



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Bastian Probst

**Vergleich von Data Mining-Verfahren für ein auf Selbstauskunft
basierendes Empfehlungssystem**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Bastian Probst

**Vergleich von Data Mining-Verfahren für ein auf
Selbstauskunft basierendes Empfehlungssystem**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Computer Science
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Dr. Susanne Draheim

Eingereicht am: 25. August 2016

Bastian Probst

Thema der Arbeit

Vergleich von Data Mining-Verfahren für ein auf Selbstauskunft basierendes Empfehlungssystem

Stichworte

Empfehlungssystem, Filmbewertungen, Selbstauskunft

Kurzzusammenfassung

Durch den beständigen Wachstum an im Internet verfügbaren Daten wird eine Filterung immer erforderlicher. Eine Möglichkeit der Filterung ist die Verwendung eines Recommender System (RS). Neben der Filterung hat das RS den Vorteil, dass sich die Angebote der Seite den Wünschen und Bedürfnissen des Benutzers anpassen. Ein Problem für RS sind neue Benutzer oder Objekte, da für diese keine oder nur ungenaue Vorschläge generiert werden können. Eine Möglichkeit, dieses Problem zu lösen, ist die Selbstauskunft. Somit wird im Kontext dieser Arbeit auch die Frage diskutiert, ob es möglich ist, Empfehlungen mithilfe von Selbstauskunft zu generieren. Es wird gezeigt, dass die Bewertungen von Filmen mit - durch Selbstauskunft gewonnene - Kategoriebewertungen, vorhergesagt werden können.

Bastian Probst

Title of the paper

Comparison of data mining procedure for a recommendation system based on self disclosure

Keywords

Recommender System, movie rating, self disclosure

Abstract

With the constantly increasing available data in the internet a filtration becomes more and more necessary. A possible filtration is the use of a Recommender System (RS). Besides filtration the RS has the advantage that the offers of a webpage adjust to the user's requests and needs. A problem for the rs are new users or objects because offers for them cannot or can only be generated vaguely. A possibility to solve this problem is self disclosure. Therefore in the context of this thesis the question is also discussed whether it is possible to generate offers by using self disclosure. It is shown that the valuation of films can be predicted with the valuation of categories won by self information.

Inhaltsverzeichnis

Abkürzungsverzeichnis	vii
1. Einleitung	2
1.1. Motivation	2
1.2. Gliederung	3
2. Analyse	4
2.1. Recommender System	4
2.1.1. Anforderungen an Recommender Systems	7
2.1.2. Daten eines Recommender System	10
2.1.3. Methoden der Vorschlagsgenerierung	13
2.1.4. Vergleich der Methoden	16
2.1.5. Hybride Recommender Systems	19
2.1.6. Evaluierung	22
2.1.7. Aktuelle Forschung	23
2.1.8. Anforderungen und Probleme	23
2.2. Votingplattform	24
2.2.1. Registrierung	25
2.2.2. Login	25
2.2.3. Webfrontend	25
2.2.4. Votingsystem	26
2.2.5. Persistenz	26
2.3. Knowledge Discovery in Databases	26
2.3.1. Auswahl der Daten	27
2.3.2. Vorverarbeitung	27
2.3.3. Transformation	28
2.3.4. Data Mining	28
2.3.5. Interpretation	29
2.4. Fazit	29
3. Experiment	30
3.1. Ziel	30
3.2. Erstellung einer Votingplattform	30
3.2.1. Wordpress	31
3.2.2. Registrierung und Login	31
3.2.3. Web Frontend	32

3.2.4.	Votingsystem	32
3.2.5.	Persistenz	35
3.3.	Voting	36
3.3.1.	Ziel des ersten Votings	36
3.3.2.	Akquirierung der Teilnehmer für das erste Voting	36
3.3.3.	Durchführung des ersten Votings	37
3.3.4.	Ziel des zweiten Votings	38
3.3.5.	Akquirierung der Teilnehmer für das zweite Voting	39
3.3.6.	Durchführung des zweiten Votings	39
3.4.	Das Recommender System	39
3.4.1.	Auswahl der Daten	39
3.4.2.	Vorverarbeitung	40
3.4.3.	Transformation	41
3.4.4.	Datamining	42
3.5.	Fazit	45
4.	Auswertung	46
4.1.	Auswertung der Informationen über die Teilnehmer	46
4.1.1.	Teilnehmer des ersten Experiments	46
4.1.2.	Teilnehmer des zweiten Experiments	49
4.2.	Auswertung der Kategoriebewertungen	51
4.3.	Auswertung der Filmbewertungen	52
4.3.1.	Filmbewertungen des ersten Experiments	52
4.3.2.	Filmbewertungen des zweiten Experiments	55
4.3.3.	Filme mehrerer Kategorien	57
4.4.	Auswertung der Methoden	58
4.4.1.	Auswertung des ersten Ansatzes	58
4.4.2.	Auswertung des zweiten Ansatzes	60
4.4.3.	Auswertung von Collaborative Filtering mit k-nearest	61
4.4.4.	Vergleich mehrerer Methoden	63
4.5.	Auswertung des gesamten Experiments	64
5.	Fazit	66
5.1.	Zusammenfassung	66
5.2.	Ausblick	66
A.	Anhang	68
A.1.	Fragen zur Person	68
A.2.	Bewertungen der Kategorie	69
A.3.	Filme des ersten Experiments	70
A.4.	Filme des zweiten Experiments	72
A.5.	Filmbewertungen des zweiten Experiments nach Herkunftsland	73
A.6.	Informationstext des ersten Votings	74

A.7. Informationstext des zweiten Votings 76

Abkürzungsverzeichnis

RS Recommender System

RSs Recommender Systems

CF Collaborative Filtering

EPG Electronic Programming Guide

IMDb Internet Movie Database

ID Kennung

KDD Knowledge Discovery in Databases

Listings

3.1. Erweiterung von Yop Poll	33
3.2. Methode: hasPollsLeft	34
3.3. Methode zum Überprüfen, ob ein Benutzer noch abstimmen darf	34
3.4. Methode zur Auswahl der Benutzer	39
3.5. Methode zum Berechnen der Distanz zwischen zwei Benutzern	41
3.6. Methode zum Berechnen des Ratings anhand der Kategoriebewertungen	44

Danksagung

Ohne Teilnehmer wären die Experimente im Rahmen dieser Arbeit nicht möglich gewesen. Daher möchte ich Michael Müller, Marcell Klepper, Hendrik Seemann, Leon Arndt, Sascha Knorr, Cornelius Wilkening, Frederike Endejan, Florian Schöttle, Lukas Grundmann, Bente Annika Ollmann, Christoph Petersen, Christopher Dargel, Sven Thölke, Igor Arkhipov, Flemming Garken, Andreas Dymek, Tim Alcantara Ortega, Carolin Raeggel, Sebastian Konrad, Felix Uelsmann, Dustin Leitold, Paula Lizakowski, Sascha Kluth, Hanna Levitskiy und allen Teilnehmern danken, welche nicht namentlich genannt werden wollten.

1. Einleitung

„If the Starbucks secret is a smile when you get your latte... ours is that the Web site adapts to the individual's taste.“

Reed Hastings

Dieses Zitat von Reed Hastings, dem Gründer von Netflix, spiegelt die Relevanz von Recommender System (RS) für Onlinedienste wieder. Er stellt die Fähigkeit des Systems, sich dem Geschmack des Benutzers anpassen zu können, sogar als Erfolgsgeheimnis dar. Aber auch für andere Onlinedienste, deren Geschäftsmodell der Onlineverkauf von Objekten oder Diensten ist, sind RS ein wesentlicher Bestandteil des Erfolgs. Immer mehr Daten werden im Internet verfügbar. Somit wird für die Suche nach relevanten Daten eine gezielte Filterung erforderlich. Eine Möglichkeit der Filterung ist Personalisierung.

1.1. Motivation

Im Kontext dieser Arbeit geht es um die Frage:

Ist es möglich, mithilfe von Selbstauskunft Empfehlungen zu generieren?

Um diese Frage beantworten zu können, wurden zwei Experimente durchgeführt: es wurde ein RS entwickelt, und es wurden die Ergebnisse von diesem RS evaluiert.

Neue Benutzer und neue Filme stellen ein RS vor eine große Herausforderung. So ist es zum Beispiel schwer, die Vorlieben von einem Benutzer zu errechnen, der noch keine Filme bewertet hat. Ebenfalls ist es ein Problem, eine Bewertung für einen Film zu berechnen, der bisher noch nicht bewertet worden ist. Sowohl das Problem der neuen Benutzer als auch das der neuen Filme werden als Kaltstartproblem bezeichnet.

Viele der Methoden von RS leiden unter dem Kaltstartproblem. Im Kontext dieser Arbeit wird eine Möglichkeit analysiert, das Kaltstartproblem zu umgehen: die Verwendung von Selbstauskunft. Somit ist das Ziel dieser Arbeit, nachzuweisen, dass das Kaltstartproblem mithilfe von Selbstauskunft gelöst werden kann.

1.2. Gliederung

In dieser Masterarbeit wird die Frage diskutiert, ob es möglich ist, Empfehlungen mithilfe von Selbstauskunft zu generieren. Hierfür wurden zwei Experimente durchgeführt, ein Empfehlungssystem erstellt und dessen Ergebnisse evaluiert. Die Arbeit ist neben dieser Einleitung in vier weitere Kapitel gegliedert:

In Kapitel 2 werden zunächst Recommender Systems (RSs) allgemein beschrieben. Es wird ausgeführt, welche Anforderungen an RSs bestehen. Ebenfalls wird betrachtet, welche Daten und Methoden für die Generierung von Empfehlungen verwendet werden. Es werden unterschiedliche Metriken untersucht, wie RS evaluiert werden können. Anschließend wird dargestellt, wie eine Votingplattform erstellt wird, gefolgt von einer Darstellung der einzelnen Schritte des Knowledge Discovery in Databases (KDD) Prozesses.

Im 3. Kapitel wird das Experiment genauer ausgeführt. Nach einer Darstellung des Ziels wird die Umsetzung der Votingplattform beschrieben. Danach werden die Durchführungen der beiden Experimente betrachtet. Am Ende des Kapitels wird die Implementierung des RS vorgestellt.

Die Auswertung der Experimente sowie die Ergebnisse des RS werden in Kapitel 4 betrachtet. Nach einer Analyse der Teilnehmer folgt eine Analyse der Kategoriebewertungen. Daraufhin werden die Filmbewertungen der beiden Experimente genauer analysiert. Abschließend werden die Ergebnisse der unterschiedlichen Methoden ausgewertet.

Im letzten Kapitel (Kapitel 5) werden die wesentlichen Aspekte zusammengefasst. Abschließend wird ein Ausblick über eine mögliche Fortsetzung dieser Arbeit und über die Zukunft von RSs allgemein gegeben.

2. Analyse

Am Anfang dieses Kapitels werden **RSs** näher betrachtet. Nach einer Ausführung der Anforderungen an ein **RS** werden die Daten und Methoden dargestellt, welche für die Generierung von Empfehlungen verwendet werden. Im Anschluss werden die unterschiedlichen Methoden verglichen. Nach einer Betrachtung von hybriden **RSs** werden Evaluierungsmöglichkeiten für die Qualität eines **RS** vorgestellt. Abschließend werden die aktuelle Forschung, Anforderungen und Probleme von **RS** aufgeführt.

In Kapitel 2.2 werden die Komponenten eines Votingplattform und in Kapitel 2.3 die Komponenten des **KDD** Prozesses beschrieben. Die Durchführung des Experiments folgt in Kapitel 3

2.1. Recommender System

RSs sind Methoden und Systeme, welche personalisierte Vorschläge von Objekten generieren, die für einen bestimmten Benutzer von Interesse oder Nutzen sind **Sarwar u. a. (2001)** **Herlocker u. a. (2004)** **Koren (2008)** **Deshpande und Karypis (2004)**.

Recommender Systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user **Ricci u. a. (2015)**.

Vorschläge anderer Menschen werden verwendet, wenn nicht genug persönliche Erfahrung über die Alternativen vorhanden ist. **RSs** unterstützen und bereichern diesen Prozess **Resnick und Varian (1997)**.

Unter anderem werden Vorschläge für folgende Fragen generiert:

- Was kaufe ich? **Leino und Rähä (2007)** **Jannach u. a. (2015)** **Zhao u. a. (2015)**
- Welche Nachrichten lese ich? **Das u. a. (2007)** **Resnick u. a. (1994)**
- Welche Musik höre ich? **Jannach u. a. (2015)** **Kaminskas u. a. (2013)**

- Oder wie im Kontext dieser Arbeit: Welchen Film schaue ich? [Gantner u. a. \(2010\)](#) [Said u. a. \(2010\)](#) [Probst \(2015b\)](#)

RSs fokussieren sich auf eine bestimmte Art von Objekten (zum Beispiel Filme). Das Design, das Interface und der Algorithmus der Vorschlagsgenerierung werden für diese spezielle Aufgabe angepasst. Ein Beispiel ist Netflix mit einer Fokussierung auf Serien und Filme. Bei Netflix werden den Nutzern personalisierte Vorschläge für Filme und Serien gemacht. Hierfür werden das Nutzungsverhalten des Benutzers und das Verhalten ähnlicher Benutzer betrachtet. Anhand dieser Daten wird berechnet, welche Videos den Benutzer am meisten interessieren [Gomez-Uribe und Hunt \(2015\)](#).

Neben den personalisierten **RSs** existieren auch nicht personalisierte **RSs**. Die Generierung von Vorschlägen bei nicht personalisierten **RSs** ist bedeutend einfacher. In der Regel werden Objekte vorgeschlagen, welche am häufigsten betrachtet oder am besten bewertet wurden. Diese sind im Bereich der Forschung von peripherem Interesse.

Eine Möglichkeit zum Erzeugen personalisierter Vorschläge ist ein Vergleich der Eigenschaften eines Nutzers mit denen der Objekte. Bei dieser Methode wird ein Benutzerprofil anhand von explizitem (abgegebene Bewertungen) und implizitem (Anschauen eines Filmes) Feedback erstellt. Die Eigenschaften des Benutzers werden mit denen der Objekte verglichen. Je ähnlicher sich Objekt und Benutzer sind, desto höher wird das entsprechende Objekt bewertet. Schließlich werden dem Benutzer die am höchsten bewerteten Objekte vorgeschlagen [Shi u. a. \(2010\)](#).

Eine andere Möglichkeit für die Generierung von Vorhersagen ist der Vergleich mit ähnlichen Benutzern. Das System vergleicht den aktuellen Benutzer mit anderen Nutzern. Zum Beispiel anhand der Historie gelesener Nachrichten. Das System schlägt dem aktuellen Benutzer die Objekte vor, welche ähnliche Nutzer ausgewählt haben. Diese Methode wird als Collaborative Filtering (**CF**) bezeichnet.

Im Gegensatz zu der Betrachtung der ähnlichen Benutzer wird die Betrachtung, welche Objekte häufig zusammen gekauft werden, als „item based **CF**“ bezeichnet („Nutzer, die diesen Artikel kauften, kauften auch...“).

RSs kommen in unterschiedlichen Bereichen zum Einsatz. Diese sind unter anderem:

- **Unterhaltung** (Filme, Musik, Spiele): Im Bereich der Unterhaltung ist unter anderem Netflix zu nennen. Netflix ist ein Streamingdienst, dessen Kernprodukt ein **RS** ist. Netflix hat 65 Millionen Mitglieder, welche mehr als 100 Millionen Stunden Videomaterial konsumieren (Dezember 2015). Die Qualität des **RS** ist ein essentieller Bestandteil. Gute Vorschläge verbessern das Nutzererlebnis und erhöhen somit die Wahrscheinlichkeit einer kontinuierlichen Nutzung des Dienstes [Gomez-Uribe und Hunt \(2015\)](#) [Berry u. a. \(2010\)](#) [Amatriain \(2013\)](#).

- **Inhalte** (Nachrichten, Dokumente, Webseiten, Email-Filter): Wie zum Beispiel Google News [Das u. a. \(2007\)](#) [Joshi und Gatica-Perez \(2006\)](#) [Leporini \(2011\)](#).
- **E-Commerce** (Produkte): Hier ist unter anderem Amazon ein bekanntes Beispiel [Leino und Rähä \(2007\)](#) [Palankar u. a. \(2008\)](#).
- **Dienste** (Reisen, Experten, Mietwohnungen, Dating-sites): Ein Beispiel hierfür ist die Seite Parship [Wikipedia \(2016a\)](#).
- **Soziales** (Freunde in social network, Inhalte von social media): Beispiele sind: LinkedIn und Facebook [Borthakur \(2013\)](#) [Camara u. a. \(2012\)](#) [Skeels und Grudin \(2009\)](#) [Unkelos-Shpigel u. a. \(2015\)](#).

Die Forschung im Bereich der **RSs** ist noch relativ neu im Vergleich zu anderen Bereichen der Informatik (zum Beispiel Datenbanken oder Texterkennung). Das Interesse an **RSs** ist in den letzten Jahren deutlich angestiegen, wie die folgenden Fakten belegen:

1. **RSs** spielen eine bedeutende Rolle in hoch bewerteten Seiten, wie YouTube, Netflix, Spotify, LinkedIn, Facebook, Tripadvisor, Last.fm, and IMDb. Nicht von ungefähr hat Netflix zum Beispiel einen Preis von einer Million Dollar an die Gruppe verliehen, welche die Performance ihres **RS** signifikant verbessern konnte.
2. Es finden Konferenzen und Workshops mit dem Thema statt. Unter anderem die jährlich stattfindende Association of Computing Machinery's (ACM) Conference Series on Recommender Systems (RecSys), welche 2007 eingeführt wurde. Diese Konferenz behandelt die aktuellen Forschungen und Anwendungen im Bereich der **RSs**. Aber auch traditionellere Konferenzen haben mittlerweile Workshops und Sessions, welche dieses Thema behandeln. Zu diesen gehören unter anderem: ACM's Special Interest Group on Information Retrieval (SIGIR); User Modeling, Adaptation and Personalization (UMAP); Intelligent User Interfaces (IUI); World Wide Web (WWW); und ACM's Special Interest Group on Management Of Data (SIGMOD).
3. Es werden immer mehr Bücher und Artikel veröffentlicht, welche **RSs** als Thema haben.
4. Viele akademischen Zeitschriften haben Ausgaben, die sich des Themas der Forschung und Entwicklung im Feld der **RSs** widmen. Einige von diesen sind: AI Communications (2008); IEEE Intelligent Systems (2007); International Journal of Electronic Commerce (2006); International Journal of Computer Science and Applications (2006); ACM Transactions on Computer Human Interaction (2005); ACM Transactions on Information

Systems (2004); User Modeling and User-Adapted Interaction (2014, 2012); ACM Transactions on Interactive Intelligent Systems (2013); and ACM Transactions on Intelligent Systems and Technology (2015).

