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.

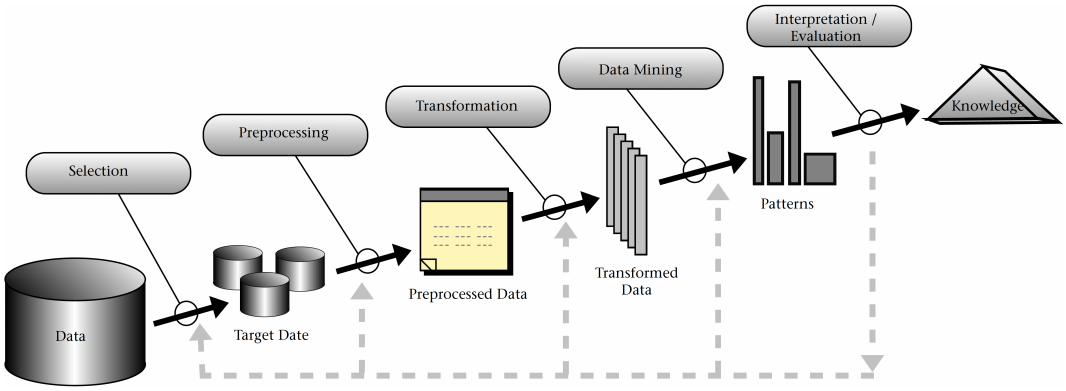

Abbildung 1: Phasen des KDD-Prozesses. Original von Fayyad et al. (1996).

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).

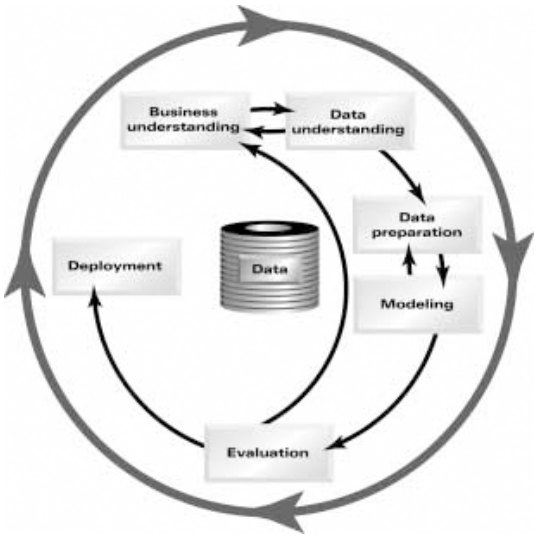

Abbildung 2: Phasen des CRISP-DM. Original von Shearer (2000).

[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.

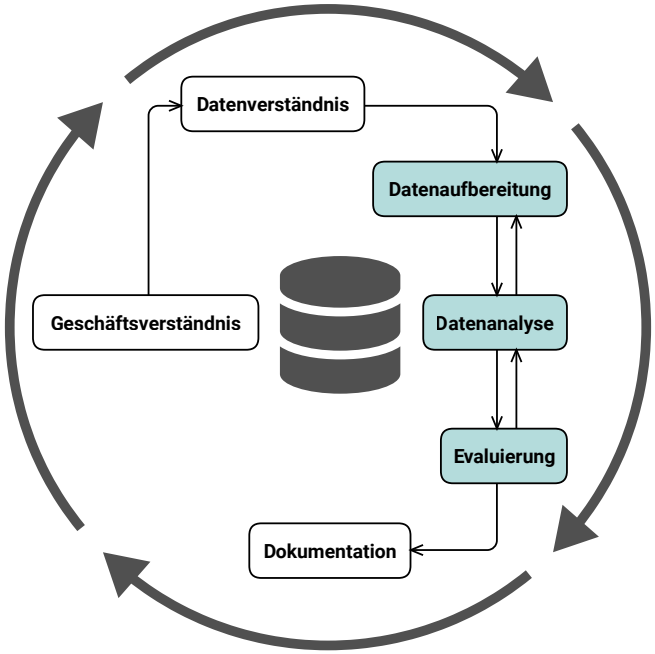

Abbildung 4: Phasen des verwendeten Vorgehensmodells.

[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 durch die *Virtuelle Fachhochschule* (*VFH*) zur Verfügung gestellter anonymisierter Datenbestand aus dem Wintersemester 2020/2021[9]. Hierin enthalten sind die Moodle-Nutzungsdaten von Studenten, Dozenten und anderem Personal 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 seitens der VFH aus der Datenbank des Moodle-Systems (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 unmittelbar erkennen zu können. Daneben wurden vorab die Merkmale *relateduserid*, *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

[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

typischen CSV-Format exportiert. Übergeben wurden die CSV-Daten schließlich als offene und komprimierte Textdateien in ASCII-Kodierung (Cerf, 1969), in der die Daten entgegen der üblichen Praxis jedoch nicht durch Kommata, sondern durch Semikola strukturiert waren.

Die freie Wahl eines Trennzeichens ist beim CSV-Format möglich, weil dieses nur allgemein beschreibt, wie die Tupel einer Relation und darin enthaltene Werte in der Regel interpretiert werden. Das Format definiert aber keinen verbindlichen Standard (Shafranovich, 2005), so dass die Daten entgegen ihrer Definition als Comma-Separated Values nicht zwingend nur durch Kommata zu strukturieren sind.

Der zur Verfügung gestellte Datenbestand umfasst insgesamt 969032 Datensätze. Dabei handelt es sich genau betrachet um eine spezifische Teilmenge von Loggings auf dem Moodle-Server der VFH, 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.

