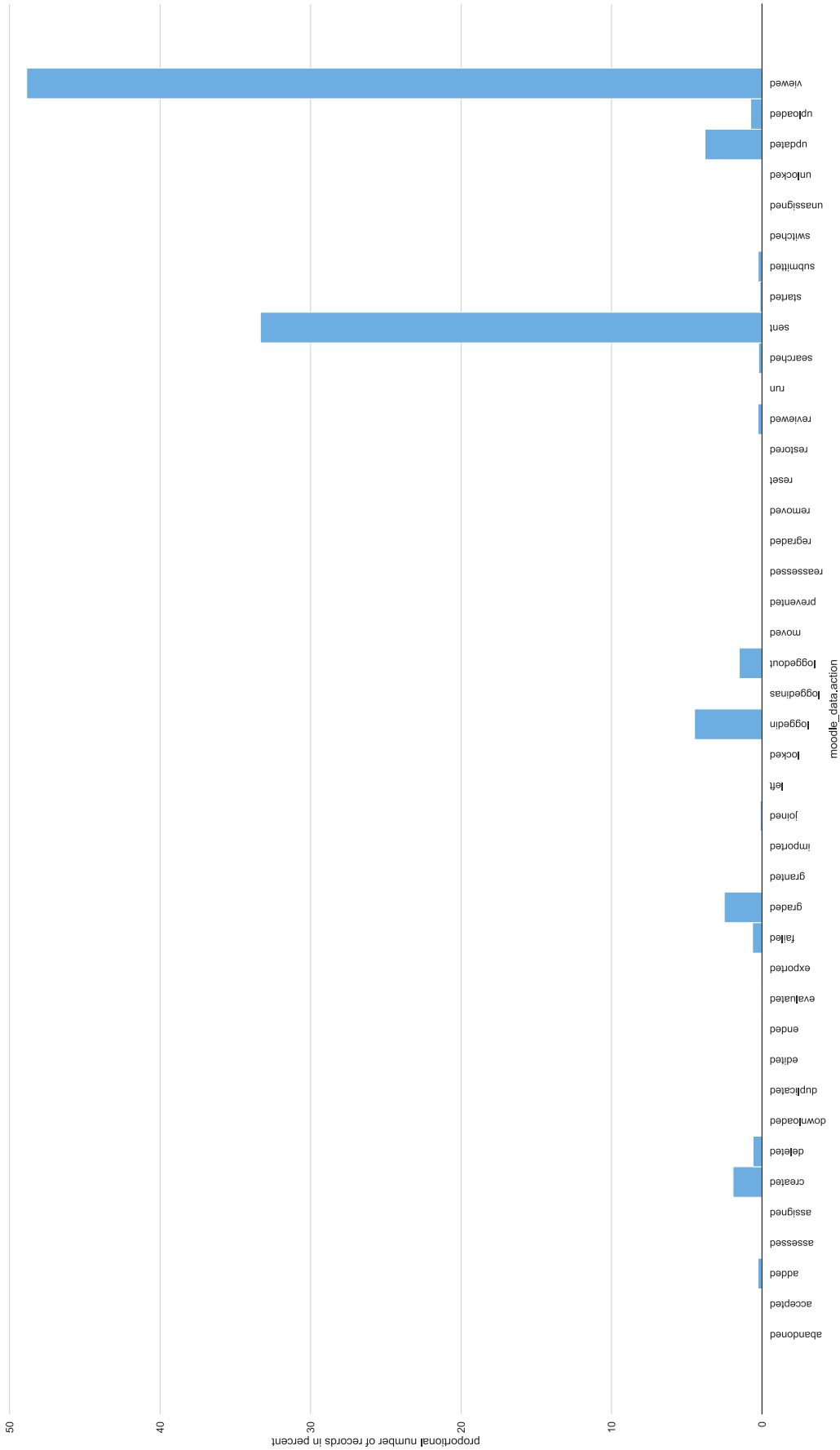

Abbildung 25.: Mengenverteilung aller Actions in der Gesamtbetrachtung

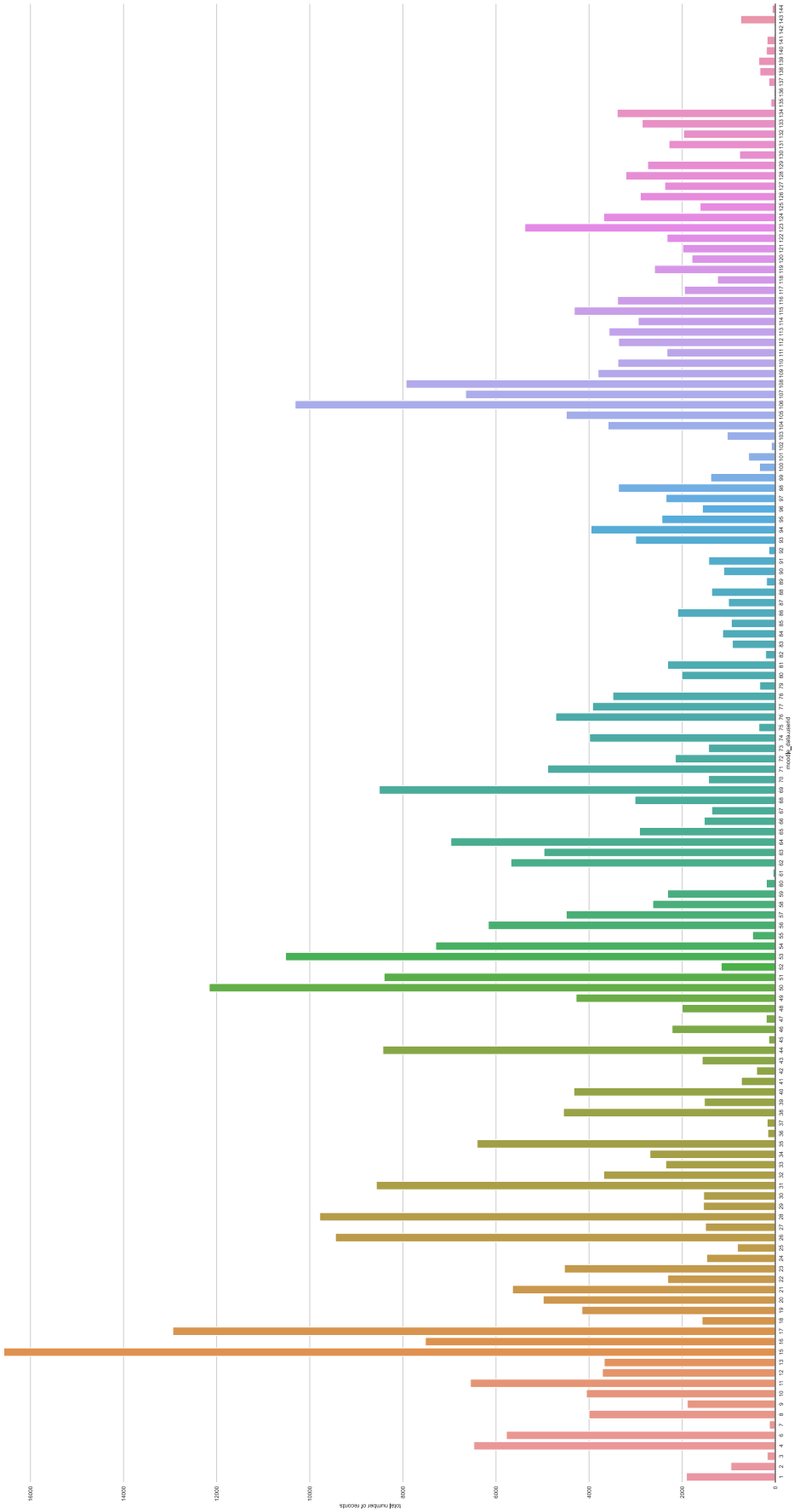

Abbildung 26.: Menge der viewed-Actions pro Benutzer

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

Verzeichnis / Beschreibung
`/1_` ...
`/2_` ...
`/3_` ...