Provider und Nutzer haben unterschiedliche Anforderungen an die in diesem Kapitel beschriebenen **RSs**. Diese Anforderungen werden im nächsten Kapitel genauer betrachtet.

2.1.1. Anforderungen an Recommender Systems

Nach der allgemeinen Einführung im vorigen Kapitel, folgt hier eine Betrachtung Anforderungen an ein **RS**.

Der Zweck eines **RS** kann aus zwei Sichten betrachtet werden: die Sicht der Provider, die **RSs** einsetzen, und die der Benutzer. Zunächst werden die Anforderungen an **RSs** dargestellt, die von den Providern gestellt werden.

Die Vorteile aus Sicht der Provider

Nach **Ricci u. a. (2015)** existieren folgende Vorteile für einen Provider bei der Nutzung eines **RS**.

- **Mehr Verkäufe:** Diese bedeutende Funktion stellt die Differenz zwischen den Verkaufsabschlüssen mit und ohne **RS** dar. Dieses Ziel wird erreicht, da dem Nutzer Objekte von Interesse angeboten werden. Und es führt dazu, dass der Nutzer mehr kauft, da die Angebote seinen Wünschen und Interessen entsprechen. Aber auch nicht-kommerzielle Anbieter haben ein Interesse, dass der Nutzer mehr Objekte auswählt. Auch wenn dadurch keine direkten Kosten für den Benutzer entstehen. Bei einer Nachrichtenseite zum Beispiel besteht ein Interesse daran, dass der Nutzer sich durch so viele Nachrichten wie möglich klickt. Somit besteht für alle Provider ein Interesse daran, dass so viele Vorschläge wie möglich akzeptiert werden.
- **Breiteres Spektrum an Verkäufen:** Eine weitere Funktion ist eine Darbietung von Optionen, welche ohne präzise Vorschläge schwer zu finden wären. Ein Onlineshop hat ein Interesse, auch weniger populäre Produkte anzubieten. Ohne ein **RS** birgt dies die Gefahr, dem Nutzer Produkte vorzuschlagen, welche nicht seinen Interessen entsprechen.
- **Nutzerzufriedenheit steigern:** Ein gut gestaltetes **RS** bewirkt, dass der Benutzer ein positives Erlebnis beim Benutzen der Webseite oder der Anwendung hat. Interessante und relevante Vorschläge in Verbindung mit einem gut gestalteten Benutzerinterface führen dazu, dass die Nutzung des Systems dem Anwender Freude bereitet. Dies führt wiederum zu einer Steigerung der Verwendung des Systems.

- **Steigern der Benutzerstreue:** Ein Benutzer bleibt einer Seite treu, wenn diese ihn als einen geschätzten Besucher erkennt und behandelt. Dies ist eine Standardfunktion von **RSs**. Es werden Daten über den Benutzer gesammelt und zu seinem Vorteil verwendet. Je länger der Benutzer mit der Seite interagiert, desto genauer wird sein Benutzermodell. Dies hat zur Folge, dass die generierten Vorschläge präziser werden.
- **Die Wünsche des Nutzers erkennen:** Eine weitere wichtige Funktion eines **RS** ist die Beschreibung der Benutzerpräferenzen. Diese können entweder explizit (Bewerten eines Filmes) oder implizit (Anschauen eines Filmes) gesammelt werden. Diese Informationen können für andere Zwecke weiterverwendet werden. Zum Beispiel für zielgerichtete Werbung oder für eine Analyse, ob ein bestimmter Film das Interesse der Benutzer wecken wird oder nicht.

Somit wird deutlich, dass aus Providersicht eine Vielzahl an Gründen dafür spricht, ein **RS** zu verwenden. Aber auch aus Sicht der Anwender gibt es Vorteile bei der Interaktion mit einem **RS**.

Vorteile aus Sicht der Anwender

Nach **Herlocker u. a. (2004)** hat ein **RS** folgende elf Aufgaben, welche für den Benutzer von Vorteil sind:

- **Finden einiger guten Objekte:** Dem Benutzer wird eine geordnete Liste von Objekten zur Verfügung gestellt, welche für ihn von Interesse sein könnten. Je wahrscheinlicher es ist, dass ein Benutzer ein Interesse an einem bestimmten Objekt hat, desto weiter oben erscheint dieses in der Liste. Dies ist die Hauptaufgabe vieler modernen **RS**.
- **Finden aller guter Objekte:** Dem Benutzer werden alle Objekte vorgeschlagen, welche die Bedürfnisse des Benutzers befriedigen. Dies gilt vor allem, wenn die Anzahl der Elemente relativ klein ist oder es sich um kritische Anwendungen handelt, wie Medizin- oder Finanzapplikationen. Bei diesen Systemen können dem Nutzer die Anzeige des Rankings oder weitere Informationen von Vorteil sein.
- **Anmerkungen im Kontext:** Bei einer Liste von Objekten, welche zu den Präferenzen des Benutzers passen, kann das System zusätzliche Informationen hinzufügen. So kann zum Beispiel ein Electronic Programming Guide (**EPG**) anzeigen, welche Sendungen es Wert sind, angeschaut zu werden.

- **Vorschlagen einer Sequenz:** Anstatt nur einzelne Objekte vorzuschlagen, kann eine Sequenz von Objekten vorgeschlagen werden. Typische Beispiele sind TV-Serien oder eine Zusammenstellung von Musikstücken.
- **Vorschlagen einer Gruppe:** Einem Benutzer kann eine Gruppe von Objekten vorgeschlagen werden, welche gut zusammenpassen. Dies wäre unter anderem ein Reiseplan mit mehreren unterschiedlichen Zielen, der von dem Benutzer als eine einzige Reise gebucht werden kann.
- **Durchblättern:** Wenn ein Benutzer einen Katalog ohne eine bestimmte Absicht durchblättert, kann das RS ihn unterstützen, ein für ihn interessantes Objekt leichter zu finden. Ein Beispiel hierfür sind die Videovorschläge von youtube.com.
- **Finden zuverlässiger Vorschläge:** Benutzer, welche RSs nicht vertrauen, können mit diesen herumspielen, um zu testen wie gut die generierten Vorschläge sind. Daher bieten einige RSs spezielle Funktionen zum Testen der generierten Vorschläge an.
- **Verbessern des Benutzerprofils:** Der Benutzer kann dem System mitteilen, was er mag und was er nicht mag. Dies ist eine fundamentale Voraussetzung, um personalisierte Vorschläge zu generieren. Ohne eine genaue Kenntnis über die Präferenzen des Benutzers kann das System nur Vorschläge generieren, die auf einen durchschnittlichen Benutzer zugeschnitten sind.
- **Sich mitteilen:** Einige Benutzer haben kein Interesse an personalisierten Vorschlägen. Ihnen ist es wichtig, einen Beitrag zu leisten, indem sie ihre Meinungen und Überzeugungen mitteilen können. Die Befriedigung dieser Tätigkeit führt zu einer häufigeren Interaktion mit dem System.
- **Anderen helfen:** Einigen Benutzern bereitet es Freude, wenn sie anderen mit ihren Kritiken und Bewertungen weiterhelfen können. Ihre Motivation dabei ist, die Gemeinschaft auf diese Weise zu bereichern. So sind zum Beispiel die Beurteilungen eines Artikels nach einem Kauf nicht mehr für den Käufer selbst von Vorteil. Das einzige Ziel ist es, anderen Benutzern bei der Kaufentscheidung weiterzuhelfen.
- **Andere beeinflussen:** Anstatt zu Helfen, kann ein Benutzer auch die Absicht haben, andere beeinflussen zu wollen. Zum Beispiel ein bestimmtes Produkt zu kaufen oder nicht zu kaufen. Dies kann eine böartige Intention haben. Ein RS sollte böartiges Verhalten erkennen und entsprechend die negative Beeinflussung bestimmter Objekte verhindern.

Diese Punkte verdeutlichen, dass für unterschiedliche Provider oder Benutzer auch die Rolle eines **RS** sehr unterschiedlich sein kann. Im nächsten Kapitel wird betrachtet, welche Daten von einem **RS** für die Vorhersage verwendet werden.

2.1.2. Daten eines Recommender System

RSs sammeln viele unterschiedliche Daten, um Vorschläge generieren zu können. Diese Daten betreffen hauptsächlich die Objekte, welche vorgeschlagen werden sollen. Oder es sammelt Daten über die Benutzer, für die Vorschläge generiert werden sollen. Ob alle gesammelten Daten von den Algorithmen verwendet werden können, hängt von der Art des **RS** und den verwendeten Methoden ab.

Im Wesentlichen werden folgende Datentypen für die Vorhersage verwendet:

- **Objekte:** Objekte sind die Dinge, welche von dem **RS** vorgeschlagen werden. Objekte werden durch ihre Eigenschaften und ihren Wert charakterisiert. Ein Objekt hat einen positiven Wert, wenn es nützlich für den Benutzer ist. Einen negativen Wert hat ein Objekt, welches für den Nutzer nicht von Vorteil ist. Die Kosten eines Objekts sind dabei der Aufwand bei der Suche und die direkten Kosten, wenn das Produkt tatsächlich gekauft wird.

So muss zum Beispiel ein Entwickler eines **RS** für Filme die Eigenschaften der Filme berücksichtigen. Unter anderem das Genre, den Regisseur, die Filmlänge und die Sprache. Auch wenn für das Anschauen der Filme keine Kosten für den Benutzer entstehen, muss der Entwickler des Systems immer berücksichtigen, dass der Benutzer einen Aufwand bei der Suche nach einem Film hat, welchen er schauen möchte. Wird also ein Film vorgeschlagen, welchen der Benutzer nicht sehen möchte, hat dieser Vorschlag einen negativen Wert.

Objekte mit niedriger Komplexität und Wert sind: Nachrichten, Bücher, CDs, Internetseiten oder wie im Kontext dieser Arbeit: Filme. Komplexere Objekte sind: Computer, Handys oder Autos. Die höchste Komplexität und den höchsten Wert haben: Versicherungspolice, Reisen, Finanzinvestments und Berufe.

In Abhängigkeit von den verwendeten Algorithmen unterscheidet sich die Anzahl der erforderlichen Eigenschaften und Merkmale der Objekte. Bei Filmen können unter anderem der Regisseur, die Schauspieler, der Drehort oder das Genre verwendet werden. Die einfachste Repräsentation eines Objektes besteht aus einer einzigen ID. Komplexere Darstellungen sind Mengen von Eigenschaften bis hin zu Konzepten ontologischer Einordnung. Im Kontext dieser Arbeit werden unterschiedliche Methoden mit divergie-

render Nutzung von Attributen verwendet. Diese reichen von einfacher Verwendung des Genres bis hin zu deutlich komplexeren Beschreibungen der Filme.

- **Benutzer:** Wie bereits weiter oben beschrieben, haben Benutzer eines **RS** viele unterschiedliche Ziele und Charakteristiken. Viele divergente Informationen werden von einem **RS** zur Kalkulation von Vorschlägen verwendet. In Abhängigkeit von dem verwendeten Algorithmus werden diese unterschiedlich strukturiert und selektiert. Ein Algorithmus ist das kollaborative Filtern. Bei diesem werden die Benutzer als einfache Listen von Ratings modelliert. Bei einem demographischen **RS** wird das Benutzerprofil angereichert um Attribute, wie Alter, Geschlecht, Bildungsstand oder Sprache. Das Benutzerprofil verfolgt das Verhalten des Benutzers und passt die Vorschläge den gewonnenen Erkenntnissen entsprechend an. Es sind viele unterschiedliche Ansätze für Benutzermodelle entwickelt und verwendet worden. So kann ein **RS** als ein System betrachtet werden, welches Benutzerprofile erstellt und zum Generieren von Vorschlägen verwendet wird. Eine ausführliche Beschreibung von Benutzermodellen findet sich weiter unten.
- **Transaktion:** Als Transaktion wird im Kontext dieser Masterarbeit eine aufgezeichnete Interaktion zwischen dem Benutzer und dem **RS** verstanden. Somit ist das Bewerten einer Kategorie auf der Versuchsseite eine Transaktion. Transaktionen sind Daten, welche mit einem Log vergleichbar sind. Sie liefern Informationen, die von den Algorithmen des **RS** für die Vorschläge verwendet werden können. Die Transaktionen geben ein implizites (der Benutzer schaut sich einen Film an) oder explizites Feedback (der Benutzer bewertet einen Film). Objekte können auf folgende Arten durch den Benutzer bewertet werden:
 - **numerische Bewertungen,**
 - **Ordinalbewertungen,**
 - **binäre Bewertungen** oder
 - **unary ratings.**

Eine weitere Möglichkeit ist das Kennzeichnen von Objekten. Zum Beispiel bewerten bei Movielens (<http://movielens.umn.edu>) die Benutzer die Filme mit Kennzeichnungen, wie „zu lang“ oder „bewegend“.

Bei Transaktionen mit implizitem Feedback hat das System das Ziel, die Meinung der Benutzer anhand der Aktionen des Benutzers zu errechnen. Das Betrachten eines kompletten Films deutet zum Beispiel auf eine positive Meinung des Benutzers über diesen Film hin.

Benutzermodelle

Benutzermodelle sind für **RSs** von besonderer Bedeutung. In diesem Kapitel werden deren Eigenschaften genauer betrachtet.

Eigenschaften eines Benutzers

Nach **Mejía u. a. (2012)** können die Eigenschaften der Benutzer in fünf Kategorien unterteilt werden. Diese sind:

- physische Eigenschaften,
- kognitive Eigenschaften,
- demographische Eigenschaften,
- Erfahrung und
- psychische Eigenschaften.

Bei den **physischen Eigenschaften** werden die körperlichen Merkmale der Benutzer betrachtet, welche bei der Interaktion mit dem System beteiligt sind. In extremer Betrachtung kann hier aufgeführt werden, ob erforderliche Körperteile vorhanden sind oder diese nur bedingt für die Interaktion mit dem System verwendet werden können. Solche Benutzereigenschaften analysieren Systeme, die für ältere oder beeinträchtigte Benutzer entwickelt werden.

RSs haben häufig ein Webinterface, über welches der Benutzer mit dem System interagiert. Diese können physische Eigenschaften betrachten, wie:

- Tippgeschwindigkeit oder
- Lesegeschwindigkeit

Zu den **kognitiven Eigenschaften** zählen unter anderem Intelligenz, Erinnerungsvermögen und Konzentration. Diese sind für interaktive Systeme sehr bedeutend. Wenn erkannt wird, dass der Benutzer unkonzentriert ist, sollten diesem keine komplexen Aufgaben gestellt werden. Auch die Hilfestellungen eines Systems sollten an die Intelligenz und das Erinnerungsvermögen des Benutzers angepasst werden. Dieser sollte weder durch zu viel Hilfe belästigt, noch durch zu wenig Hilfe überfordert werden.

Für ein Vorhersagesystem sind diese Eigenschaften irrelevant, da sie nur indirekt mit den Präferenzen des Benutzers in Verbindung gebracht werden können. Somit ist eine direkte Analyse der Präferenzen sinnvoll. Diese ist unter anderem in den Kontext der Zeit zu setzen.

Eigenschaften wie Alter, Geschlecht und Bildung zählen zu den **demographischen Eigenschaften**. Diese sind zum Einen im Bereich der Interaktion von Interesse. Hier ist davon auszugehen, dass ältere Menschen mehr Schwierigkeiten bei der Bedienung eines Programms haben als jüngere. Bei sehr jungen Benutzern ist zu erwarten, dass diese Bilder besser als Text verstehen.

Für die Generierung von Filmvorschlägen haben diese Informationen ebenfalls eine hohe Relevanz. Benutzer unterschiedlicher Altersgruppen haben unterschiedliche Filminteressen. So werden Zeichentrickfilme eher von jüngeren Benutzern angesehen. Auch das Geschlecht kann für die Generierung von Filmvorschlägen verwendet werden. Aktionfilme werden eher von männlichen Benutzern präferiert. Liebesfilme finden eine höhere Akzeptanz bei weiblichen Zuschauern.

Die **Erfahrung** ist ausschließlich für die Interaktion mit einem System interessant. Wenn betrachtet wird, inwieweit ein Benutzer Erfahrungen im Umgang mit Computern hat, kann erkannt werden, ob deutliche Hilfestellung für einfache Aktionen erforderlich sind.

Beim Generieren von Filmvorschlägen spielt die Erfahrung im Umgang mit Computern keine Rolle.

Zur Kategorie der **psychischen Eigenschaften** zählt unter anderem die Motivation. Im Kontext eines Vorschlagsystems für Filme ist davon auszugehen, dass der Benutzer eine gewisse Motivation hat, einen Film zu schauen.

Nach der Betrachtung der Daten eines **RS** im Allgemeinen und den Benutzermodellen im Speziellen, werden im nächsten Kapitel die Methoden der Vorschlagsgenerierung analysiert.