Formale Angaben über die Daten

Erste interessante Einblicke in die Art, den Umfang und die Struktur der zu untersuchenden Daten ergeben sich nach deren Import in eine MySQL-Datenbank durch einfache statistische SQL-Abfragen:

Abfrage zu Art und Umfang der implementierten MySQL-Datenbank

```
mysql> SELECT TABLE_SCHEMA, TABLE_NAME, ENGINE,
      (SELECT COUNT(*) FROM moodle_data) AS TABLE_ROWS, TABLE_COLLATION
       FROM information_schema.tables WHERE table_name = "moodle_data";
+-----------------+-------------+--------+------------+------------------+
| table_schema    | table_name  | engine | table_rows | table_collation  |
+-----------------+-------------+--------+------------+------------------+
| vfh_moodle_ws20 | moodle_data | InnoDB |     969032 | ascii_general_ci |
+-----------------+-------------+--------+------------+------------------+
```

Die Ergebnistabelle zeigt einen Ausschnitt der Metadaten, die standardmäßig vom MySQL-Server in der Datenbank INFORMATION_SCHEMA zu jeder verwalteten

Datenbank gespeichert werden.[10] Die aufgelisteten Werte informieren u. a. über das Speichersystem *engine*, die Anzahl der Datensätze *table_rows* und die *table_collation*, die definiert, nach welchen Regeln Zeichenketten miteinander verglichen werden. Übereinstimmend mit der ASCII-Kodierung der CSV-Daten wurde bei Erstellung der Datenbank *vfh_moodle_ws20* hier die Kollation *ascii_general_ci* gewählt.

Abfrage zu Struktur und Inhalt der importierten Originaldaten

```
mysql> DESCRIBE moodle_data;
+--------------------+-------------+------+-----+---------+-------+
| 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    |       |
+--------------------+-------------+------+-----+---------+-------+
```

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]

[10] Siehe auch die MySQL Documentation: 24.1 Introduction, MySQL 5.7 Reference Manual, 05/2022
[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. *Fremdschlüssel zur Identifikation eines bestimmten Benutzers (z. B. der Sender einer Nachricht) in einer von der VFH für diesen Zweck bereitgestellten weiteren Relation.*
relateduserid	Kennzahl zur Identifikation eines weiteren Benutzers. *Fremdschlüssel 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.

Abfrage zur Menge aller Benutzer

```
mysql> SELECT COUNT(DISTINCT userid) AS "total_number_users"
       FROM moodle_data;
+--------------------+
| total_number_users |
+--------------------+
|                144 |
+--------------------+
1 row in set (0,00 sec)
```

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). Abzüglich dieser beiden Gruppen erhielte man im Ergebnis lediglich 142 Einzelpersonen.

Abfrage zur Menge der Log-Einträge pro Benutzer

```
mysql> SELECT userid, COUNT(userid) AS "total_number_records"
       FROM moodle_data
       GROUP BY userid;
+--------+----------------------+
| 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)
```

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.

Abfrage zur Menge der Benutzer pro Studiengang

```
mysql> SELECT Studiengang, COUNT(DISTINCT userid) AS "total_number_users"
       FROM moodle_data
       GROUP BY Studiengang;
+-------------+--------------------+
| Studiengang | total_number_users |
+-------------+--------------------+
| 0           |                144 |
| 1           |                 54 |
| 2           |                 40 |
| 3           |                 33 |
| 4           |                 25 |
+-------------+--------------------+
5 rows in set (0,46 sec)
```

In der Ausgabe enthalten ist neben den oben genannten Studiengängen 1 bis 4 überraschenderweise ein weiterer Studiengang 0. Hierbei handelt es sich jedoch nicht um einen Studiengang wie die anderen vier, sondern um eine spezifische Entität, die sich nur auf Aktivitäten bezieht, die außerhalb des eigentlichen Kursgeschehens stattfanden, z. B. Logins, Chats oder Aufrufe des Kalenders bzw. Dashboards.

Bemerkenswert ist auch, dass dem Studiengang 0 alle zuvor ermittelten Benutzer zugeordnet sind, die Summe der Benutzer 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 grundsätzlich nicht am eigentlichen Lehrbetrieb teilnehmen.

Abfrage zur Menge der Kurse pro Benutzer

```
mysql> SELECT userid, COUNT(DISTINCT courseid) AS "total_number_courses"
       FROM moodle_data
       GROUP BY userid
       ORDER BY total_number_courses;
+--------+----------------------+
| 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)
```

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 mit einer Dozententätigkeit zu begründen sein oder auf eine administrative 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.

Abfrage zu Benutzern mit überdurchschnittlich vielen Kursen

```
mysql> SELECT userid, COUNT(DISTINCT courseid) AS "total_number_courses"
       FROM moodle_data
       WHERE userid > 0
       GROUP BY userid
       HAVING total_number_courses >= 12
       ORDER BY total_number_courses;
+--------+----------------------+
| 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)
```

Abfrage zur Menge der Studiengänge 1 bis 4 pro Benutzer

```
mysql> SELECT userid, COUNT(DISTINCT Studiengang) AS "total_number_studies"
       FROM moodle_data
       WHERE Studiengang > 0 AND userid > 0
       GROUP BY userid
       HAVING total_number_studies > 1
       ORDER BY total_number_studies;
+--------+----------------------+
| 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)
```

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.[12]

Um das Vorgehen vollständig abzubilden, werden in diesem Analysebeispiel auch die Vorbereitungen im einleitenden Prolog exemplarisch vorgestellt. Die folgenden Listings zeigen u. a. die erforderlichen Anweisungen zur Einrichtung der Arbeitsumgebung oder dem Import der Arbeitsdaten. Bei den weiteren Untersuchungen in diesem Abschnitt und im Hauptteil werden diese stets vorausgesetzt, und nur in besonderen Fällen wird darauf hingewiesen.

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 1: Import von Bibliotheken und anderen Erweiterungen

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

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

Listing 2: Definitionen zur Darstellung der Visualisierungen

```
user = "root"
password = "root"
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 3: Herstellung der Verbindung zur MySQL-Datenbank

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

Listing 4: Import der Arbeitsdaten aus der MySQL-Datenbank

Datenaufbereitung

```
# Konvertierung des Datentyps des Tabellenmerkmals timecreated
moodle_data['timecreated'] =
    pd.to_datetime(moodle_data['timecreated'], unit='s')

# Gegenstand der Untersuchungen sind 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).
moodle_data = moodle_data[moodle_data.userid > 0]
moodle_data
```

Listing 5: Auswahl der Arbeitsdaten

Datenanalyse: Menge der Log-Einträge pro Benutzer

```
# Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(32, 16)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# 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 6: Menge der Log-Einträge pro Benutzer

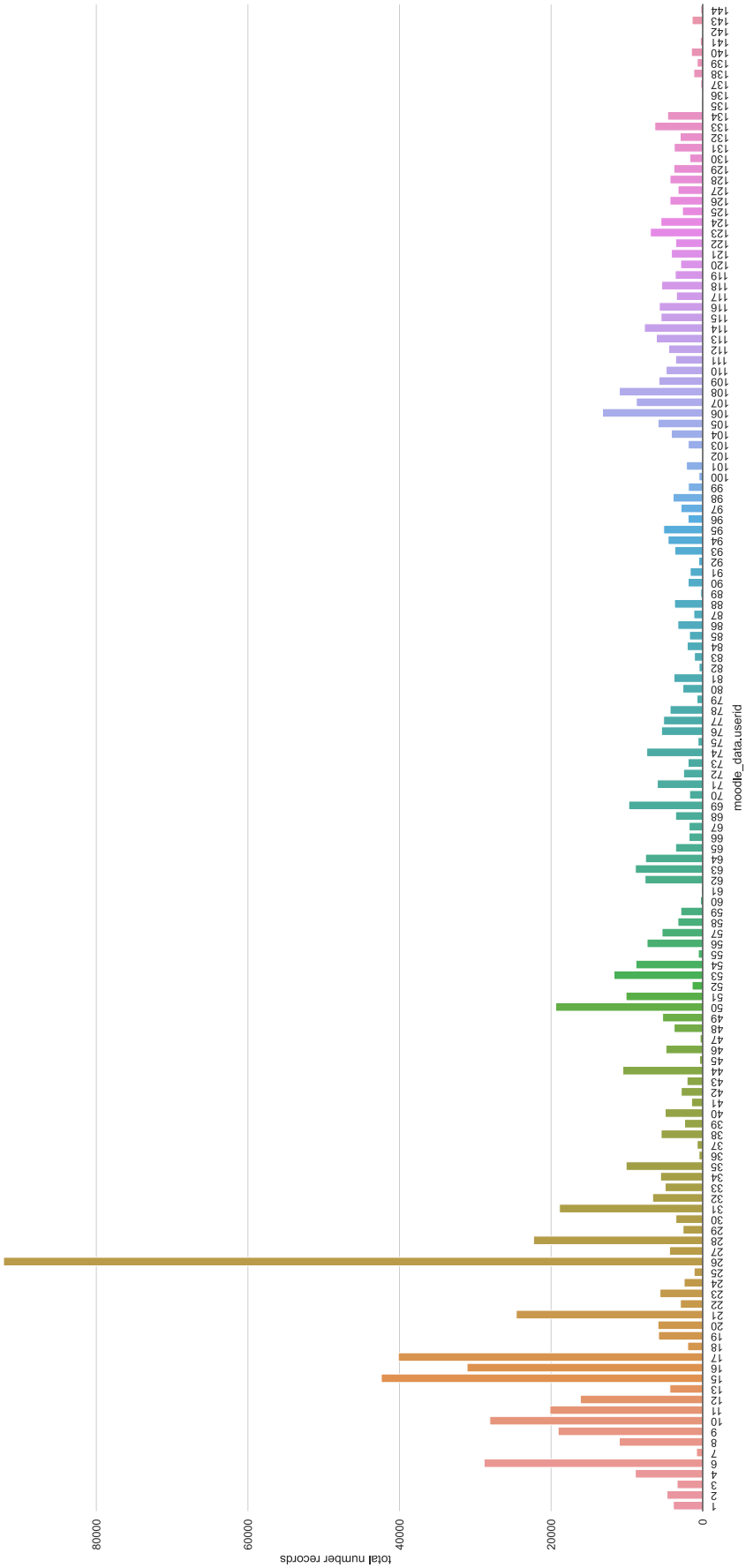

Abbildung 5: Menge der Log-Einträge pro Benutzer

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 von höheren 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 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 7: Menge der Benutzer pro Studiengang

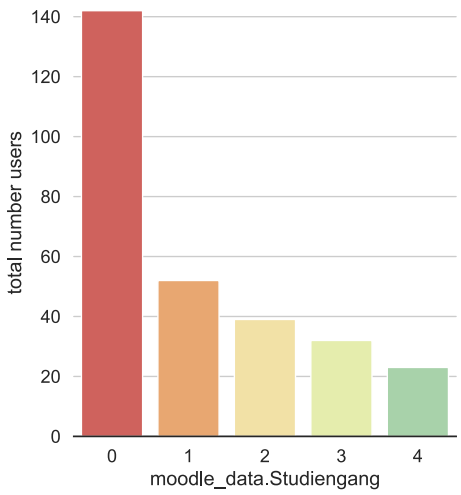

Abbildung 6: Menge der Benutzer pro Studiengang

Evaluierung: *Die vorherige Grafik zur Menge der Benutzer pro Studiengang präsentiert nicht nur die reinen Zahlen, die auch die entsprechende Ergebnistabelle im voherigen Abschnitt bereits auflistete. Sie verdeutlicht darüberhinaus sehr schnell gerade auch die Größenverhältnisse zwischen den einzelnen Werten des Diagramms auf eindrucksvollsvolle Weise. 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. Zur besseren Lesbarkeit sind die Erläuterungen und Hinweise zu den einzelnen Analyseteilen dann, wie schon in diesem Abschnitt, stets *in kursiver Schrift* gesetzt.