2.1.3. Methoden der Vorschlagsgenerierung

Die Hauptfunktion eines **RS** ist das Identifizieren nützlicher Objekte für den Benutzer. Zum Erfüllen dieser Aufgabe müssen die Objekte bestimmt werden, die es wert sind, vorgeschlagen zu werden. Somit muss ein **RS** die Nützlichkeit eines Objektes bestimmen oder wenigstens vergleichen können. Nicht alle Algorithmen berechnen explizit den Nutzen eines Objektes. Jedoch kann diese Verallgemeinerung verwendet werden, um die allgemeine Rolle eines **RS** zu beschreiben.

Die tatsächliche Nützlichkeit eines Objektes wird mit der Funktion: $R(u, i)$ beschrieben. Dies entspricht in der Regel der tatsächlichen Bewertung eines Objektes durch den Benutzer. Die Hauptaufgabe eines **RS** ist die Berechnung dieses Wertes. Ein **RS** berechnet $\hat{R}(u, i)$. Wobei

\hat{R} der angenäherte Wert der tatsächlichen Bewertung R ist. Das System berechnet für einen aktiven Benutzer u und eine Menge an Objekten den Wert von \hat{R} .

$$\hat{R}(u, i_1), \dots, \hat{R}(u, i_N) \quad (2.1)$$

Das System schlägt dem Benutzer die Objekte i_{j1}, \dots, i_{jK} ($K \leq N$) vor, welche den höchsten berechneten Nutzen haben. Typischer Weise ist die Anzahl der Objekte (N) bedeutend größer als die der vorgeschlagenen Objekte (K). Somit fungiert das **RS** als Filter.

Nicht alle **RSs** bestimmen genau den Nutzen eines Objekts für den Benutzer. Es werden eher Heuristiken angewendet, um zu bestimmen, welche Objekte für den Benutzer von Nutzen sind. Dies ist typisch für wissensbasierte **RSs**. Es werden spezielle Algorithmen verwendet, welche unterschiedliches Wissen über die Objekte, Benutzer und Nutzenfunktion verwenden. Ist die Nutzenfunktion zum Beispiel boolesch, wird lediglich bestimmt, ob ein Objekt von Nutzen ist oder nicht.

Die Nützlichkeit eines Objektes ist manchmal vom Kontext abhängig: zum Beispiel die Expertise in der entsprechenden Domäne (Experte im Vergleich zu einem Anfänger im Bereich der Digitalkameras). Die Nützlichkeit kann auch von dem Ort abhängen. Bei Vorschlägen für Restaurants ist zu erwarten, dass Restaurants in der Nähe des Benutzers einen hohen Nutzen haben. Ein Restaurant, welches weit entfernt ist, hat eher einen kleinen oder keinen Nutzen. Die Vorschläge anhand des Kontextes anzupassen, erschwert eine korrekte Bestimmung der richtigen Vorschläge.

Burke (2002) charakterisierte fünf unterschiedliche Typen von **RSs** und erweiterte diese um hybride **RSs**, welche Kombinationen aus zwei oder mehr der anderen Typen sind.

- **Kollaborative:** Kollaboratives Filtern ist vermutlich die bekannteste, bewährteste und am weitesten verbreitete Methode. Beim kollaborativen Filtern werden die Bewertungen von Benutzern für Objekte gespeichert. Benutzer können anhand der von ihnen abgegebenen Bewertungen verglichen werden. Vorschläge werden mithilfe des Vergleichs mit anderen Benutzern generiert. Ein Benutzerprofil eines **RS** mit kollaborativen Filtern besteht typischerweise aus einem Vektor von Objekten und deren Ratings. Eine Interaktion des Benutzers mit dem System führt zu einer Aktualisierung des Benutzerprofils. Einige Systeme bewerten aktuellere Ratings höher als ältere. Auf diese Weise reagiert das System, wenn sich die Interessen des Benutzers ändern.

Beim kollaborativen Filtern ist keine maschinenlesbare Darstellung der Objekte erforderlich. Dies ist die größte Stärke des kollaborativen Filterns, da es auch ohne weiteren Aufwand für komplexe Elemente, wie Filme oder Musik, verwendet werden kann.

- **Demographische:** Ein demographisches **RS** kategorisiert die Benutzer anhand ihrer Eigenschaften und verwendet diese Informationen, um Vorschläge zu generieren. Die Benutzer werden in manuell erstellte Stereotypen oder demographische Gruppen eingeordnet. Somit werden Vorschläge anhand von Altersgruppen oder der Sprache angepasst. Im Kontext dieser Arbeit werden über die Nutzer demographische Daten gesammelt, wie Alter, Geschlecht und Konsumverhalten. Der Vorteil von demographischen **RSs** ist, dass keine Historie über abgegebene Bewertungen benötigt wird. Nach **Ricci u. a. (2015)** existiert zwar eine hohe Popularität der **RSs** in der Marketingliteratur, doch hat es wenig geeignete Forschung zu demographischen **RSs** gegeben.
- **Inhaltsbasierte:** In einem inhaltsbasierten **RS** werden die Objekte durch ihre Eigenschaften definiert. Das Benutzerprofil wird anhand der Ratings vom Benutzer und den Eigenschaften der bewerteten Objekten erlernt. Die Art des Benutzerprofils hängt von der Lernmethode ab. So kann ein System zum Beispiel lernen, wenn ein Benutzer Filme vom Genre *Action* gut bewertet, dass dieser ein Interesse an Actionfilmen hat. Im Kontext dieser Arbeit werden unter anderem die Genre der Internet Movie Database (**IMDb**) verwendet. Wie beim kollaborativen Filtern wird auch beim inhaltsbasierten **RS** das Benutzermodell durch die Interaktion mit dem System aktualisiert.
- **Nutzen-basierte:** Ein nutzen-basiertes oder wissensbasiertes **RS** erstellt kein langfristiges Benutzermodell. Die Vorschläge eines nutzen-basierten **RS** haben das Ziel, die akuten Bedürfnisse des Benutzers zu befriedigen. Somit berechnet ein nutzen-basiertes **RS** den Nutzen der Objekte für den Benutzer. Anhand dieser Berechnung werden dann die Empfehlungen generiert.
Das Benutzermodell eines nutzen-basierten **RS** besteht aus der Nutzenfunktion für den jeweiligen Benutzer. Somit ist das Hauptproblem die Erstellung der Nutzenfunktion. Der Vorteil eines nutzen-basierten **RS** besteht darin, dass es auch auf Attribute eingehen kann, welche nicht direkt mit dem Objekt zusammenhängen. Somit kann das **RS** zum Beispiel die Lieferzeit höher bewerten als den Preis, wenn ein akutes Bedürfnis besteht.
- **Wissensbasierte:** Ein wissensbasiertes **RS** generiert Vorschläge aufgrund von Schlussfolgerungen über die Vorlieben und Bedürfnisse der Benutzer. Es besitzt Wissen, welche Objekte bestimmte Bedürfnisse der Benutzer befriedigen. Somit können mögliche Empfehlungen mit den Benutzerbedürfnissen assoziiert werden. Das Benutzermodell muss dabei keine bestimmte Struktur haben. Es beinhaltet im einfachsten Fall nur die Anfrage des Benutzers, wie zum Beispiel bei Google. Komplexere **RSs** benutzen Methoden wie fallbasiertes Schließen.

Es existieren unterschiedliche Arten, wie das Wissen bei wissensbasierten **RS** gespeichert wird. Google zum Beispiel verwendet Links zwischen Seiten, um auf die Popularität und die Zuverlässigkeit zu schließen. Entree hingegen vergleicht Restaurants anhand des Wissens über deren Küchen.

Tabelle 2.1.: Methoden der Vorschlaggenerierung von **RSs** nach **Burke (2002)**

Technik	Daten	Input	Prozess
CF	Ratings von U für die Objekte in I	Ratings von u für die Objekte in I	Identifizieren der Benutzer in U , welche ähnlich sind wie u , und Berechnung der Bewertung für i anhand der Bewertungen dieser Benutzer.
Inhaltsbasiert	Eigenschaften der Objekte in I	Ratings von u für die Objekte in I	Klassifizierung des Ratingverhaltens von u und Anwendung dieser auf i
Demographisch	Demographische Informationen über U und deren Ratings für die Objekte in I	Demographische Informationen über u	Identifizieren der Benutzer, welche demographisch ähnlich sind wie u , und Berechnung der Bewertung für i anhand der Bewertungen dieser Benutzer.
Anwendungsspezifisch	Eigenschaften der Objekte in I	Eine Nutzenfunktion, welche die Präferenzen von u beschreibt, über die Objekte in I	Anwenden der Nutzenfunktion auf die Objekte und Berechnung des Rangs von i
Wissensbasiert	Eigenschaften der Objekte in I . Wissen darüber, wie diese Objekte Bedürfnisse eines Benutzers befriedigen können.	Eine Beschreibung von den Bedürfnissen oder Interessen von u	Ableiten einer Übereinstimmung von i und den Bedürfnissen von u

2.1.4. Vergleich der Methoden

Die in Kapitel 2.1.3 vorgestellten Methoden, werden in diesem Kapitel miteinander verglichen. Nach **Burke (2002)** können **RSs** wie folgt verglichen werden:

Alle der oben genannten Methoden haben Stärken und Schwächen. Diese werden im Folgenden genauer beschrieben und in Tabelle 2.2 zusammengefasst. Das vermutlich bekannteste Problem ist das Kaltstartproblem. Das Kaltstartproblem besteht aus zwei unterschiedlichen, aber dennoch voneinander abhängigen Problemen.

Das eine Problem sind **neue Benutzer**. Da Empfehlungen auf dem Vergleich zu anderen Benutzern beruhen, können für neue Benutzer keine personalisierten Empfehlungen gemacht werden.

Das andere Problem sind **neue Objekte**. Wie bei neuen Benutzern können auch bei neuen Objekten keine personalisierten Empfehlungen erstellt werden, solange noch nicht genug Bewertungen für diese vorhanden sind. Kollaborative **RSs** erstellen ihre Empfehlungen anhand von Benutzern, welche dieselben Objekte ähnlich bewerten. Dies funktioniert nicht, wenn die Dichte der Bewertungen relativ klein ist. Also wenige Benutzer dieselben Objekte bewertet haben.

Am besten funktioniert ein **RS** mit kollaborativen Filtern, wenn viele Bewertungen für die Objekte existieren und die Fluktuation der Objekte gering ist. Ändert sich die Menge der Objekte zu schnell, werden ältere Ratings irrelevant. Dies hat zur Folge, dass die Ratings der neuen Benutzer nicht mit denen der Benutzer verglichen werden können, welche schon länger im System existieren. Somit ist kollaboratives Filtern in diesem Fall ineffektiv.

Kollaboratives Filtern funktioniert am Besten für Benutzer, welche in eine Randgruppe gehören mit vielen Nachbarn. Die Methode funktioniert nicht für sogenannte „gray sheeps“, welche sich in der Mitte zwischen zwei Gruppen befinden. Dieses Problem existiert auch für demographische Methoden. Dafür haben diese keine Probleme mit neuen Benutzern, da keine Ratings für die Generierung der Empfehlungen erforderlich sind. Durch die verstärkte Sensibilisierung für Online-Datenschutz existieren wenige demographische **RSs**, gerade im Bereich des elektronischen Handels. Dies ist begründet durch die Tatsache, dass Benutzer ihre Präferenzen typischerweise nicht gerne bekanntgeben.

Inhaltsbasierte **RSs** haben ebenfalls das Startproblem. Sie benötigen genügend Ratings, um einen zuverlässigen Qualifizierer zu generieren. Im Gegensatz zum kollaborativen Filtern haben inhaltsbasierte Methoden das Problem, dass diese limitiert sind durch die Eigenschaften, welche explizit mit den Objekten assoziiert sind. Ein Empfehlungssystem für Filme ist zum Beispiel limitiert durch die Eigenschaften der Filme, wie Schauspieler, Regisseur, Drehort oder Anzahl der Wendungen. Der Film selbst ist dem System unbekannt. Somit sind solche Systeme stark von der Qualität der verfügbaren Objektbeschreibungen abhängig. Da kollaborative Methoden für Empfehlungen nur die Ratings verwenden, benötigen diese keine Objektbeschreibungen. Einige Experimente haben gezeigt, dass selbst wenn Objektbeschreibungen vorhanden sind,

kollaborative Methoden exakter sein können als inhaltsbasierte.

Die größte Stärke des kollaborativen gegenüber dem inhaltsbasierten Ansatz ist, dass dieser ohne weitere Anpassungen genreübergreifend funktioniert. Ein Actionfilmfan schaut möglicherweise auch gerne Dokumentarfilme. Doch ein inhaltsbasiertes **RS** welches auf einen Actionfilmfan trainiert ist, kann keine Vorschläge für Dokumentarfilme generieren. Da diese in der Regel keine Eigenschaften teilen, wie Darsteller, Drehort, Regisseur oder Budget. Nur wenn nicht nur die Präferenzen betrachtet werden, können solche Vorschläge generiert werden.

Sowohl inhaltsbasierte als auch kollaborative Methoden leiden unter dem „portfolio effect“. Im Idealfall wird kein **RS** ein bereits gewähltes Objekt vorschlagen. Einem Benutzer wird kein gesehener Film oder gelesenes Buch vorgeschlagen werden. Das Problem wird noch viel komplizierter in Bereichen wie Nachrichtenartikel. Hier können ähnliche Nachrichten wie die, welche der Benutzer bereits gelesen hat, dennoch von Wert für diesen sein. Sie können wesentliche neue Fakten oder Perspektiven beinhalten. Eine Nachricht von einer anderen Zeitung und anders geschrieben aber mit den selben Inhalten, wie eine bereits gelesene, ist hingegen ohne Nutzen für den Anwender.

Nutzen- und wissensbasierte **RSs** haben kein Kaltstartproblem oder ein Problem mit zu niedriger Dichte an Bewertungen, da ihre Empfehlungen nicht auf gesammelten statistischen Daten beruhen. Anwendungsbezogene Techniken erfordern eine Nutzenfunktion, welche alle Eigenschaften der Objekte mit einbezieht. Ein Vorteil dieser Methode ist, dass auch Eigenschaften betrachtet werden können, welche das Produkt nicht direkt betreffen, wie zum Beispiel das Lieferdatum. Diese Eigenschaften sind sehr spezifisch. So hat das Lieferdatum zum Beispiel eine hohe Bedeutung für einen Benutzer, der kurz vor dem Ablauf einer Frist steht. Daher lassen solche Systeme den Nutzer die relevanten Eigenschaften definieren.

Da der Benutzer bei anwendungsspezifischen Systemen definieren muss, welche Eigenschaften in welchem Maße für ihn relevant sind, haben anwendungsspezifische Systeme weniger Flexibilität. Dies führt häufig zu einer erhöhten Interaktion mit dem System. Bei Objekten mit wenigen Eigenschaften, wie Lieferdatum und Preis, ist dieses für den Benutzer möglich. Für Objekte mit einer höheren Komplexität, wie Filme oder Nachrichten ist dieses nicht möglich. Die Möglichkeit, die Eigenschaften komplex zu definieren, ist für einige Benutzer sinnvoll. Typische Nutzer hingegen werden dadurch überfordert. Auch signifikante Änderungen in der Produktpalette sind ein Problem für anwendungsspezifische **RSs**. So erfordert zum Beispiel ein Sprung von Sport- zu Familienautos eine komplette Refaktorisierung des System, da die wesentlichen Eigenschaften deutlich divergieren.

Wissensbasierte **RSs** haben - wie alle wissensbasierten Systeme - den Nachteil, dass das Wissen

akquiriert werden muss. Drei unterschiedliche Arten von Wissen können von solchen Systemen verwendet werden:

- **Katalogwissen:** Wissen über die Objekte, für welche Vorschläge generiert werden sollen, und deren Eigenschaften.
- **Funktionales Wissen:** Das System muss eine Abbildung zwischen Benutzerwünschen und den Objekten, welche diese Bedürfnisse befriedigen, erstellen.
- **Benutzerwissen:** Um personalisierte Vorschläge generieren zu können, benötigt das System Wissen über den Benutzer. Dieses ist im einfachsten Fall allgemeine demographische Daten. Komplexere Systeme verwenden spezifische Informationen über die Bedürfnisse, für die Vorschläge generiert werden sollen.

Wissensbasierte **RSs** haben den Vorteil, dass weniger Informationen über die Nutzer erforderlich sind, als dies bei anwendungsspezifischen **RSs** der Fall ist. Ein weiterer Vorteil ist, dass keine Startphase erforderlich ist, in welcher die Vorschläge wenig Qualität haben. Sie leiden somit nicht unter dem Kaltstartproblem. Benutzer aus Randgruppen werden schlechter identifiziert als bei kollaborativen **RSs**. Das Spektrum der Vorschläge ist abhängig von der Wissensdatenbank. Die Vorschläge können also sehr spezifisch sein, wenn das Wissen des Systems umfangreich genug ist.

2.1.5. Hybride Recommender Systems

Ein hybrides **RS** verwendet zwei oder mehr der oben genannten Methoden. Dabei ist das Ziel, die Schwächen einer Methode durch die Stärken einer weiteren Methode auszugleichen. Häufig wird das **CF** mit einer anderen Methode kombiniert, um das Kaltstartproblem des **CF** durch die zweite Methode auszugleichen. Zum Kombinieren wird eine der folgenden sieben Methoden verwendet.

Weighted

Bei einem gewichteten hybriden **RS** wird der Rang eines Objektes anhand aller im System vorhandener **RSs** berechnet. Im einfachsten Fall ist die Bewertung eine lineare Kombination der Bewertungen.