Um in dieser Arbeit auch größere Visualisierungen leicht verständlich abbilden zu können, sollen diese nach Möglichkeit nicht unterteilt, sondern eher in übersichtlicher und kompakter Form auf maximal einer Seite präsentiert werden. Dies bedingt mitunter unterschiedliche und manchmal auch recht kleine Schriftgrößen und geht daher gelegentlich auch zu Lasten der Lesbarkeit. In solchen Fällen sei auch noch einmal auf die Plots in den beigefügten Jupyter Notebooks verwiesen.

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 8: Menge der Kurse pro Benutzer

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.*

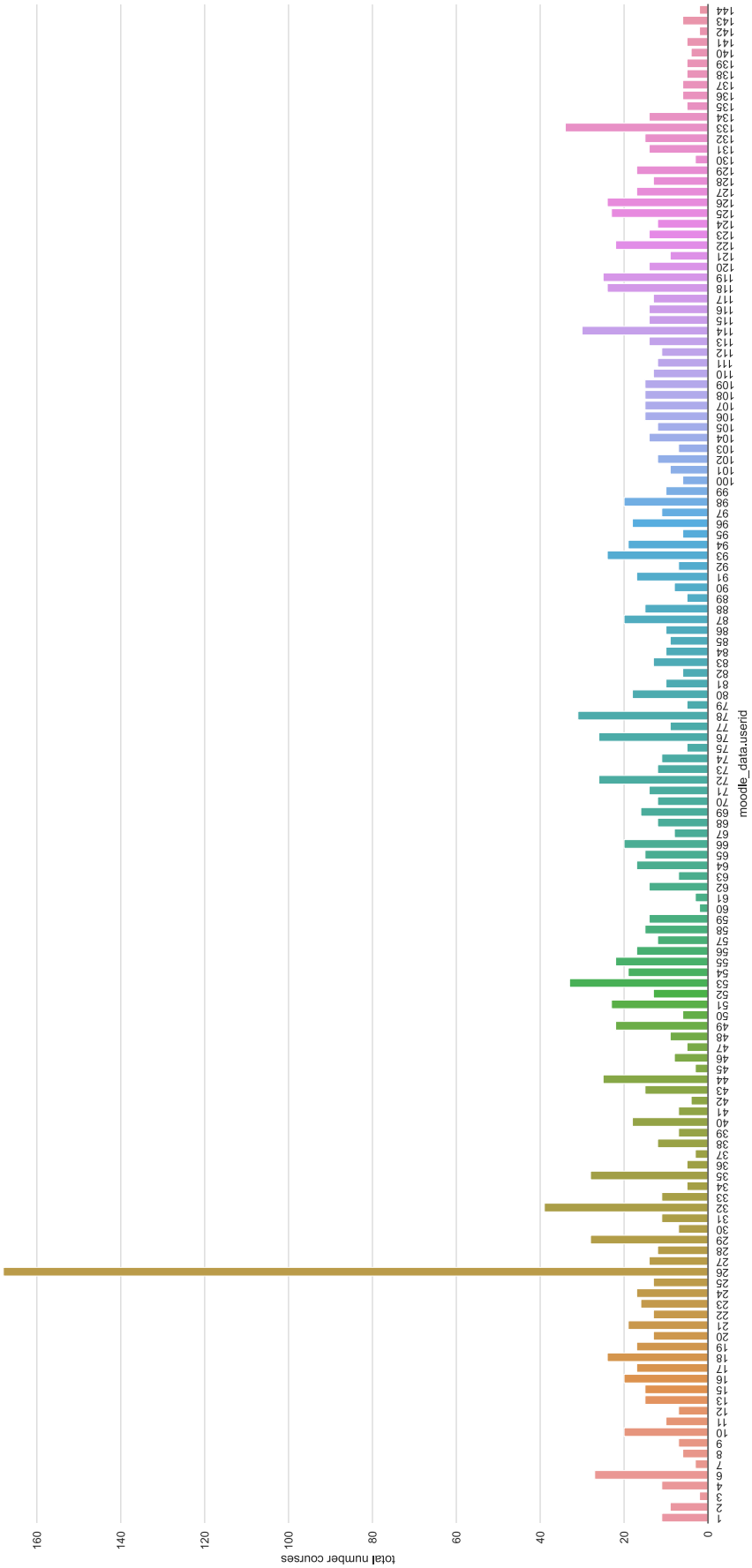

Abbildung 7: Menge der Kurse pro Benutzer

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 soll.

Nach Informationen der Hochschule, wird in Moodle die Rolle eines Benutzers nur auf Kursebene fesgelegt. Das bedeutet, dass ein Benutzer in mehreren Kursen auch verschiedene Rollen einnehmen kann, ganz unabhängig von seinem offiziellen Status. Diese sehr flexible Rollenzuweisung ist für die Zwecke des DiSEA-Projekts allerdings nur bedingt geeignet.

Ein erster Teil der Daten wurde folglich erst nach einer Umfrage erhoben, bei der die Benutzer um ihr Einverständnis gebeten wurden. Auf diese Weise konnte zwar eine Menge von *75 Studenten* gesichert ermittelt werden. Menge und Qualität der gewonnenen Informationen waren jedoch nicht ausreichend, und mussten daher in einem zweiten Schritt durch einen Kontext von Daten interagierender *Personen, deren offizieller Status aber unklar ist*, zusätzlich ergänzt werden.

Generell besitzt die Datenqualität 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 daher einvernehmlich mit dem Betreuerteam entschieden, jene Datensätze die sich nicht sicher auf Aktivitäten von Studenten beziehen bei den anschließenden Untersuchungen gesondert zu behandeln.

Die praktische Unterscheidung von Studenten und anderen Benutzern, wie z. B. Dozenten, Studiengangskoordinatoren oder weiterem Personal der Hochschulen, erfolgte anschließend gemäß der nachfolgenden Schritte.

Untersuchungen verschiedener Tabellenmerkmale

Mehrere Überlegungen zu der einleitenden Frage, durch was sich nun ein typisch studentisches Verhalten tatsächlich auszeichnen könnte, führten zunächst zu zahlreichen testweisen Untersuchungen über die verschiedensten Merkmale des Datenbestands. Während manche Betrachtungen insbesondere in zeitlicher Hinsicht auf den ersten Blick wenig aufschlussreiche Ergebnisse lieferten, fiel bei Untersuchung des Merkmals *action* schnell auf, dass bestimmte Benutzer oft einen hohen Anteil an Werten vom Typ *viewed* aufwiesen.

Benutzer mit hohem Anteil an viewed-Actions

```
mysql> SELECT userid, COUNT(action) AS "all_actions",
            (SELECT COUNT(action) FROM moodle_data md2
             WHERE md2.userid = md1.userid AND md2.action = "viewed")
             AS "viewed"
       FROM moodle_data md1
       GROUP BY userid
       HAVING viewed > (0.8 * all_actions)
       ORDER BY viewed DESC;
```

userid	all_actions	viewed
53	11699	10520
69	9756	8507
44	10536	8430
51	10118	8404
54	8813	7295
64	7544	6970
56	7335	6165
20	5909	4986
71	5985	4889
76	5434	4716
38	5478	4551
23	5634	4531
57	5361	4491
40	4953	4328
49	5286	4280
94	4561	3958
13	4330	3675
104	4136	3592
78	4300	3490
98	3894	3368
68	3579	3015
93	3685	3000
65	3576	2918
58	3268	2632
97	2861	2347
59	2885	2314
72	2526	2147
96	1928	1566
66	1795	1526
70	1727	1434
...	...	...

35 rows in set (6,34 sec)

Dass nun diese Besonderheit für das Verhalten von Studenten charakteristisch sein könnte, es vielleicht weitere Indizien dieser Art geben könnte, oder solche die für Studenten dagegen relativ untypisch sind, legte schließlich den Gedanken nahe, dass die Einordung der Benutzer in die beiden großen Kategorien Studenten und Hochschulpersonal mittels einer genaueren Betrachtung spezifischer Aktivitäten zu realisieren sein müsste.

Betrachtung von Kursprofilen einzelner Benutzer

Interessant war in diesem Zusammenhang auch die bereits im Grundlagenkapitel durchgeführte Analyse zur Menge der offiziellen Studiengänge (Studiengänge > 0) pro Benutzer. Die Betrachtung von Kursprofilen einzelner Benutzer, für die bereits ein Dozentenstatus vermutet wurde, war so wie beim Benutzer mit der userid 26 sehr aufschlussreich.

Menge an Log-Einträgen pro Kurs für Benutzer 26

```
mysql> SELECT courseid, COUNT(courseid) AS "total_number_records"
       FROM moodle_data
       WHERE userid = 26
       GROUP BY courseid
       ORDER BY total_number_records DESC;
+----------+----------------------+
| courseid | total_number_records |
+----------+----------------------+
|        0 |                83088 |
|    24044 |                  814 |
|    11807 |                  762 |
|        1 |                  732 |
|     2466 |                  612 |
|     1750 |                  562 |
|     4213 |                  360 |
|     1335 |                  346 |
|     4286 |                  345 |
|     1317 |                  320 |
|     4212 |                  292 |
|     4276 |                  244 |
|      ... |                  ... |
+----------+----------------------+
168 rows in set (0,31 sec)
```

Durch Ergänzung der URL https://moodle.oncampus.de/enrol/index.php?id= mit der courseid konnten für die drei Kurse mit den meisten Log-Einträgen dann im Anschluss auch die entsprechenden Moodle-Webseiten ermittelt werden (courseid 0 und 1 sind an dieser Stelle nicht relevant):

- courseid 24044: VFH Verbundmanagement (VFH FV Organisation)
- courseid 11807: VFH Studiengangskoordination
- courseid 2466: VFH Team (VFH FV 00 [01])

Weitere Personen wie z. B. die Benutzer mit der userid 21 und 32, die in mehr als nur einem der offiziellen Studiengänge aktiv waren oder die ebenfalls einen Kurs des Benutzers 26 besuchten, wurden anschließend stichprobenartig einzeln überprüft. Auch sie wiesen oft verhältnismäßig große Mengen an Log-Einträgen für courseids im drei- und vierstelligen Bereich auf.[13]

Bei testweisen Analysen von Kursprofilen einzelner Benutzer mit hohen Anteilen an viewed-Actions (s. o.), ergab sich dann interessanterweise ein ganz anderes Bild. Im wesentlichen waren diese Personen in Kursen mit relativ hohen courseids im fünstelligen Bereich ab ca. 25000 aktiv. Die Untersuchung des Benutzers mit der userid 53 ergab z. B. folgendes Resultat:

Menge an Log-Einträgen pro Kurs für Benutzer 53

```
mysql> SELECT courseid, COUNT(courseid) AS "total_number_records"
       FROM moodle_data
       WHERE userid = 53
       GROUP BY courseid
       ORDER BY total_number_records DESC;
+----------+----------------------+
| courseid | total_number_records |
+----------+----------------------+
|        0 |                 3718 |
|    27500 |                 1249 |
|     4245 |                 1016 |
|    28259 |                  968 |
|    27512 |                  778 |
|    27499 |                  606 |
|    28256 |                  570 |
|    27498 |                  541 |
|    27501 |                  527 |
|    27515 |                  262 |
|    27503 |                  259 |
|     4217 |                  188 |
|    27502 |                  161 |
|    28258 |                  132 |
|    28229 |                  120 |
|      ... |                  ... |
+----------+----------------------+
33 rows in set (0,03 sec)
```

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

Die Sichtung der entsprechenden Webseiten in Moodle bestätigte die Vermutung: Bis auf nur wenige Ausnahmen wurden von dem Benutzer 53 fast nur Fachkurse, hier im Bachelor-Studiengang Wirtschaftsingenieurwesen, besucht:

- courseid 27500: Technical English (BHTB WIG 18 W20)
- courseid 4245: Fachbereichskurs WIG (BHTB) (TFHB V)
- courseid 28259: Methodische Produktentwicklung (BHTB WIG 18 S21)
- courseid 27512: Controlling (BHTB WIG 18 W20)

Zu beachten ist hier aber nicht nur die courseid und der Name des Kurses, sondern auch die Angabe zum jeweiligen Semester am Ende der Kursbezeichnung.