Die jeweiligen Sets der Kandidaten werden unter Berücksichtigung aller beteiligten Methoden bewertet. Hierfür existieren zwei unterschiedliche Ansätze:

Tabelle 2.2.: Vergleich der Methoden nach **Burke** (2007)

Technik	Vorteil	Nachteil
CF	A. Kann Genre übergreifende Nischen identifizieren. B. Kein domänenspezifisches Wissen erforderlich. C. Adaptiv: Qualität verbessert sich mit der Zeit. D. Implizites Feedback ist ausreichend.	I. Kaltstartproblem bei neuen Benutzern J. Kaltstartproblem bei neuen Objekten. „gray sheep“ Problem. L. Qualität hängt von einer großen Menge an historischen Daten ab. M. Stabilität gegen Verformbarkeit Problem.
Inhaltsbasiert	B, C, D	I, L, M
Demographisch	A, B, C	I, K, L, M N. Muss demographische Daten erfassen.
Anwendungsspezifisch	E. Keine Startphase erforderlich. F. Empfindlich für Änderungen der Präferenzen. G. Eigenschaften, welche nicht direkt das Produkt betreffen, können miteinbezogen werden.	O. Der Benutzer muss die Nutzenfunktion eingeben. P. Generierung der Vorschläge ist statisch (kann nicht lernen).
Wissensbasiert	E, F, G H. Benutzerbedürfnisse können auf die Objekte abgebildet werden.	P Q. Wissensverarbeitung erforderlich.

- **Schnittmenge:** Es werden nur Objekte berücksichtigt, die für beide Methoden als Kandidaten betrachtet werden. Dieser Ansatz hat den Nachteil, dass in einigen Fällen nur ein geringer Teil der Objekte berücksichtigt werden kann.
- **Vereinigungsmenge:** Es werden alle Objekte berücksichtigt. Das Problem dieser Methode ist, dass nicht jeder Algorithmus alle Objekte bewerten kann. Eine Möglichkeit ist hier, ausgeglichene oder neutrale Bewertungen zu verwenden, wenn kein Wert berechnet werden kann.

Mixed

Ein gemischtes hybrides **RS** vereinigt die Vorschläge von zwei oder mehreren **RSs** nebeneinander. Ein Vorteil dieser Methode ist, dass recht schnell viele Vorschläge generiert werden können.

Eine Herausforderung ist die nachträgliche Evaluierung, welchen Effekt das Hinzufügen oder Entfernen eines **RS** auf die Qualität des Gesamtsystems hat. Das Problem ist, dass die Ergebnisse aller **RSs** zusammen dargestellt werden.

Switching

Empfehlungssysteme dieser Art wählen für eine Vorhersage einer bestimmten Situation die potentiell erfolgreichste Methode aus. So können die Stärken der jeweiligen Methoden optimal ausgenutzt werden. Zum Beispiel kann die Schwäche des **CF** bei neuen Items durch die Verwendung einer Methode umgangen werden, welche nicht unter dem Kaltstartproblem leidet. Die Auswahl der optimalen Methode erhöht die Komplexität.

Feature combination

Bei einem **RS** mit feature combination wird nur eine bestimmte Methode verwendet. Die verwendeten Eigenschaften werden von einem beitragenden virtuellen **RS** erweitert. So können zum Beispiel die Eigenschaften eines inhaltsbasierten **RS** um kollaborative Eigenschaften, wie Bewertungen, erweitert werden.

Feature augmentation

Bei einem hybriden **RS** mit feature augmentation wird eine Technik benutzt, um eine Bewertung oder Klassifizierung durchzuführen. Diese wird von der zweiten Methode zur Generierung der Vorschläge verwendet. Der Vorteil ist, dass eine Methode verbessert werden kann, ohne dass diese verändert werden muss.

Cascade

Bei einer kaskadierten Anordnung der beteiligten **RSs** wird der das primäre **RS** für die Vorschlagsgenerierung verwendet. Das zweite **RS** dient der Verfeinerung der Ergebnisse.

Meta-level

Ein meta-level hybrid **RS** ist vergleichbar mit einem mit feature augmentation. Der Unterschied besteht darin, dass die zugrunde liegende Wissensbasis komplett verändert wird. Das zweite **RS** arbeitet mit einem erlernten Modell und nicht mit der ursprünglichen Wissensbasis. Um die Qualität der Empfehlungen eines **RS** bewerten zu können müssen diese evaluiert werden. Das nächste Kapitel zeigt die unterschiedlichen Metriken, um **RS** zu bewerten.

2.1.6. Evaluierung

Um die Qualität eines **RS** zu bestimmen, müssen die Ergebnisse unterschiedlicher **RSs** verglichen werden. Zur Evaluierung von **RSs** existieren diverse Ansätze. So können einfache Distanzfunktionen verwendet werden. Eine weitere Möglichkeit ist die Betrachtung, welche Vorschläge korrekt sind oder ob welche fehlen. Folgende Methoden werden im Kontext dieser Arbeit für die Evaluierung der unterschiedlichen Methoden verwendet:

- **MAE (Mean Average Error)**: Die einfachste Möglichkeit zum Vergleichen von zwei **RSs** ist die Betrachtung des durchschnittlichen Abstands zwischen tatsächlicher und berechneter Bewertung, also der durchschnittliche Fehler der jeweiligen Methoden. Je kleiner dieser Wert ist, desto genauer sind die Berechnungen des **RS**.
- **RSME (Rooted Square Mean Error)**: Geringe Abweichungen beeinträchtigen die Qualität eines **RS** weniger als große. Eine Möglichkeit, dies zu berücksichtigen, ist das Quadrieren der Abstände. Auf diese Weise fallen große Abweichungen stärker ins Gewicht.
- **precision**: Bei der precision wird betrachtet, wie viele Vorschläge korrekt oder falsch sind. Eine Möglichkeit ist die Betrachtung, welche der Vorschläge tatsächlich von dem Benutzer angenommen werden. Als „false positive“ werden Vorschläge bezeichnet, die zwar vorgeschlagen, aber nicht vom Benutzer angenommen werden. Das Verhältnis von angenommenen zu vorgeschlagenen Objekten wird als precision bezeichnet.
- **recall**: Eine weitere Möglichkeit ist die Betrachtung, welche der vorgeschlagenen Objekte von dem Benutzer tatsächlich gewählt werden. Für ein **RS** ist dies im Betrieb schwer oder gar nicht zu berechnen. Für eine korrekte Berechnung ist ein Trainingsdatensatz erforderlich. Da im Kontext dieser Arbeit ein solcher Trainingsdatensatz erstellt worden ist, kann die Berechnung durchgeführt werden.
- **f-score**: Beim f-Score handelt es sich um eine Kombination aus precision und recall. Diese Metrik betrachtet precision und recall zusammen, da gute Werte für beide für ein erfolgreiches **RS** erforderlich sind.
- **nDCG**: Beim nDCG wird betrachtet, wie korrekt die Reihenfolge der Vorschläge ist. Diese Metrik betrachtet weniger konkrete Wertigkeiten der Objekte. Es wird lediglich analysiert, ob die Objekte, welche dem Benutzer am besten gefallen, vorne in der Liste der Vorschläge enthalten sind.

2.1.7. Aktuelle Forschung

Bei der Aktuellen Forschung im Bereich der **RSs** ist vor allem „The ACM Conference on Recommender Systems“ zu nennen. Weitere Konferenzen, welche dieses Thema ebenfalls behandeln, sind: ACM’s Special Interest Group on Information Retrieval (SIGIR); User Modeling, Adaptation and Personalization (UMAP); Intelligent User Interfaces (IUI); World Wide Web (WWW); und ACM’s Special Interest Group on Management Of Data (SIGMOD). Themen in diesen Konferenzen, welche ebenfalls den Kontext dieser Arbeit tangieren, sind:

- das Kaltstartproblem und
- hybride **RSs**.

Weitere Themen, die nicht im Kontext dieser Arbeit behandelt werden, sind:

- Einbeziehung der Benutzer in die Vorschlagsgenerierung,
- Teilen von Accounts,
- Einbeziehung sozialer Netzwerke,
- Einbeziehung des Kontextes und
- die Auswahl und Anordnung von Online Video Werbung.

2.1.8. Anforderungen und Probleme

Um für die Benutzer Vorschläge zu generieren, müssen folgende Probleme betrachtet werden:

- Das **Kaltstartproblem**: Ein Benutzer meldet sich neu an dem System an. Es gibt für ihn noch keine oder zu wenig Daten über seine Präferenzen. Das Kaltstartproblem widmet sich diesem Problem, um dem Benutzer dennoch angemessene Vorschläge zu generieren. Auch neue Objekte müssen bei den Vorschlägen berücksichtigt werden. **Li und Li (2013)** **Weiß u. a. (2008)**
- Der **Kontext**: Das Interesse an bestimmten Inhalten ist kontextsensitiv. Zum Beispiel wirkt sich der Zeitpunkt aus. So unterscheidet sich das Konsumverhalten in Abhängigkeit von Uhrzeit und Wochentag. Ebenfalls wirken sich die anwesenden Personen aus, da die Filme, welche alleine gesehen werden, sich von den Filmen unterscheiden, welche mit Freunden oder einem Partner gesehen werden. **Biancalana u. a. (2011)** **Shi u. a. (2010)** **Barjasteh u. a. (2015)**

- Die **Diversität**: Das System identifiziert die Interessen eines Benutzers. Hier besteht die Gefahr, dass das System nur noch Vorschläge generiert, die nur zu einem Teil der Interessen des Benutzers gehören. Das System befindet sich in einer sogenannten „Filter Bubble“ [Nguyen u. a. \(2014\)](#). Es ist von Interesse, ein möglichst breites Spektrum an Benutzerinteressen abzudecken. [Li und Li \(2013\)](#) [Vargas und Castells \(2014\)](#) Hierzu zählt auch, dass dem Benutzer nicht immer die gleichen Vorschläge gemacht werden. [Lathia u. a. \(2010\)](#)
- Die **Performance**: Gerade wenn über einen Browser auf ein System zugegriffen wird, muss es schnell sein [Perisic](#). Unabhängig von der Menge an Benutzern und Inhalten muss ein Vorschlagsystem schnelle Antworten liefern. [Das u. a. \(2007\)](#)
- Die **Disruption**: Gravierende Erlebnisse eines Benutzers können das Konsumverhalten deutlich verändern. So kann zum Beispiel eine Schwangerschaft das Interessengebiet einer Nutzerin erweitern. Ein Vorschlagsystem sollte starke Veränderungen erkennen und auf diese reagieren können.
- **Geteilte Accounts**: Accounts können auch von mehreren Personen verwendet werden. Das System sollte das erkennen. Wenn nicht identifiziert werden kann, welcher Benutzer gerade mit dem System interagiert, sollten zumindest Vorschläge für alle Benutzer des Accounts generiert werden. [Verstrepen und Goethals \(2015\)](#)

Für die Verwendung eines **RS** ist ein Trainingsdatensatz erforderlich. Zur Evaluierung eines **RS** ist ein Testdatensatz erforderlich. Zum Erstellen dieser beiden Datensätze wurden im Rahmen dieser Arbeit zwei Experimente durchgeführt. Diese sind in Kapitel 3.3 ausführlich beschrieben. Für diese Experimente war eine Votingplattform erforderlich. Sie wird im nächsten Kapitel erklärt.

2.2. Votingplattform

In diesem Kapitel wird die Erstellung einer Votingplattform beschrieben, welche für das Experiment aus Kapitel 3.3 erforderlich ist. Um eine Votingplattform zu erstellen sind folgende Komponenten erforderlich:

- eine **Registrierung**,
- ein **Login**,
- ein **Web Frontend**,

- eine **Persistenz** und
- ein **Votingsystem**.

2.2.1. Registrierung

Damit sich die Benutzer am System anmelden können, müssen diese zunächst registriert werden. Einerseits muss die Registrierung die Benutzerdaten validieren, andererseits müssen neue Benutzer angelegt und gespeichert werden.

Ein neuer Benutzer muss sich mit einer E-Mailadresse und einem Passwort registrieren. Zunächst wird überprüft, ob diese Adresse schon im System vorhanden ist. Existiert noch kein Benutzer mit einer solchen Adresse, wird eine Authentifizierungsmail an die E-Mailadresse gesendet. Auf diese Weise kann sichergestellt werden, dass dem Benutzer diese auch gehört. Einerseits kann ein Benutzer eindeutig anhand seiner Mailadresse identifiziert werden, andererseits kann diese für die weitere Kommunikation mit dem Benutzer verwendet werden.

Nach einer erfolgreichen Registrierung muss ein Account im System angelegt und in der Datenbank gespeichert werden. Die Daten werden zum Speichern an die Persistenzkomponente weitergegeben. Diese stellt sie der Loginkomponente zur Verfügung.

2.2.2. Login

Um den Benutzer bei der Interaktion mit dem System eindeutig identifizieren zu können ist ein Login erforderlich. Damit ein Benutzer sich im System einloggen kann, ist eine Registrierung erforderlich. Der Benutzer muss beim Login seine E-Mailadresse und sein Passwort eingeben. Nach einem erfolgreichen Login wird eine Session erstellt. Dies ermöglicht dem System, alle Aktionen eindeutig dem jeweiligen Benutzer zuzuordnen. Um einen für den Kontext dieser Arbeit ausreichenden Schutz zu gewährleisten, muss die Anzahl der Loginversuche von einer IP-Adresse eingeschränkt werden. Somit wird eine IP-Adresse blockiert, wenn von dieser zu viele fehlgeschlagene Anmeldeversuche kommen. Für den Fall, dass ein Benutzer sein Passwort vergessen hat, muss ihm von der Loginkomponente eine Möglichkeit der Passwortwiederherstellung gegeben werden.

2.2.3. Webfrontend

Zur Interaktion des Benutzers mit dem System ist ein Webfrontend erforderlich. Dies ist die Schnittstelle zwischen Benutzer und System. Für die Registrierung wird dem Benutzer eine Registrierungsmaske zum Eingeben der E-Mailadresse und des Passwortes zur Verfügung

gestellt. Diese Informationen werden an die Registrierungskomponente weitergeleitet. Deren Feedback wird über das Webfrontend wieder an den Benutzer kommuniziert. Ebenfalls wird eine Anmeldemaske zur Verfügung gestellt, die es dem Benutzer ermöglicht, mit der Login-komponente zu kommunizieren. Schließlich muss das Webfrontend noch eine Navigation, allgemeine Informationen zu dem Experiment und eine Votingoberfläche bieten. Letztere kommuniziert mit dem Votingsystem.

2.2.4. Votingsystem

Damit dem Benutzer Fragen gestellt und seine Antworten verarbeitet werden können, wird ein Votingsystem benötigt. Es muss die Möglichkeit bestehen, Fragen und vorgefertigte Antworten zu definieren. Der Benutzer beantwortet diese im Webfrontend, und seine Antworten werden von dem Votingsystem verarbeitet und in der Datenbank mithilfe der Persistenzkomponente abgespeichert.

2.2.5. Persistenz

Für Persistenz und einen schnellen Zugriff auf die Daten ist eine Datenbank erforderlich. In dieser werden folgende Daten gespeichert:

- Die E-Mailadresse und ein Hashwert des Passworts der **Benutzer**.
- Die **Bewertungen** der Filme und Kategorien mit Zeitpunkt, Wert und dem Benutzer.
- Die **Antworten bei den Allgemeinen Fragen** von den Benutzern.

Die mit der in diesem Kapitel beschriebenen Votingplattform gewonnenen Trainings- und Testdatensätze müssen vom **RS** mit den Schritten des Knowledge Discovery in Databases (**KDD**) Prozesses verarbeitet werden. Dieser Prozess ist im nächsten Kapitel genauer vorgestellt.

2.3. Knowledge Discovery in Databases

In diesem Kapitel werden die grundsätzlichen Schritte erklärt, welche für die Gewinnung von Wissen aus Datenbanken benötigt werden. Im Wesentlichen sind folgende Schritte erforderlich **Fayyad u. a. (1996)**:

- eine **Auswahl** der Daten, die für den Prozess verwendet werden sollen,
- eine **Vorverarbeitung** der Daten,

- eine **Transformation** der Daten in eine gewünschte Struktur,
- ein **Datamining-Prozess** und
- eine **Interpretation** der Ergebnisse.

2.3.1. Auswahl der Daten

Zunächst müssen die Daten extrahiert werden, die für die Generierung der Vorschläge relevant sind. Im Kontext dieser Arbeit werden die Benutzer entfernt, die nicht am zweiten Voting teilgenommen oder nur einen Teil der Filme bewertet haben. Ebenfalls werden Benutzer mit einem auffälligen Verhalten entfernt. Dabei handelte es sich um Benutzer,

- die das Voting zu schnell durchlaufen haben;
- die immer das Gleiche voten;
- die immer eine bestimmte Folge voten, wie zum Beispiel eine alternierende Folge (-3, 3, -3, 3, -3, 3).

Nach der Auswahl der Daten müssen diese vorverarbeitet werden. Dies wird im nächsten Kapitel genauer erläutert.

2.3.2. Vorverarbeitung

Um die ausgewählten Daten für die Vorschlagsgenerierung verwenden zu können, müssen diese in einem zweiten Schritt verarbeitet werden. Zum einen werden die Benutzermodelle erstellt, zum anderen werden die Filmdaten zusammengefasst. Im Kontext dieser Arbeit werden folgende Eigenschaften in den Benutzerprofilen gespeichert:

- das Alter,
- das Geschlecht,
- das Konsumverhalten,
- die Bewertungen der Filme und
- die Bewertungen der Kategorien.

Zu den Filmen werden folgende Eigenschaften gespeichert:

- die Kategorie,

- das Land, in welchem der Film produziert wurde, und
- die Bewertungen des Films.

Die verarbeiteten Daten werden für die Vorhersagen transformiert.

2.3.3. Transformation

Bei der Transformation werden die verarbeiteten Film- und Benutzerdaten in Clustern zusammengefasst. Im Kontext dieser Arbeit werden nur Cluster von den Film- und Benutzerdaten gebildet. Die Cluster werden anhand der Kategorien erstellt. Nach der Bildung der Cluster werden diese für das Data Mining verwendet. Dies entspricht der Methode von [Gantner u. a. \(2010\)](#) [Abbildung 2.2](#) liefert eine Darstellung des Data Clustering.