Mit Hilfe dieser Information (und der Menge aller courseids) war es dann mit wenigen manuellen Stichproben leicht möglich, die Menge der Fachkurse im Untersuchungszeitraum auf Kurse mit einer ***courseid >= 27040*** einzugrenzen und damit eine ***neue notwendige Bedingung zur Identifikation von Studenten*** zu schaffen.

Untersuchung der Aktivitäten von Dozenten und Studenten

Gemäß den vorab gewonnenen Erkenntnissen wurden in einem nächsten Schritt bestimmte Benutzer ausgewählt und deren Log-Einträge in neuen Test-Datensets für mutmaßliche Studenten und Dozenten zusammengefasst (Dozenten sind hier stellvertretend für das gesamte Hochschulpersonal zu betrachten):

```
1 md = moodle_data # Umbenennung der Variable, um den Code zu verkürzen
2 records_teachers = [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_teachers = pd.concat(records_teachers)
```

Listing 9: Auswahl der Log-Einträge der Dozenten

```
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 10: Auswahl der Log-Einträge der Studenten

```
# Ermittlung der Menge der Log-Einträge pro Action
teachers_actions = md_teachers.action.groupby(md.action).count()
students_actions = md_students.action.groupby(md.action).count()

# Erstellung eines kombinierten Datensets für Dozenten und Studenten
users_actions =
    pd.concat([teachers_actions, students_actions], axis=1,
                keys=['teachers', 'students']).sort_index()

# Ersetzung von NaN-Werten durch den Wert 0
users_actions = users_actions.fillna(0)

# Ausgabe des kombinierten Datensets
display(users_actions)
```

Listing 11: Konkatenation der Test-Datensets von Dozenten und Studenten

Die Tabelle unten zeigt als Ergebnis des obigen Listings die Menge der Log-Einträge pro Aktivität und Benutzergruppe. Zeilen mit Werten, die im weiteren Verlauf der Untersuchung dann noch von größerem Interesse waren, sind hierbei mit einem Pfeil markiert:

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

Die nachfolgende Visualisierung veranschaulicht noch einmal deutlich die bereits in der Ergebnistabelle markierten Wertdifferenzen. Zu beachten ist, dass die Action mit dem Wert viewed im Diagramm nicht enthalten ist, um die Differenzen der anderen Werte in ihren Proportionen besser darstellen zu können:

```
# Entfernung der letzten Zeile betreffend die action viewed,
# um Differenzen anderer Werte besser visualisieren zu können.
users_actions = users_actions[:-1]

# Visualisierung der Menge der Log-Einträge
# pro Aktivität für Dozenten und Studenten
result = users_actions.stack().reset_index().set_index('action').
    rename(columns={'level_1': 'teachers', 0: 'students'})
chart = sns.barplot(x=result.index, y='students',
                    data=result, hue='teachers')
```

Listing 12: Menge der Log-Einträge pro Aktivität und Benutzergruppe

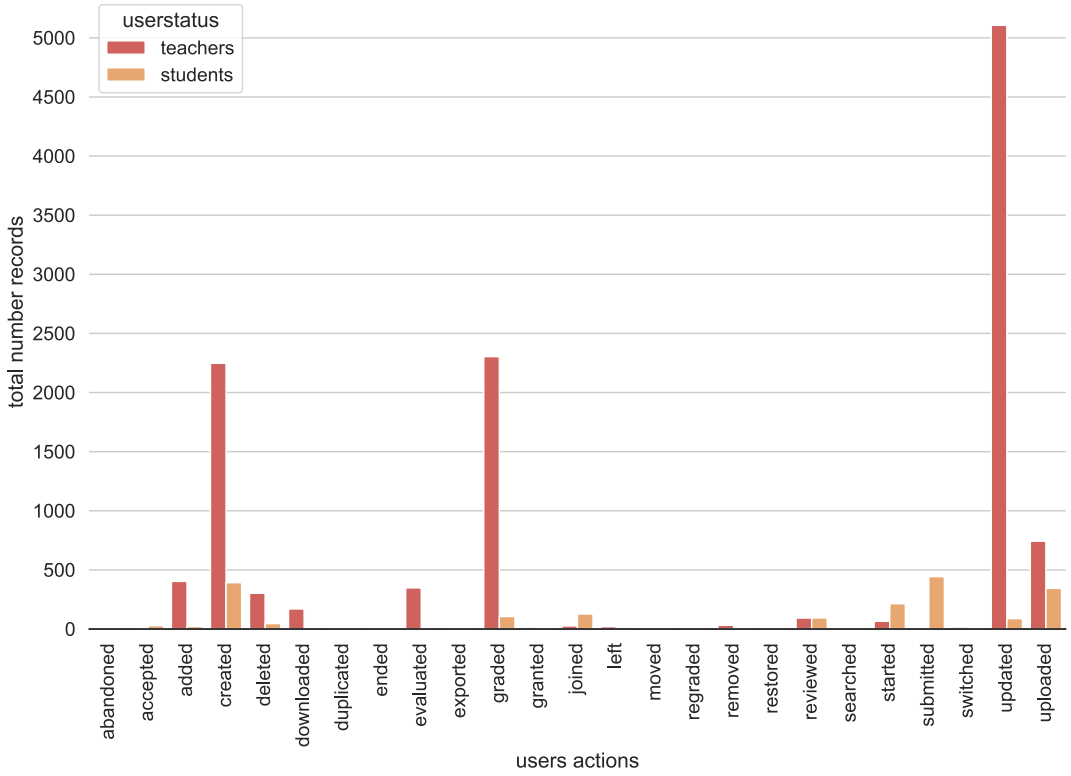

Abbildung 8: Menge der Log-Einträge pro Aktivität und Benutzergruppe

Wie die Ergebnistabelle und das korrespondierende Diagramm auf den ersten Blick zeigen, überragen z. B. bei den Werten *created*, *graded* und *updated* die Log-Einträge der Dozenten die der Studenten um ein Vielfaches, während man erst bei genauem Hinsehen erkennt, dass es sich beim Wert *submitted* genau andersherum darstellt.

Bestätigten also die Untersuchungen mittels vordefinierter Benutzergruppen die eingangs formulierte Vermutung, dass sich Studenten und Angehörige des Hochschulpersonals anhand ihrer Aktivitäten unterscheiden müssten, so stellte sich nun jedoch die Frage ob und wie sich mit dieser Erkenntnis die Studenten im Gesamtkontext auch auf direktem Wege identifizieren ließen.

Zur Beantwortung dieser Frage 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 wie die zuvor genannten Aktivitäten selbst eine gewisse Anzahl an Log-Einträgen aufwiesen, andererseits aber auch eine mindestens genauso beachtliche Mengendifferenz.

Spezifizierung der Auswahlkriterien

Die gesammelten Kriterien zur Identifikation von Studenten lauteten bis hierhin folgendermaßen:

1. Studenten werden nur als einzelne Personen betrachtet
2. Studenten waren in mindestens einem der offiziellen Studiengänge aktiv
3. Studenten waren über das Semester in mindestens einem der Fachkurse aktiv
4. Studenten besitzen einen hohen Anteil an viewed-Actions
5. Studenten besitzen nur einen geringen Anteil an added-, created-, deleted-, evaluated-, graded- oder updated-Actions
6. Studenten besitzen einen hohen Anteil an submitted-Actions

In dieser Phase waren zwar noch einige Untersuchungen notwendig, insbesondere wurden wiederholt Einzelanalysen von Benutzern und deren Kursprofilen durchgeführt, es wurde aber dennoch deutlich, wie die Lösung des Problems aussehen könnte: ***Eine direkte Identifikation von Studenten müsste über eine dem Gesamtkontext angemessene Gewichtung der ausgewählten Aktivitäten herzustellen sein.***

Einzeln oder in Gruppen zusammengefasst müssten die Mengen der Log-Einträge zu den Aktivitäten der Benutzergruppen mithin wie Stellschrauben justiert werden können, um die Studenten aus der Gesamtmenge der Benutzer herauszufiltern.

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 anzupassen. Hinzugefügt wurden so eine neue WHERE-Klausel, in der die obigen notwendigen Kriterien 1 bis 3 berücksichtigt wurden. Außerdem wurde die HAVING-Klausel um weitere Bedingungen ergänzt, die eine flexible Gewichtung der Log-Einträge zu den verschiedenen Aktivitäten (s. o. die Kriterien 4 bis 6) erlaubten:

Identifikation von Studenten