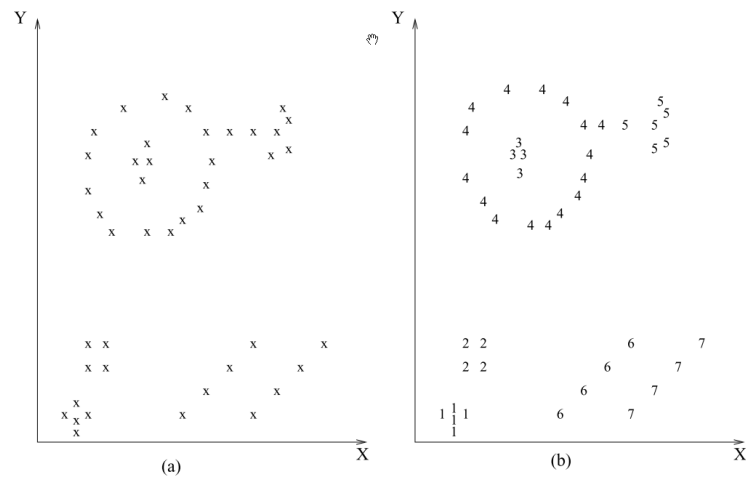


Abbildung 2.1.: Darstellung des Data Clusterings [Jain u. a. \(1999\)](#)

Abbildung 2.2.:

2.3.4. Data Mining

Die gebildeten Cluster werden beim Data Mining verwendet. Zum Generieren von Vorschlägen existieren viele unterschiedliche Methoden. Im Rahmen dieser Arbeit erfolgt die Berechnung einer Clusterbewertung anhand der Film- und Kategoriebewertungen. Unter Verwendung dieser Clusterbewertungen wird die Bewertung für einen Film berechnet. Nachdem die Vorschläge generiert wurden, werden diese interpretiert.

2.3.5. Interpretation

Die generierten Vorschläge müssen analysiert und interpretiert werden. In dieser Arbeit wird der Abstand zwischen berechneten und tatsächlichen Bewertungen berechnet. Dieser wird als „mean absolute error“ bezeichnet [Baltrunas \(2008\)](#) [Biancalana u. a. \(2011\)](#) [Lakiotaki u. a. \(2008\)](#). Mit den aus der Analyse gewonnenen Erkenntnissen können Verbesserungen für die vorherigen Schritte erarbeitet werden. So stellte sich im Kontext dieser Arbeit heraus, dass ein Reduzieren der Kategorien pro Film und die Angabe des Herkunftslandes die Ergebnisse verbessert.

2.4. Fazit

Die in diesem Kapitel gewonnenen Erkenntnisse werden verwendet um das im nächsten Kapitel beschriebene Experiment durchzuführen. Hierfür müssen das [RS](#), die Votingplattform und der [KDD](#) Prozess implementiert werden.

3. Experiment

Für die Betrachtung von **RSs** wurde im Rahmen dieser Masterarbeit folgendes Experiment durchgeführt: zunächst fand eine Umfrage statt, bei der die Teilnehmer 50 Filme und 23 Kategorien bewerteten. Ungefähr ein Dreivierteljahr später wurden die Personen gebeten, zehn weitere Filme zu bewerten, die beim ersten Mal alle Filme bewertet hatten. Parallel dazu wurde ein **RS** entwickelt, um die Bewertungen der Teilnehmer zu berechnen.

3.1. Ziel

Das Ziel des Experiments ist die Erstellung eines Trainingssets. Mit diesem können weitere Experimente durchgeführt werden, um unterschiedliche **RSs** oder Einstellungen dieser **RSs** zu vergleichen. Somit können folgende Ziele für das Experiment definiert werden:

1. Erstellung eines Trainingssets,
2. Analyse der Teilnehmer bezogen auf deren Bewertungsverhalten,
3. Analyse der Filmbewertungen unter Berücksichtigung des Herkunftslandes und Budgets des Films,
4. Validierung von Algorithmen,
5. Validierung von Einstellungen und
6. Optimierung der Algorithmen, durch - unter anderem - eine Fehlerminimierung.

3.2. Erstellung einer Votingplattform

Zur Durchführung eines Votings ist eine Votingplattform erforderlich. Das ist eine Webseite, auf der die Teilnehmer ihre Bewertungen für die Filme und Kategorien durchführen können. Wie in Kapitel 2.2 beschrieben, sind folgende Komponenten erforderlich, um eine Votingplattform zu erstellen: eine Registrierung, ein Login, eine Web Frontend, eine Persistenz und ein

Votingsystem. Die Webseite wurde als eine Wordpress-Installation umgesetzt. Das nächste Kapitel beschreibt das Framework Wordpress. In den folgenden Kapiteln wird beschrieben, wie die Komponenten des Votingsystems realisiert wurden.

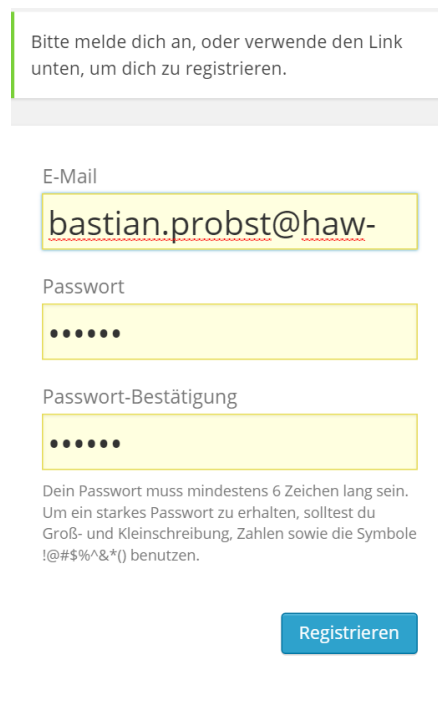
3.2.1. Wordpress

Wordpress ist eine freie Webanwendung, um Inhalte einer Webseite zu verwalten. Wordpress bietet die Darstellung von Webseiten, eine Menüleiste zum Navigieren und einen Login. Durch die Verwendung von Plugins, können viele weitere Funktionen hinzugefügt werden. [Wikipedia \(2016b\)](#)

3.2.2. Registrierung und Login

Der Login ist eine Standardfunktion von Wordpress. Benutzer können sich über einen Benutzernamen und ein Passwort anmelden. Nach erfolgreicher Anmeldung können ihnen mit dem System alle Interaktionen eindeutig zugeordnet werden. Um sich auf der Webseite anmelden zu können, ist eine Registrierung erforderlich. Zur Realisierung der Registrierung wurden folgende Plugins verwendet:

- **Register:** Das Plugin „Register“ erweitert die Loginmaske um einen Link für die Registrierung. Durch Klicken auf diesen Link wird eine Registrierungsmaske aufgerufen. Der Benutzer muss dort seine Mailadresse, ein gewünschtes Passwort und eine Passwort-Bestätigung angeben. Danach wird dem Benutzer eine Mail mit einem Registrierungslink geschickt. Der Aufruf dieses Links registriert den Benutzer. Das Plugin bietet die zusätzliche Funktion, ein vergessenes Passwort wiederherzustellen.
- **Privat only:** Damit Benutzer nur angemeldet mit dem System interagieren können, wurde das Plugin „Privat only“ verwendet. Dieses leitet den Benutzer beim Aufruf der Webseite zum Login weiter. Darüber hinaus wird dem Benutzer eine Information angezeigt, die ihn darauf hin weist, dass er sich anmelden oder registrieren muss. [Abbildung 3.1](#) zeigt das Registrierungsformular mit dem Hinweis zur Registrierung.



Bitte melde dich an, oder verwende den Link unten, um dich zu registrieren.

E-Mail
bastian.probst@haw-

Passwort
•••••

Passwort-Bestätigung
•••••

Dein Passwort muss mindestens 6 Zeichen lang sein. Um ein starkes Passwort zu erhalten, solltest du Groß- und Kleinschreibung, Zahlen sowie die Symbole !@#\$\$%^&*() benutzen.

Registrieren

Abbildung 3.1.: Registrierungsformular

3.2.3. Web Frontend

Als Web Frontend wurde eine Wordpress Installation verwendet. Es wurden Seiten für die Votings und eine für weitere Informationen erstellt. Neben einfachen Texten und den Votings, welche in Kapitel 3.2.4 genauer beschrieben werden, wurden auch Youtube Videos dargestellt. Da nicht jeder Teilnehmer alle Filme gesehen hat, wurden den Teilnehmern die Trailer zur Verfügung gestellt. Das Darstellen von Youtube Videos ist eine Standardfunktion bei Wordpress. Der einzige Nachteil besteht darin, dass die Ladezeit der Seite negativ beeinflusst wird. Da sich zehn Videos auf einer Seite befinden, wurden die Ladezeiten der Seiten zu lang. Um dieses Problem zu beheben, wurde das Plugin „Youtube Lite“ verwendet. Mit diesem Plugin werden für die Videos nur die minimal erforderlichen Informationen geladen. Erst beim Betrachten eines Videos werden die fehlenden Inhalte nachgeladen.

3.2.4. Votingsystem

Das Votingsystem ist mit dem Plugin „Yop Poll“ realisiert worden. Dieses Plugin bietet die Funktion, Fragen zu erstellen. Die Antworten können dabei fest oder als Freitext definiert

FILME

Mad Max: Fury Road (2015)



Abbildung 3.2.: Darstellung eines Youtube Videos auf der Webseite

werden. Im Kontext dieser Arbeit wurden die Antworten fest definiert. Die Bewertungen für Kategorien haben eine Skala von -3 bis 3.

Die Anzahl der Bewertungen für einen Film wurden pro Benutzer auf eins limitiert. Die Anzeige des Votings zeigt jedoch nicht an, ob ein Film bereits bewertet worden ist. Um dies zu realisieren muss das „Yop Poll“ erweitert werden. Die Datei „yop_poll_model.php“ beinhaltet die logische Steuerung des Datenmodells. In dieser ist unter anderem die Methode „return_poll_html“ definiert, welche die HTML-Repräsentation eines Votings für einen Benutzer unter bestimmten Bedingungen liefert. Diese muss um folgende Abfrage erweitert werden:

```
1 if( $this->hasPollsLeft()){
2     $template .= ?<div id="yop-poll-container-error-?_.$poll_id
3     _.$unique_id_?" class="yop-poll-container-error"></div>?;
4 }
5 else {
6     $template .= ?<div id="yop-poll-container-error-?_.$poll_id_
7     _.$unique_id_?" class="yop-poll-container-error">
8     Dies hast du schon gevotet!</div>?;
```

```
9 }
```

Listing 3.1: Erweiterung von Yop Poll

In dieser Abfrage wird überprüft, ob der Benutzer für das Voting noch abstimmen darf. Hierfür wird die Methode „hasPollsLeft“ aufgerufen. Diese wurde wie folgt implementiert:

```
1 private function hasPollsLeft(){
2   global $current_user;
3   $voter = array();
4   $voter['poll_id']   = $this->poll['id'];
5   $voter['user_id']   = $current_user->ID;
6   $voter['user_type'] = 'wordpress';
7   return $this->user_have_votes_to_vote( $voter );
8 }
```

Listing 3.2: Methode: hasPollsLeft

In dieser Methode wird ein Array erstellt, in dem alle relevanten Informationen über den Benutzer abgespeichert werden. Mit diesem Benutzer wird die Methode „user_have_votes_to_vote“ aufgerufen. Das Resultat dieser Methode wird danach zurückgegeben. Sie ist wie folgt umgesetzt:

```
1 public function user_have_votes_to_vote( $voter ) {
2   $poll_options = $this->poll_options;
3   if ( $voter['user_id'] > 0 ){
4     if ( 'yes' == $poll_options['limit_number_of_votes_per_user'] ){
5       if ( $this->get_voter_number_of_votes( $voter )
6         >= $poll_options['number_of_votes_per_user'] ){
7         return false;
8       }
9     }
10  }
11  return true;
12 }
```

Listing 3.3: Methode zum Überprüfen, ob ein Benutzer noch abstimmen darf

Bei dieser Methode wird zunächst anhand der „user_id“ überprüft, ob ein valider Benutzer übergeben wurde. Danach wird abgefragt, ob für das Voting ein Limit definiert worden ist. Dem folgt eine Abfrage, ob die Menge der Votes des Benutzers größer oder gleich den erlaubten Votes ist. Ist dieses der Fall, gibt die Methode „false“ zurück. Ansonsten wird „true“ zurückgegeben.

Wie findest du den Film:
"Mad Max: Fury Road"?

- 3 (sehr schlecht)
- 2
- 1
- 0 (neutral)
- 1
- 2
- 3 (sehr gut)

VOTE

Abbildung 3.3.: Yop Poll Darstellung eines Votings

3.2.5. Persistenz

Um einen schnellen Zugriff auf die Daten und eine Persistenz der Daten sicherzustellen wurde ein Datenbankserver verwendet. Für die Votingsseite wurde der von dem Hostingprovider zur Verfügung gestellte „MySQL Community Server“ verwendet. Die verwendete WordPress-Installation legt automatisch eine Tabelle für die Nutzer an. Diese enthält die für die Login- und Registrierungskomponente erforderlichen Attribute:

- die **Kennung (ID)** des Benutzers,
- den **Loginnamen**, welcher der Mailadresse entspricht,
- die **Mailadresse** und
- ein **Hashwert des Passworts**.

Somit können alle erforderlichen Informationen für einen Login bei einer erfolgreichen Registrierung abgespeichert werden.

Das in Kapitel 3.2.4 beschriebene Plugin „Yop Poll“ erstellte unter anderem die Tabelle „yop_poll_logs“, in dieser werden alle getätigten Bewertungen abgespeichert. Hierzu werden folgende Daten bei der Durchführung einer Bewertung gespeichert:

- die **ID der Frage**,
- die **ID der Antwort**,
- die **ID des Benutzers** und
- das **Datum**.

Die konkreten Texte der Fragen und Antworten sind in den Tabellen: „polls“ und „poll_ answers“ gespeichert. Um die Anonymität der Benutzer zu erhalten, wurde bei den Benutzern nur die **ID** verwendet.

Nach der Erstellung der Votingplattform konnte das Experiment durchgeführt werden. Dieses wird im nächsten Kapitel beschrieben.

3.3. Voting

In diesem Kapitel werden die beiden Votings beschrieben. Hierfür wird die in Kapitel 3.2 beschriebene Votingplattform verwendet. Nach einer Betrachtung der Ziele wird die Akquirierung der Teilnehmer dargestellt. Abschließend wird die Durchführung betrachtet. Eine Analyse von Vorhersagesystemen im Kontext dieser Arbeit wurde in Kapitel 2.1 durchgeführt. Die Methode zum Berechnen der Bewertungen wird in Kapitel 3.4.4 genauer beschrieben.

3.3.1. Ziel des ersten Votings

Das Ziel des ersten Votings ist die Erstellung eines Trainingsdatensatzes. Ein Trainingsdatensatz ist für ein **RS** erforderlich, um unter anderem das Votingverhalten der Benutzer zu erlernen.

3.3.2. Akquirierung der Teilnehmer für das erste Voting

Zum Durchführen des Experiments war es erforderlich Teilnehmer zu akquirieren. Folgende Aktionen wurden hierfür durchgeführt:

- Bitten um Teilnahme in einer Erstsemester-Vorlesung.
- Bitten um Teilnahme in einem Kurs des NextMedia Studiengangs.
- Aufruf über den Mailverteiler der HAW.
- Anfragen im privaten Umfeld.

Um einen Anreiz für die Teilnahme zu schaffen, wurde ein Amazongutschein über 20 Euro verlost. Aus dem ersten Voting ist über die Teilnehmer folgendes bekannt:

- 322 Personen haben sich angemeldet.
- 90 Teilnehmer haben alles gevotet.
- 13 Teilnehmern fehlten noch 1-5 Votes.
- 5 Teilnehmern fehlten noch 6-10 Votes.
- 1 Teilnehmer fehlten noch 11-15 Votes.
- 3 Teilnehmern fehlten noch 16-20 Votes.

Eine genaue Analyse der Teilnehmer findet in Kapitel 4.1 statt.

3.3.3. Durchführung des ersten Votings

Die Webseite des ersten Votings bestand aus:

- einer Informationsseite,
- einer Seite für das Bewerten der 23 Kategorien und
- jeweils einer Seite für jede der fünf Kategorien, für die zehn Filme bewertet wurden.

Die Webseite für das Voting ist in acht Seiten aufgeteilt. Auf der Startseite wurden die Teilnehmer über den Verlauf des Experiments, die bisherigen Ausarbeitungen zu diesen Thema und das Ziel dieser Arbeit informiert.

Auf der zweiten Seite wurden den Teilnehmer Fragen zur Person gestellt. Diese betrafen Alter, Geschlecht und Konsumverhalten. Ebenfalls wurde den Teilnehmern die Möglichkeit geboten, Anregungen oder Kritik zu äußern.

Auf der nächsten Seite wurden die Teilnehmer gebeten, die 23 Kategorien der **IMDb** zu bewerten. Von diesen wurden Action, Biografie, Romantik, Science-Fiction und Sport näher betrachtet.

Auf den letzten fünf Seiten wurden die Teilnehmer aufgefordert, jeweils zehn Filme aus den fünf Kategorien zu bewerten. Da nicht jedem Teilnehmer alle Filme bekannt sind, waren die Trailer zu den Filmen als Youtube-Videos in die Seite eingebunden. Die Trailer waren hauptsächlich in deutscher Sprache. Nur wenn kein deutscher Trailer verfügbar war, wurde ein Trailer in einer anderen Sprache verlinkt.

Das erste Voting ist in **Probst (2015a)** detailliert vorgestellt worden.

3.3.4. Ziel des zweiten Votings

Um weitere Bewertungen zu erfassen, wurde das zweite Voting durchgeführt. Es hatte unter anderem das Ziel, Bewertungen mit einem zeitlichen Abstand zu erhalten. Auf diese Weise ist es möglich, die Bewertungen eines Filmes beim ersten Voting mit der Bewertung des Filmes beim zweiten Voting zu vergleichen. Hierbei kann eine Tendenz des sogenannten statistischen Rauschens annähernd ermittelt werden.

Ein weiterer Aspekt ist die Berechnung der Bewertungen von neuen Benutzern oder Filmen. Hier muss das Kaltstartproblem gelöst werden, weil [Albanese u. a. \(2013\)](#) [Bu u. a. \(2010\)](#) [Lin u. a. \(2013\)](#) Bewertungen für Filme berechnet werden, welche noch nicht von Benutzern bewertet wurden. Oder es müssen Bewertungen für Benutzer berechnet werden, die noch keine Filme bewertet haben.

Im Kontext dieser Arbeit existieren im Testdatensatz nur Bewertungen von bekannten Benutzern. Die Filme hingegen haben noch keine Bewertungen. Ein möglicher Ansatz zum Lösen dieses Problems ist der Vergleich mit ähnlichen Filmen.

Die meisten Algorithmen benötigen für ihre Vorhersagen einen Trainingsdatensatz [Said u. a. \(2014\)](#) [Bu u. a. \(2010\)](#) [Lakiotaki u. a. \(2008\)](#). Selbst bei dem Kaltstartproblem werden Trainingsdaten benötigt. Das Verhalten der unbekanntem Benutzer wird daher anhand von ähnlichen Benutzern berechnet. Bei den Filmen werden für die Berechnung der Bewertungen die Bewertungen ähnlicher Filme herangezogen.

Um die berechneten Bewertungen mit tatsächlichen vergleichen zu können, sind ein Trainings- und ein Testdatensatz erforderlich. Der Trainingsdatensatz wird von den Algorithmen zum Lernen verwendet. Der Testdatensatz wird für den Vergleich zwischen Berechnung und Realität benötigt [Said und Bellogín \(2014\)](#). Um die unterschiedlichen Algorithmen zu vergleichen, können Metriken wie der Abstand zwischen tatsächlicher und berechneter Bewertung verwendet werden. Weitere Vergleichsmöglichkeiten bieten die Betrachtung der „false positive“ und der „false negative“, „precision“, „recall“ oder „nDGC“ [Said u. a. \(2014\)](#). Eine ausführliche Analyse der Metriken und deren Anwendung wird in der Masterarbeit durchgeführt werden.

Somit haben die beiden Votings das Ziel, einen Trainings- und einen Testdatensatz zu erstellen. Die Daten des ersten Votings sind als Trainingsdatensatz zu betrachten. Beim zweiten Voting wurden die Testdaten erstellt. Die Durchführung des Votings wird im nächsten Kapitel beschrieben.

3.3.5. Akquirierung der Teilnehmer für das zweite Voting

Für das zweite Voting wurden alle Teilnehmer des ersten Votings per Mail angeschrieben, welche dieses komplett durchgeführt hatten. Um auch für das zweite Voting einen erhöhten Anreiz zu schaffen, wurde erneut ein Amazongutschein über 20 Euro verlost.

3.3.6. Durchführung des zweiten Votings

Bei dem zweiten Voting wurden zehn weitere Filmbewertungen erfasst. Dabei handelte es sich um jeweils zwei Filme aus den im ersten Voting verwendeten Kategorien (Aktion, Liebe, Sport, Science-Fiction und Biografien). Die Teilnehmer bei dem ersten Voting, welche alle Filme bewertet hatten, wurden um die Teilnahme am zweiten Voting gebeten. Dies waren ca. 100 Personen. Davon haben 42 das zweite Voting beendet. Am Login wurde nichts verändert, sodass die Teilnehmer die Logindaten des ersten Votings verwenden konnten. Für den Fall eines vergessenen Passwortes gab es - wie im ersten Voting - die Möglichkeit einer Passwortwiederherstellung. Das Ende des Votings wurde auf den 18. Juli 2015 gesetzt. Im zweiten Voting wurden keine Fragen zur Person gestellt.

Nach der Durchführung des ersten und des zweiten Experiments konnten die gewonnenen Trainings- und Testdatensätze von dem **RS** verwendet werden. Dieses wird im nächsten Kapitel genauer beschrieben.

3.4. Das Recommender System

Das im Rahmen dieser Arbeit entwickelte **RS** wurde in Java implementiert. Es verwendet die im ersten und zweiten Experiment gewonnenen Trainings- und Testdatensätze. Die beiden Experimente werden in Kapitel 3.3 genauer beschrieben. Das Programm durchläuft die von **Fayyad u. a. (1996)** definierten Schritte des in Kapitel 2.3 beschriebenen **KDD** Prozesses.

3.4.1. Auswahl der Daten

Zunächst werden alle Benutzer, Filme und Bewertungen aus der Datenbank geladen. Danach findet die Auswahl der Daten statt. Hierfür werden ausschließlich die abgegebenen Bewertungen der Kategorien und Filme betrachtet. Benutzer, die keine Angaben zur Person gemacht, aber alle anderen Fragen beantwortet haben, werden somit nicht aussortiert. Die Methode zum Selektieren der Benutzer wurde wie folgt implementiert:

```
1 private void selectUser() {  
2     for (User u : users) {
```

```
3   if (u.getGenreRatings().size() > 21) {  
4       if (u.getOldRatings().size() > 49) {  
5           if (u.getNewRatings().size() > 9) {  
6               selectedUsers.add(u);  
7           }  
8       }  
9   }  
10 }  
11 }
```

Listing 3.4: Methode zur Auswahl der Benutzer

Es wird über alle Benutzer iteriert. Zunächst wird überprüft, ob alle Kategorien bewertet wurden. Danach wird geprüft, ob alle Filme des ersten und zweiten Experiments bewertet wurden. Nur Benutzer, bei denen alles zutrifft, werden zu den ausgewählten Benutzern hinzugefügt.

Für die Filme wurde die Auswahl vor den Experimenten getroffen. So wurden zum einen die fünf Kategorien Aktion, Biografie, Romantik, Science-Fiction und Sport gewählt. Daraufhin wurden die am besten bewerteten Filme dieser Genres ausgewählt. Hier wurde als Bedingung definiert, dass diese in den letzten vier Jahren veröffentlicht wurden.

Nach der Auswahl der Daten müssen diese vorverarbeitet werden.

3.4.2. Vorverarbeitung

Nach der Auswahl der Daten werden diese vorverarbeitet. Bei der Vorverarbeitung werden die Filmobjekte und Benutzerprofile erstellt. Die Filmobjekte bestehen aus folgenden Elementen:

- der Datenbank **ID**,
- dem Titel,
- dem Budget,
- dem Herkunftsland,
- dem Genre, welchem der Film beim Experiment zugeordnet war, und
- den Genres der **IMDb**

In dem Benutzerprofil sind gespeichert:

- die Datenbank **ID**,
- das Alter,

- das Geschlecht,
- das Konsumverhalten,
- die Bewertungen der Kategorien,
- die Bewertungen der Filme des ersten Experiments und
- die Bewertungen der Filme des zweiten Experiments.

Ebenfalls ist in dem Benutzerprofil eine sortierte Liste der ähnlichen Benutzer gespeichert. Diese wird bei der Transformation generiert.

3.4.3. Transformation

Nach der Vorverarbeitung werden bei der Transformation zum einen die Listen der ähnlichen Benutzer erstellt, zum anderen werden die Cluster der Filme gebildet. Zum Erstellen der Listen ähnlicher Benutzer musste eine Distanzfunktion zwischen zwei Benutzern definiert werden. Diese betrachtet die Unterschiede in den abgegebenen Bewertungen und ist für zwei Benutzer (u_a, u_b) wie folgt definiert:

$$\text{Distanz}(u_a, u_b) = \frac{\sum_{i=1}^n |\text{Bewertung}(u_a, m_i) - \text{Bewertung}(u_b, m_i)|}{n}$$

Die Methode: $\text{Bewertung}(u, m)$ liefert die tatsächliche Bewertung eines Benutzers (u) für einen bestimmten Film (m). n entspricht der Anzahl der Filmbewertungen. Da im Kontext dieser Arbeit nur Benutzer betrachtet werden, die alle Filme bewertet haben, hat n immer den Wert: 50 (alle Filme des ersten Experiments). Aus dem gleichen Grund existieren auch keine Bewertungen von einem Benutzer für einen Film, der von dem anderen Benutzer nicht bewertet wurde. Somit muss hierfür keine Logik entwickelt werden. Ein möglicher Ansatz wäre, die maximale Distanz der Benutzer von 6 anzunehmen.

Die Distanz zwischen zwei Benutzern wird von der Methode „k-nearest“ verwendet und ist wie folgt implementiert:

```
1 private int calculateDist(User u, User ux) {
2     int dist = 0;
3     Map<Integer, MovieRating> uxr = ux.getOldRatings();
4     for (MovieRating mr : u.getOldRatings().values()) {
5         dist += Math.abs(mr.getValue()
6             - uxr.get(mr.getMovieID()).getValue());
7     }
```

```
8 return dist;  
9 }
```

Listing 3.5: Methode zum Berechnen der Distanz zwischen zwei Benutzern

In dieser Methode wird zunächst der Abstand auf Null gesetzt. Danach werden die Filmbewertungen des Benutzers u durchlaufen und mit den Bewertungen von Benutzer ux verglichen. Am Ende wird die kumulierte Summe der Einzeldistanzen zurückgeliefert. Diese entspricht der Gesamtdistanz zwischen den beiden Benutzern.

Für die Bildung der Cluster werden die Genres der Filme verwendet, welche diesen bei der **IMDb** zugeordnet sind. Somit ist zum Beispiel der Film „Guardians of the Galaxy“ mit den Genres: Action, Abenteuer und Science-Fiction dem Cluster mit genau diesen Genres zugeordnet. Ebenfalls wurde für die Cluster eine Gewichtung eingeführt. Cluster mit einer höheren Anzahl an Genres sollen stärker gewichtet werden. Daher wurde die Anzahl an Genres als Gewichtung eingeführt:

$$\text{Gewichtung}_{\text{Cluster}} = \text{Anzahl}_{\text{Genres}} \quad (3.1)$$

Für ein Cluster wird für jeden Benutzer eine Bewertung berechnet. Diese berechnet sich aus der durchschnittlichen Bewertung der Filme des Clusters:

$$\text{Clusterbewertung}(u, c) = \frac{\sum_{i=1}^n \text{Bewertung}(u, c_{m_i})}{n}$$

Dabei entsprechen: n = Anzahl der Filme des Clusters, c_{m_i} = der i . Film des Cluster, u = der Benutzer, für den die Clusterberechnung durchgeführt wird, und c = das Cluster, für welches die Berechnung durchgeführt wird.

Nach der Transformation können die unterschiedlichen Methoden des Dataminings verwendet werden, um Empfehlungen zu generieren.

3.4.4. Datamining

Im Kontext dieser Arbeit wurden unterschiedliche Methoden zur Generierung von Bewertungen verwendet. Berücksichtigt werden:

- die Filmbewertungen,
- die Kategoriebewertungen und
- die Bewertungen ähnlicher Benutzer.

Verwendung der Filmbewertungen

Für der Berechnung der Bewertungen mithilfe der Filmbewertungen werden die in Kapitel 3.4.3 beschriebenen Cluster verwendet. Alle Cluster, die mit dem Film mindestens eine Kategorie gemeinsam haben, werden für die Berechnung der Bewertung verwendet. Zunächst muss die Abhängigkeit berechnet werden. Diese stellt dar, wie stark die Clusterbewertung die Gesamtbewertung beeinflusst. Sie berechnet sich wie folgt:

$$\text{Abhängigkeit} = \frac{\text{Kategorien}_{\text{Film}} \cap \text{Kategorien}_{\text{Cluster}}}{\text{Kategorien}_{\text{Cluster}}} \quad (3.2)$$

Hierbei wird die Schnitt- und Vereinigungsmenge der Kategorien des Films und des Clusters betrachtet. Die Abhängigkeit ist das Ergebnis der Division von der Schnittmenge durch die Vereinigungsmenge. Wird zum Beispiel ein Film mit den Kategorien: „Sport, Aktion, Liebe“ und ein Cluster mit den Kategorien: „Sport, Aktion, Biografie“ betrachtet, berechnet sich die Abhängigkeit wie folgt:

$$\begin{aligned} \text{Abhängigkeit} &= \frac{\text{Kategorien}_{\text{Film}} \cap \text{Kategorien}_{\text{Cluster}}}{\text{Kategorien}_{\text{Cluster}}} \\ &= \frac{\text{Sport, Aktion, Liebe} \cap \text{Sport, Aktion, Biografie}}{\text{Sport, Aktion, Biografie}} \\ &= \frac{\text{Sport, Aktion}}{\text{Sport, Aktion, Biografie}} = \frac{2}{3} \end{aligned} \quad (3.3)$$

Somit ergibt sich unter Einbeziehung der Gewichtung und Abhängigkeit folgende Gesamtformel für die Berechnung einer Bewertung:

$$\text{Filmbewertung}(\text{Film}_x, u) = \frac{\sum_{i=1}^n \text{Clusterbewertung}_{u_i} * \text{Abhängigkeit}_{u_i}^2 * \text{Gewichtung}_i}{\sum_{i=1}^n \text{Abhängigkeit}_i^2 * \text{Gewichtung}_i}$$

Dabei entsprechen: Film_x = dem Film, für den die Bewertung durchgeführt werden soll, u = dem Benutzer, $\text{Clusterbewertung}_i$, Abhängigkeit_i und Gewichtung_i = Der Bewertung, Abhängigkeit und Gewichtung des jeweiligen Clusters. Um geringe Abhängigkeiten weiter abzuschwächen, werden diese quadriert.

Verwendung der Kategorien

Bei der zweiten im Experiment entwickelten Methode wird die Filmbewertung anhand der Kategorien berechnet. Hierfür werden die tatsächlichen Bewertungen der Kategorien eines Filmes betrachtet und deren Mittelwert gebildet:

$$\text{Filmbewertung}(\text{Film}_x, u) = \frac{\sum_{i=1}^n \text{Kategoriebewertung}(c_i, u)}{n}$$

Dabei entsprechen: u = dem Benutzer, Film_x = dem Film, für den die Bewertung berechnet werden soll, n = der Anzahl von Kategorien für den Film, c_i = der i . Kategorie des Films und $\text{Kategoriebewertung}(c_i, u)$ = der tatsächlichen Kategoriebewertung des Benutzers. Dies wurde wie folgt implementiert:

```
1 protected double calculateRating(User u, Movie m) {  
2     double rating = 0.0;  
3     List<Genre> genreList = genresForMovie.get(m);  
4     for (Genre g : genreList) {  
5         rating += u.getGenreRatings().get(g);  
6     }  
7     return Math.round(rating / genreList.size());  
8 }
```

Listing 3.6: Methode zum Berechnen des Ratings anhand der Kategoriebewertungen

In dieser Methode werden zunächst die Genres des Filmes in einer Liste gespeichert. Anschließend wird über diese Liste iteriert, und es werden die Bewertungen, für das jeweilige Genre von dem Benutzer aufsummiert. Die Summe wird durch die Anzahl der Genres geteilt, gerundet und schließlich zurückgeliefert.

Verwendung der Bewertungen ähnlicher Benutzer

Bei der Methode des **CF** werden die Bewertungen der Benutzer verwendet, die in ihrem Bewertungsverhalten dem Benutzer am ähnlichsten sind. Diese Methode wird im Kontext dieser Arbeit als „k-nearest“ bezeichnet. Sie verwendet die Listen ähnlicher Benutzer, die in Kapitel 3.4.3 beschrieben werden. Zur Berechnung der Bewertung werden die „k“ ähnlichsten Benutzer betrachtet und es wird ein Mittelwert aus deren tatsächlichen Bewertungen gebildet. Somit wird die Bewertung wie folgt berechnet:

$$\text{Filmbewertung}(\text{Film}_x, u) = \frac{\sum_{i=1}^k \text{Filmbewertung}_{\text{tatsächlich}}(\text{Film}_x, u_i)}{k}$$

Die Resultate der Methoden werden nach den in Kapitel 2.1.6 beschriebenen Metriken bewertet. Die Interpretation der Ergebnisse wird in Kapitel 4 beschrieben.

3.5. Fazit

Zu Beginn dieses Kapitels wurden die Ziele des Experiments erläutert. Das Ziele sind einen Trainings- und Testdatensatz zu erstellen und ein RS zu entwickeln. Danach wurde das erste Experiment beschrieben. Es wurde erläutert, dass mithilfe des ersten Experiments der Trainingsdatensatz erstellt wurde. Daraufhin wurde das zweite Experiment beschrieben. Der Testdatensatz ist das Resultat des zweiten Experiments. Abschließend wurde die Umsetzung des RS beschrieben. Es wurden die einzelnen Schritte des KDD erläutert und die unterschiedlichen Methoden genauer dargestellt.

Mit dem entwickelten RS unter Verwendung des Trainings- und Testdatensatzes ist es erforderlich, Tests durchzuführen und diese zu evaluieren. Die Ergebnisse und deren Evaluation werden im nächsten Kapitel dargestellt.

4. Auswertung

In diesem Kapitel werden zunächst die Informationen der Teilnehmer des ersten und zweiten Votings genauer betrachtet und analysiert. Anschließend werden die Ergebnisse der im Rahmen des zweiten Experiments entwickelten Methoden genauer betrachtet. Gefolgt von einer Betrachtung der verbesserten Version. Daraufhin werden die Ergebnisse typischer Methoden analysiert. Abschließend wird ein Vergleich aller Methoden durchgeführt.

4.1. Auswertung der Informationen über die Teilnehmer

In diesem Kapitel werden die Teilnehmer des ersten und zweiten Experiments betrachtet. Berücksichtigung finden Alter, Geschlecht und Konsumverhalten.