```
mysql> SELECT userid,
       COUNT(action) AS "all_actions",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "added") AS "added",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "created") AS "created",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "deleted") AS "deleted",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "graded") AS "evaluated",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "graded") AS "graded",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "submitted") AS "submitted",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "updated") AS "updated",
       (SELECT COUNT(action) FROM moodle_data md2
            WHERE md2.userid = md1.userid
                AND md2.action = "viewed") AS "viewed"
        FROM moodle_data md1
        WHERE !((userid < 0) OR (Studiengang = 0) OR (courseid < 27040))
        GROUP BY userid
        HAVING (viewed > (0.8 * all_actions)) AND
                    (((added + created + deleted + evaluated +
                       graded + updated) < (0.025 * viewed))
                       OR (submitted > (0.001 * viewed)));
```

userid	all_actions	added	created	deleted	evaluated	graded	submitted	updated	viewed
1	1648	0	43	0	0	0	12	20	1909
13	2945	2	40	2	15	15	51	11	3675
18	1261	2	17	1	0	0	24	14	1573
19	3470	2	77	10	24	24	75	11	4154
20	3637	3	58	3	19	19	55	10	4986
22	1864	1	26	0	0	0	22	5	2311
23	3943	5	76	3	35	35	106	12	4531
24	1017	0	17	36	0	0	13	3	1475
25	652	6	21	0	0	0	16	2	812
38	4212	0	46	5	13	13	94	10	4551
40	3546	0	43	9	9	9	44	21	4328
43	1043	1	5	0	2	2	8	4	1571
49	3979	6	51	5	57	57	97	77	4280
51	6691	1	49	2	17	17	58	157	8404
52	948	2	23	0	0	0	18	13	1162
53	6488	2	35	2	10	10	34	46	10520
54	6255	1	57	16	22	22	63	192	7295
56	5731	3	51	2	63	63	164	71	6165
57	3834	2	74	0	26	26	89	31	4491
58	2611	0	27	0	8	8	27	104	2632
59	2108	1	50	4	11	11	50	18	2314
62	4064	4	61	0	21	21	72	10	5680
64	5079	2	42	0	8	8	35	12	6970
65	2153	1	49	1	8	8	47	13	2918
66	1266	0	25	13	1	1	17	5	1526
67	1240	4	29	2	7	7	29	31	1363
68	2704	2	41	0	26	26	72	6	3015
69	7129	1	63	0	16	16	78	15	8507
70	1267	0	13	6	9	9	18	6	1434
71	3768	1	68	0	16	16	72	21	4889
72	1390	1	5	0	3	3	7	2	2147
73	853	0	3	0	0	0	3	0	1433
76	2958	1	12	0	2	2	12	0	4716
78	2815	1	35	2	3	3	23	17	3490
80	1654	2	47	0	2	2	24	48	2009
83	810	1	11	0	0	0	10	1	922
87	734	0	6	0	1	1	8	0	1006
91	1165	4	19	0	0	0	23	2	1430
93	2537	7	42	3	26	26	41	102	3000
94	3413	3	27	2	49	49	78	25	3958
96	1207	2	16	0	9	9	24	4	1566
97	2347	0	25	0	36	36	60	89	2347
98	2698	6	37	3	12	12	36	14	3368
99	1031	1	22	0	11	11	26	10	1387
104	3382	1	67	0	11	11	57	20	3592
105	3660	2	85	2	16	16	70	31	4491
107	6410	2	167	33	0	0	11	358	6653
109	4018	4	193	144	0	0	10	387	3807
111	2528	2	141	32	1	1	10	292	2327
112	3470	4	145	31	1	1	11	351	3366
113	3842	2	254	67	0	0	9	377	3570
115	4639	4	166	29	1	1	11	300	4318
116	3450	4	191	16	0	0	10	328	3388
117	2248	3	162	37	0	0	9	329	1952
119	2465	1	87	2	0	0	9	244	2595
120	2176	1	175	10	0	0	5	284	1791
122	2369	1	150	15	0	0	8	288	2325
123	4945	3	134	56	0	0	9	358	5380
124	3419	2	162	22	0	0	9	296	3687
125	1588	1	108	8	0	0	6	219	1615
126	2865	1	171	80	0	0	5	297	2900
127	2078	2	78	17	0	0	5	264	2375
128	3423	3	134	41	0	0	8	311	3211
129	2710	0	114	57	0	0	8	303	2742
131	2605	33	162	17	1	1	7	276	2284
132	2103	2	112	19	0	0	6	279	1971
134	3756	2	146	22	0	0	12	304	3397
136	13	0	0	0	0	0	2	0	26
143	459	0	11	0	0	0	4	2	741

69 rows in set (23,17 sec)

Die obige Tabelle umfasst im Ergebnis sämtliche Mengen an Log-Einträgen zu den ausgewählten Aktivitäten für insgesamt 69 Benutzer und liegt damit relativ nah an der von der Hochschule mittels einer Umfrage ermittelten Zahl von 75 Studenten.

Dennoch lässt sich an dieser Stelle aber nicht sicher sagen, dass es sich bei den Benutzern im obigen Ergebnis tatsächlich nur um Studenten handelt.

Wie es bei Analyseverfahren dieser Art generell der Fall ist, muss stets mit einer gewissen Unschärfe in den Ergebnissen gerechnet werden. Daher kann folglich das Ziel nur sein, in Anbetracht eines zu leistenden Aufwands eine möglichst optimale Annäherung an einen eventuell auch nur theoretischen Zielwert zu versuchen.

Das hier beschriebene Vorgehen folgte genau diesem Ansatz. Es erforderte bei der Auswahl der zu gewichtenden Aktivitäten wie auch bei der Feinjustierung der Bedingungen in der HAVING-Klausel sicher etwas Zeit und Geduld, und weitere Betrachtungen wie z. B. Einzelanalysen waren ebenfalls notwendig, um parallel die Auswirkungen geänderter Einstellungen zu prüfen. Hiermit ließ sich aber auch die Qualität der Ergebnisse fortlaufend immer weiter verbessern, bis letztlich manuelle Stichproben das Ergebnis nur noch positiv bestätigten.

Definition des Benutzerstatus

Um im weiteren Verlauf der Arbeit die Auswahl der identifizierten Studenten zu vereinfachen und damit auch den gesamten Prozess 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

```
mysql> CREATE TABLE moodle_data_students
       AS
       /*
       SQL-Statement zur Identifikation von Studenten
       */
```

Anschließend wurde in der Relation *moodle_data* nach dem Merkmal *userid* das neue Merkmal *userstatus* mit dem Default-Wert *other* eingefügt.

Einfügen des neuen Merkmals userstatus

```
mysql> ALTER TABLE moodle_data
       ADD COLUMN userstatus
       VARCHAR(10)
       DEFAULT 'other'
       AFTER userid;
```

In einem letzten Schritt wurde in allen Datensätzen der Tabelle *moodle_data*, die für das Merkmal *userid* einen zuvor identifizierten Studenten aufwiesen der *userstatus* auf *student* geändert.

Kennzeichnung von Studenten

```
mysql> UPDATE moodle_data SET userstatus = 'student'
       WHERE userid IN (SELECT userid FROM moodle_data_students);
```

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

```
mysql> SELECT DISTINCT userid FROM moodle_data
       WHERE userstatus = 'student';
+--------+
| userid |
+--------+
|      1 |
|     13 |
|     18 |
|    ... |
|    134 |
|    136 |
|    143 |
+--------+
69 rows in set (4,41 sec)
```

Überprüfung der Änderungen auf Richtigkeit

```
mysql> SELECT * FROM moodle_data
       WHERE (userstatus != 'student') AND
             (userid IN (SELECT userid FROM moodle_data_students));
Empty set (0,79 sec)
```

3.2 Zeitbezogene Untersuchungen

...

3.3 Aktivitätsbezogene Untersuchungen

...

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).

Cerf, V. (1969, Oktober). *ASCII format for network interchange* (Nr. 20). RFC 20. RFC Editor. Zugriff auf https://www.rfc-editor.org/info/rfc20 doi: 10.17487/RFC0020

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

Shafranovich, Y. (2005, Oktober). *Common Format and MIME Type for Comma-Separated Values (CSV) Files* (Nr. 4180). RFC 4180. RFC Editor. Zugriff auf https://www.rfc-editor.org/info/rfc4180 doi: 10.17487/RFC4180

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.

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_` ...