4.1.1. Teilnehmer des ersten Experiments

Alle Teilnehmer

Folgende Informationen konnten über die Teilnehmer gewonnen werden:

- 323 Personen haben teilgenommen,
- 222 Personen haben alle Fragen zur Person beantwortet,
- 87 davon sind weiblich und 135 männlich,
- 26 sind unter 21,
- 159 sind zwischen 21 - 30,
- 23 sind zwischen 31 - 40,
- 11 sind zwischen 41 - 50,
- 1 ist zwischen 51 - 60 und
- 2 sind zwischen 61 - 70 Jahre alt

4. Auswertung

Abbildung 4.1 zeigt eine Übersicht über die Teilnehmer des ersten Votings bezogen auf Geschlecht, Alter und Konsumverhalten. Es werden die Teilnehmer berücksichtigt, welche alle Fragen zur Person beantwortet haben. Es werden auch die Teilnehmer berücksichtigt, welche ausschließlich die Fragen zur Person beantwortet haben und/ oder einen Großteil der anderen Fragen nicht beantwortet haben.

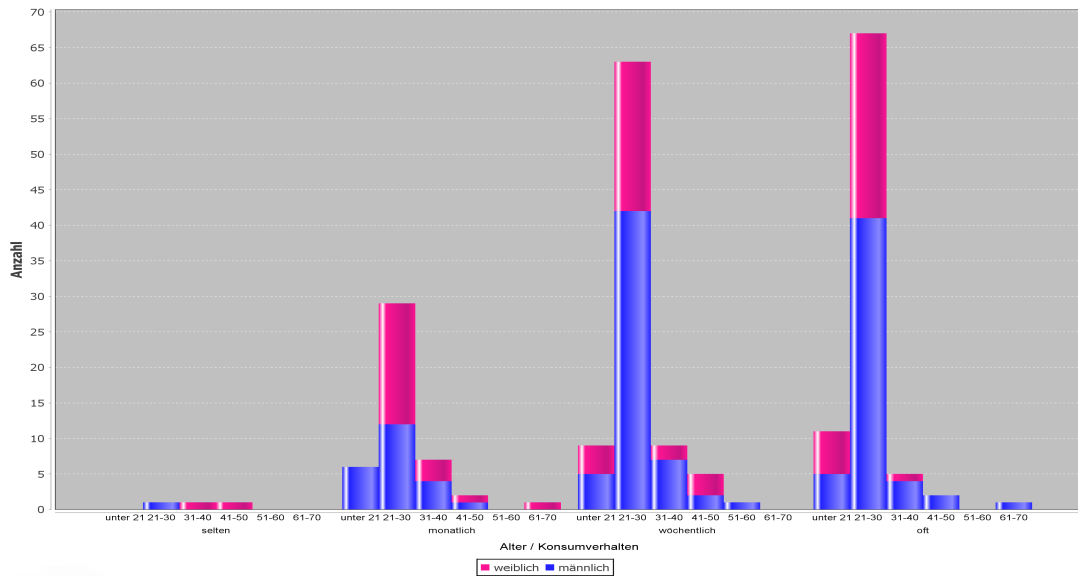


Abbildung 4.1.: Informationen der Teilnehmer (durch Selbstauskunft gewonnen), der weibliche Anteil der Teilnehmer wird pink, der Anteil der männlichen Teilnehmer blau angezeigt. Ebenfalls sind die Ergebnisse nach Konsumverhalten gruppiert und in die Altersgruppen aufgesplittet

Die meisten Teilnehmer sind zwischen 21 und 30 Jahren alt. Weil hauptsächlich Studenten für die Teilnahme akquiriert und ein großer Teil von ihnen zwischen 21 und 30 Jahren alt ist, war dieses Ergebnis zu erwarten. Eine Methode zur Generierung der Filmbewertungen unter Verwendung des Alters ist nicht sinnvoll. Die Verteilung des Alters ist sehr ungleichmäßig und somit nicht aussagekräftig genug.

Viele der Teilnehmer schauen Filme wöchentlich oder öfter. Einige sehen sich monatlich Filme an und lediglich eine Person schaut seltener Filme. Der Anteil der männlichen Teilnehmer ist größer.

Teilnehmer, welche das Voting komplett mitgemacht haben

Werden nur die Teilnehmer berücksichtigt, welche auch alle Fragen beantwortet haben, ergeben sich folgende Informationen:

- 91 Teilnehmer haben das Voting zuende durchgeführt,
- 32 davon sind weiblich und 59 männlich,
- 11 sind unter 21,
- 56 sind zwischen 21 - 30,
- 15 sind zwischen 31 - 40,
- 6 sind zwischen 41 - 50,
- 1 ist zwischen 51 - 60 und
- 2 sind zwischen 61 - 70 Jahre alt.

Abbildung 4.2 zeigt eine Übersicht über die Teilnehmer des ersten Votings, welche das Voting bis zu Ende durchgeführt haben, bezogen auf Geschlecht, Alter und Konsumverhalten.

4. Auswertung

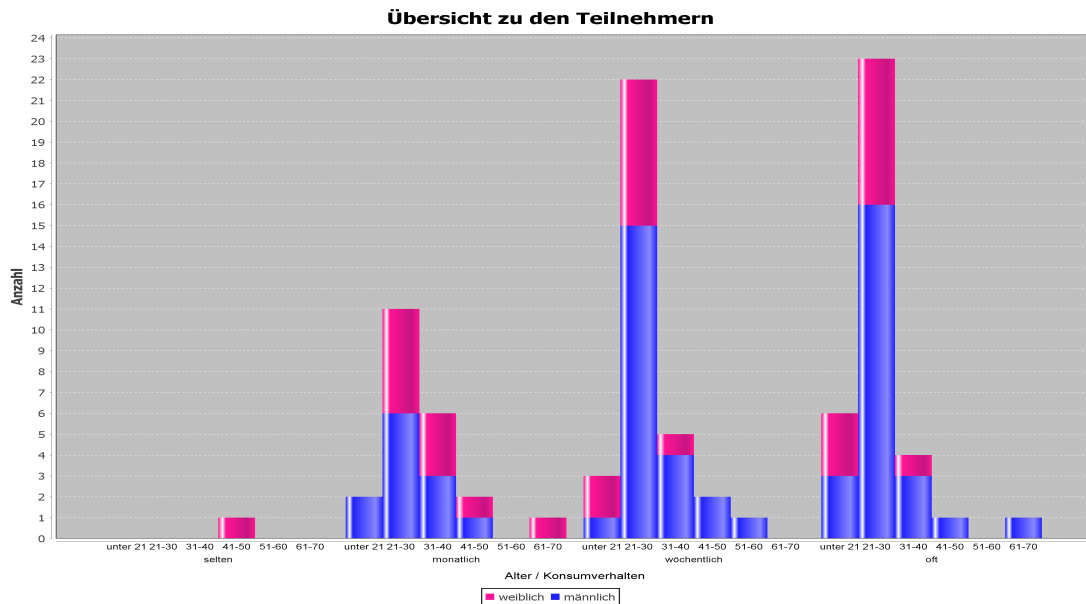


Abbildung 4.2.: Informationen der Teilnehmer (durch Selbstauskunft gewonnen), der weibliche Anteil der Teilnehmer wird pink, der Anteil der männlichen Teilnehmer blau angezeigt. Ebenfalls sind die Ergebnisse nach Konsumverhalten gruppiert und in die Altersgruppen aufgesplittet

Von den anfänglichen 323 Teilnehmern haben lediglich 91 Personen das erste Voting bis zum Ende durchgeführt. Dies ist unter anderem mit dem hohen Aufwand begründet, welcher für die Teilnehmer anfällt. So dauert das Betrachten der Trailer alleine schon mehr als zwei Stunden.

Der Anteil der männlichen Teilnehmer, welche das Voting bis zum Ende durchgeführt haben ist größer. Die Verteilung ist vergleichbar mit der der ursprünglichen Teilnehmer. Die meisten sind zwischen 21 und 30 Jahren alt. Der Anteil der männlichen Teilnehmer ist doppelt so groß wie der der weiblichen. Die meisten schauen Filme wöchentlich oder öfter.

4.1.2. Teilnehmer des zweiten Experiments

Für das zweite Experiment wurden alle Teilnehmer des ersten Votings angeschrieben, welche dieses bis zum Ende durchgeführt hatten. Folgende Informationen konnten gesammelt werden:

- 54 Personen haben am zweiten Experiment teilgenommen,
- 42 Teilnehmer haben das zweite Voting zuende durchgeführt,

4. Auswertung

- 18 davon sind weiblich und 24 männlich,
- 5 sind unter 21,
- 25 sind zwischen 21 - 30,
- 4 sind zwischen 31 - 40,
- 5 sind zwischen 41 - 50,
- 1 ist zwischen 51 - 60 und
- 2 sind zwischen 61 - 70 Jahre alt

Abbildung 4.3 zeigt eine Übersicht über die Teilnehmer des zweiten Experiments bezogen auf Geschlecht, Alter und Konsumverhalten.

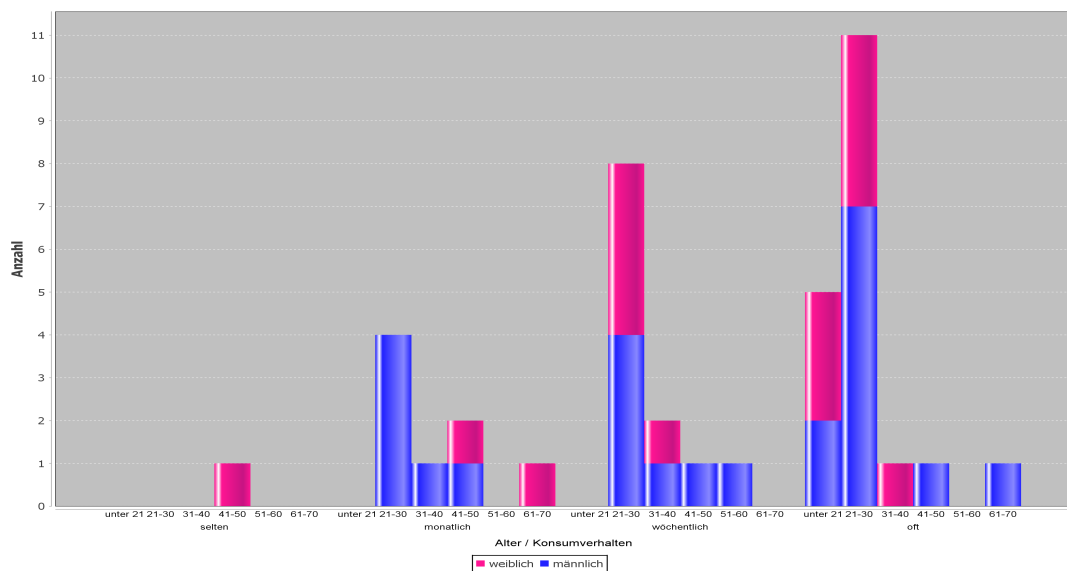


Abbildung 4.3.: Informationen der Teilnehmer des zweiten Experiments (durch Selbstauskunft gewonnen), der weibliche Anteil der Teilnehmer wird pink, der Anteil der männlichen Teilnehmer blau angezeigt. Ebenfalls sind die Ergebnisse nach Konsumverhalten gruppiert und in die Altersgruppen aufgesplittet

Von den 91 Teilnehmern, welche beim ersten Experiment alle Fragen beantwortet haben, haben 54 am zweiten Experiment teilgenommen. Von diesen haben 42 das zweite Voting bis

4. Auswertung

zum Ende durchgeführt. Die Verteilung ist mit der des ersten Experiments vergleichbar. Die meisten Teilnehmer sind zwischen 21 und 30 Jahren alt und schauen wöchentlich oder öfter Filme. Wobei der Anteil der Personen größer geworden ist, welche häufiger als wöchentlich Filme schauen. Dies deutet darauf hin, dass ein stärkeres Interesse an Filmen die Motivation zur Durchführung des Votings erhöhte.

4.2. Auswertung der Kategoriebewertungen

In diesem Kapitel werden die Kategoriebewertungen genauer betrachtet. Abbildung 4.4 zeigt die durchschnittlichen Bewertungen der Kategorien beim ersten Experiment.

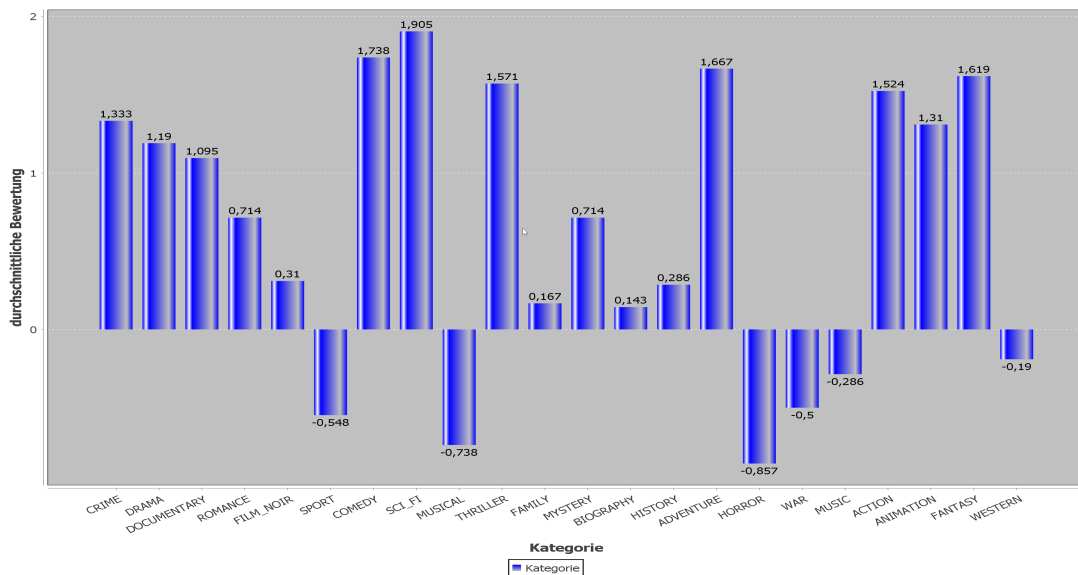


Abbildung 4.4.: Vergleich der durchschnittlichen Bewertungen der Kategorien

Die meisten Kategorien wurden positiv bewertet. Sechs der 22 Kategorien wurden negativ bewertet. Für die fünf Kategorien, zu denen die Teilnehmer zehn Filme bewerteten, ergaben sich folgende durchschnittliche Bewertungen:

- Action: 1,52
- Biografie: 0,14
- Romantik: 0,71

4. Auswertung

- Science-Fiction: 1,91
- Sport: -0,55

Die Kategorie Science-Fiction wurde mit 1,91 am besten bewertet. Action wurde ebenfalls sehr gut bewertet mit 1,52. Sport wurde negativ bewertet. Somit ist davon auszugehen, dass Filme der Kategorie Action oder Science-Fiction besser bewertet werden, als Filme der Kategorie Sport oder Romantik.

4.3. Auswertung der Filmbewertungen

In diesem Kapitel werden die Bewertungen der Filme genauer analysiert. Dabei werden die Kategorie, das Herkunftsland und das Budget genauer betrachtet.

4.3.1. Filmbewertungen des ersten Experiments

Betrachtung der Kategorie

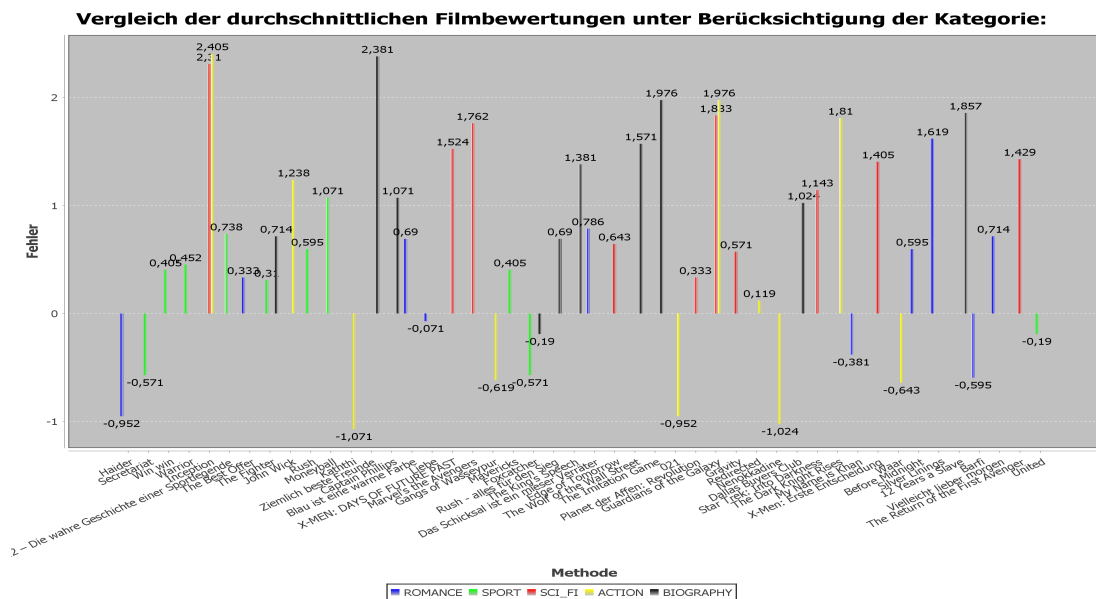


Abbildung 4.5.: Vergleich der durchschnittlichen Bewertungen unter Berücksichtigung der Kategorien

4. Auswertung

Die Filmbewertungen unter Berücksichtigung der Kategorien lassen kein eindeutiges Muster erkennen. Lediglich Science Fiction hat ausschließlich positive Bewertungen. Biografien sind ebenfalls recht gut bewertet und haben nur eine Bewertung im negativen Bereich. Action hat fünf positive und negative Bewertungen. Es gibt insgesamt mehr positive als negative Bewertungen. Dies war zu erwarten, da die am besten bewerteten Filme in der **IMDb** ausgesucht worden waren.

Betrachtung des Herkunftslandes

In diesem Kapitel werden die Filmbewertungen unter Berücksichtigung des Herkunftslandes genauer analysiert.

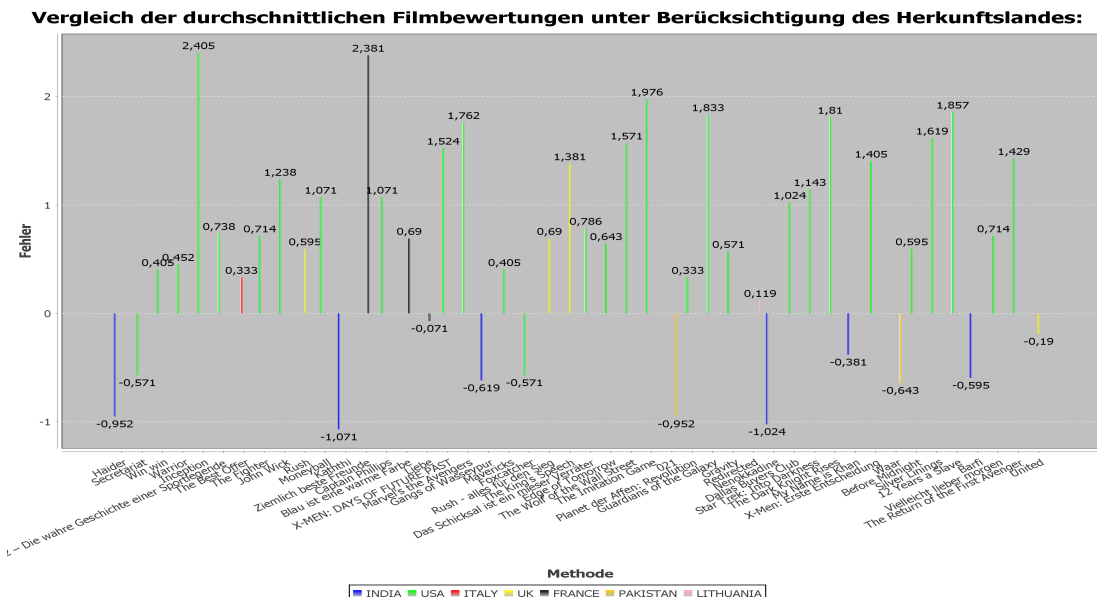


Abbildung 4.6.: Vergleich der durchschnittlichen Bewertungen unter Berücksichtigung des Herkunftslandes

Bei der Betrachtung der Herkunftslandes fällt auf, dass sich dieses massiv auf die Bewertungen auswirkt. Somit gibt es lediglich eine negative Bewertung eines amerikanischen Films. Dahingegen sind alle indischen Filme negativ bewertet worden. Die restlichen Länder haben vier oder weniger Bewertungen. Die geringe Zahl der Bewertungen führt dazu, dass zu erkennende Tendenzen zufällig entstanden sein können. Die beiden französischen Filme sowie der eine italienische Film sind positiv bewertet worden. Der größte Teil der Filme aus dem

4. Auswertung

Vereinigten Königreich (drei der vier Filme) sind positiv bewertet worden. Die beiden Pakistanschen Filme wurden negativ bewertet.

Somit wird deutlich, dass sich das Herkunftsland signifikant auf die Bewertung auswirkt. Filme aus Amerika werden besonders gut bewertet. Auch Filme aus Europa haben eine gute Bewertung. Filme aus Pakstian oder Indien hingegen wurden sehr schlecht bewertet.

Betrachtung des Budgets

In diesem Kapitel werden die Filmbewertung unter Berücksichtigung des Budgets genauer betrachtet.

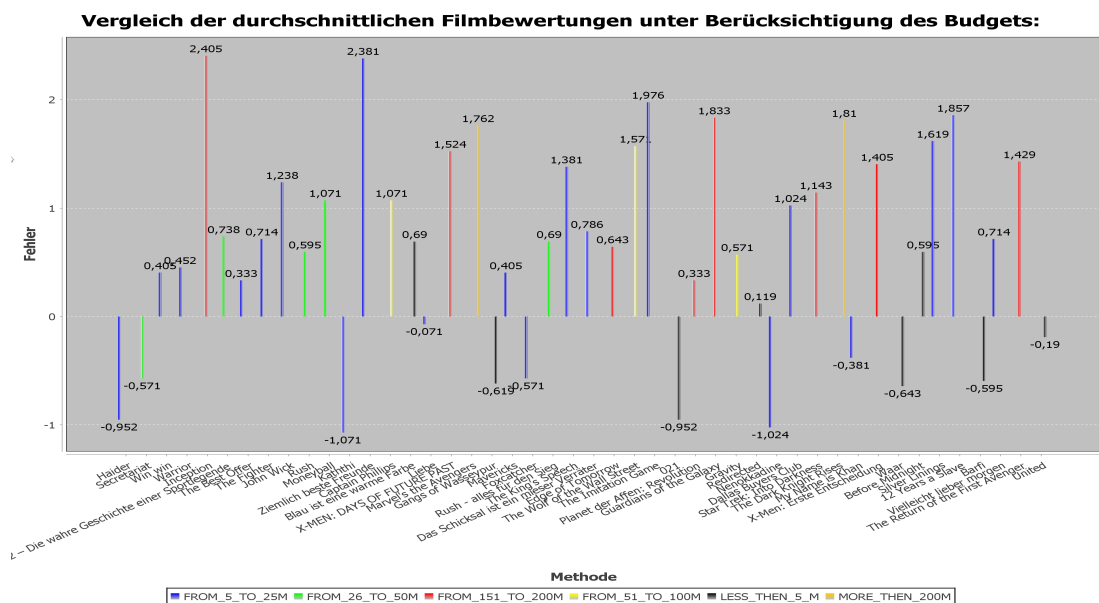


Abbildung 4.7.: Vergleich der durchschnittlichen Bewertungen unter Berücksichtigung des Budgets

Bei der Betrachtung des Budgets fällt auf, dass alle Filme, die ein Budget von mehr als 50 Millionen Dollar hatten, positiv bewertet wurden. Die beiden Filme, die mehr als 200 Millionen Dollar Budget hatten, wurden sehr gut bewertet (beide ca. 1.8). Von den Filmen, welche zwischen 151 und 200 Millionen Dollar budget hatten, sind alle positiv bewertet worden. Lediglich zwei der Filme wurden schlechter als 1 im Durchschnitt bewertet. Die drei Filme mit einem Budget 51 und 100 Millionen haben alle positive Bewertungen. Nur einer der fünf Filme, bei denen das Budget zwischen 26 und 50 Millionen Dollar lag, hatte eine negative Bewertung.

Die meisten Filme hatten ein Budget zwischen 5 und 25 Millionen Dollar. Die meisten von diesen hatten eine positive Bewertung. Sechs der 20 Filme hatten eine negative Bewertung. Der größte Teil der Filme mit einem Budget von weniger als fünf Millionen Dollar wurde negativ bewertet.

Zusammenfassend lässt sich sagen, dass Filme mit einem großen Budget im Durchschnitt auch besser bewertet werden. So kann davon ausgegangen werden, dass Filme mit einem Budget von mehr als 50 Millionen Dollar besser bewertet werden und Filme mit einem Budget von unter 5 Millionen schlecht bewertet werden.

Fazit der Filmbetrachtungen des ersten Experiments

Werden Kategorie, Budget und Herkunftsland zusammen betrachtet, wird deutlich, dass nur die Betrachtung der Kategorie nicht ausreichend ist. Im Bereich der Action-Filme zum Beispiel sind die Bewertungen sehr unterschiedlich. Es ist davon auszugehen, dass diese mehr mit dem Herkunftsland und dem Budget, als mit der Kategorie selbst zutun haben. Daraus ist zu schließen, dass eine reine Betrachtung der Kategorien weniger effizient ist, als wenn Budget und Herkunftsland mit einbezogen werden.

4.3.2. Filmbewertungen des zweiten Experiments

In diesem Kapitel werden die Filmbewertungen des zweiten Experiments betrachtet. Dabei werden die Kategorie, das Herkunftsland und das Budget genauer analysiert. Eine Übersicht der Bewertungen unter Berücksichtigung des Herkunftslandes ist in Anhang [A.5](#). Die Abbildung [4.8](#) zeigt die Filmbewertungen mit der Betrachtung des Budgets.

4. Auswertung

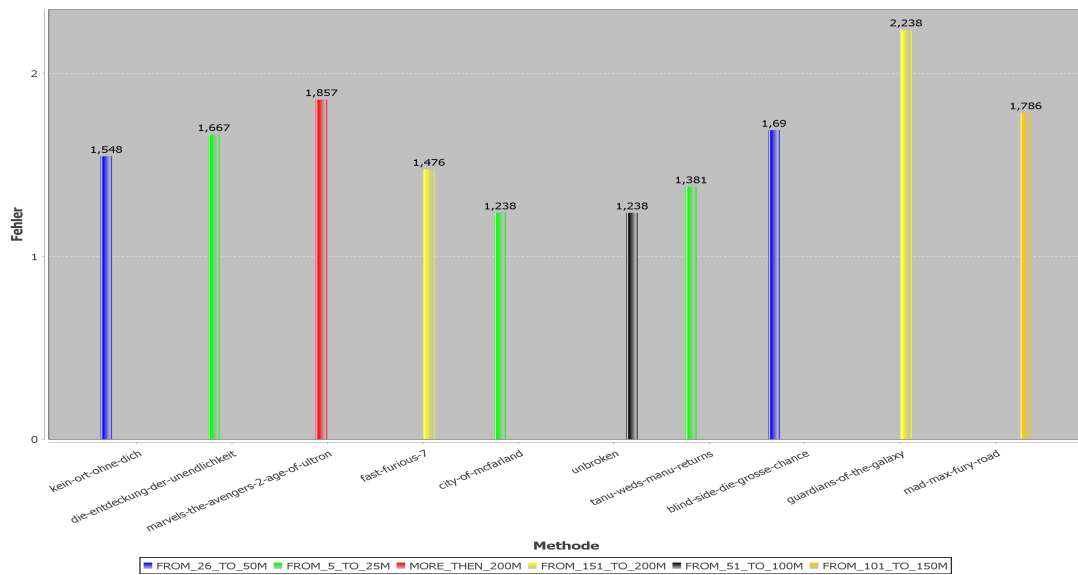


Abbildung 4.8.: Filmbewertungen des zweiten Experiments mit Betrachtung des Budgets

Die Filmbewertungen des zweiten Experiments sind alle positiv. Obwohl die Budgets der Filme sich unterscheiden, sind alle Filme sehr gut bewertet worden. Alle Filme sind im Durchschnitt besser als 1,2 bewertet worden. Beim zweiten Experiment wurden keine Filme mit einem Budget unter fünf Millionen Dollar verwendet.

Bei der Berücksichtigung des Herkunftslandes fällt auf, dass acht der zehn Filme in Amerika produziert wurden. Einer der Filme kommt aus den Vereinigten Königreichen und ein Film kommt aus Indien. Das die amerikanischen Filme positiv bewertet wurden, konnte erwartet werden. Das der einzige indische Film positiv bewertet wurde war hingegen überraschend. Die Filmbewertungen mit Berücksichtigung der Kategorie ergibt folgendes Bild:

4. Auswertung

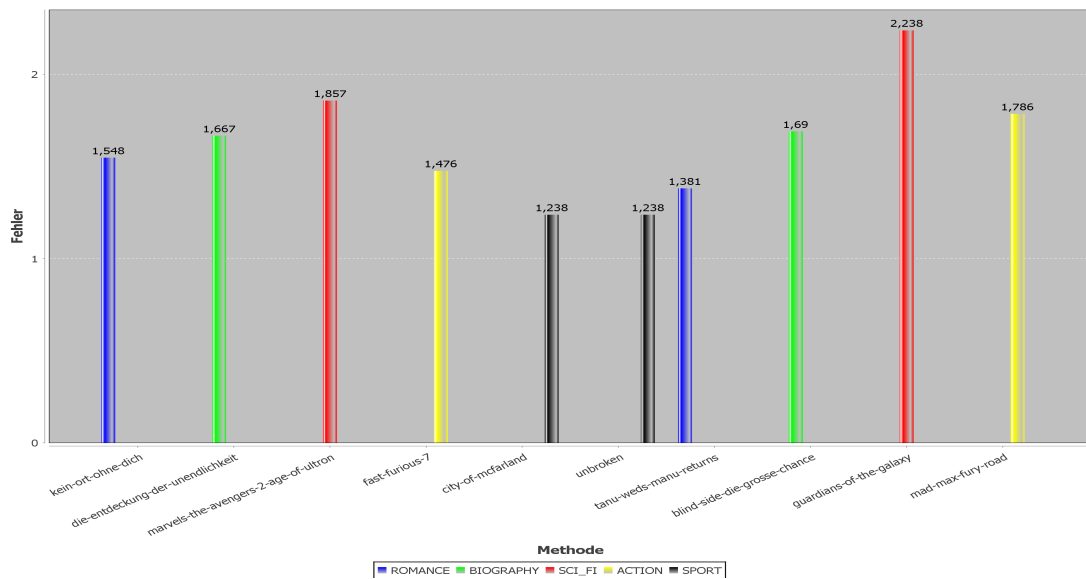


Abbildung 4.9.: Filmbewertungen des zweiten Experiments mit Betrachtung der Kategorie

Es ist zu erkennen, dass sich hier die Tendenzen der Kategoriebewertungen widerspiegeln. Die Filme der Kategorie Science-Fiction wurden am besten bewertet. Die beiden Sportfilme wurden am schlechtesten bewertet. Nur die Biografien waren unerwartet hoch bewertet worden. Das Budget dieser beiden Filme ist unter 50 Millionen Dollar. Somit ist davon auszugehen, dass die Themen besonders interessant sind und/ oder die Umsetzung dieser beiden Filme überdurchschnittlich ist.

4.3.3. Filme mehrerer Kategorien

In diesem Kapitel werden die fünf Filme betrachtet, die mehrfach bewertet wurden, da sie mehreren Kategorien zugeordnet waren. Abbildung 4.10 zeigt, wie diese Filme in den jeweiligen Kategorien bewertet wurden. Der Film „Guardians of the Galaxy“ wurde sowohl beim ersten, als auch beim zweiten Experiment bewertet.

4. Auswertung

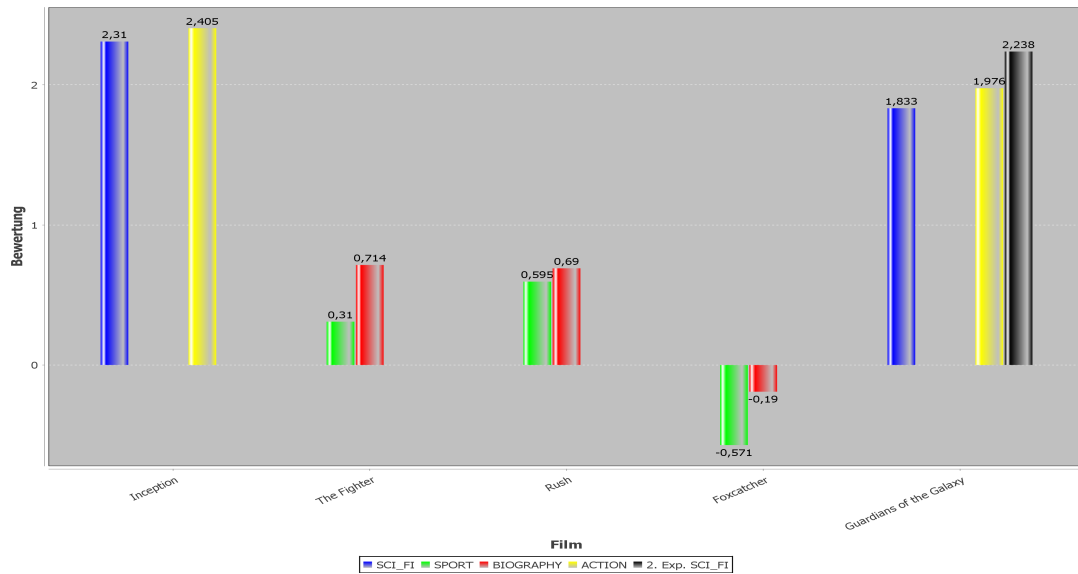


Abbildung 4.10.: Filmbewertungen des zweiten Experiments mit Betrachtung der Kategorien

Die beiden Filme „Inception“ und „Guardians of the Galaxy“ wurden in der Kategorie Action besser bewertet, als in der Kategorie Science-Fiction. Das Ergebnis war so nicht zu erwarten, da die Kategorie Science-Fiction deutlich besser bewertet worden war. Die Unterschiede sind eher gering. Um den genauen Grund für die unterschiedlichen Bewertungen zu erfahren, müsste eine Benutzerbefragung durchgeführt werden.

Bei den drei Filmen, die sowohl der Kategorie Sport, als auch der Kategorie Biografie zugeordnet wurden, sind die Ergebnisse den Kategoriebewertungen entsprechend. Die Bewertungen bei der Kategorie Sport waren negativer, bei „Foxcatcher“ (0,38) und „The Fighter“ sogar sehr deutlich negativer.

4.4. Auswertung der Methoden

Um die Qualität der Methoden zu vergleichen, werden die in Kapitel 2.1.6 dargestellten Metriken verwendet.

4.4.1. Auswertung des ersten Ansatzes

Beim ersten Ansatz wurden alle der ImDB Kategorien, welche den Filmen zugeordnet sind, für die Vorhersage verwendet. Dieser Ansatz wurde im Rahmen eines Projekts erarbeitet. Es

4. Auswertung

sind zwei unterschiedliche Methoden entwickelt worden. Bei der ersten Methode werden die Filmbewertungen anhand der bereits abgegebenen Filmbewertungen berechnet. Bei der zweiten Methode werden die Filmbewertungen anhand der Kategoriebewertungen berechnet. Das Diagramm 4.11 zeigt eine Übersicht über die Ergebnisse der Methode.

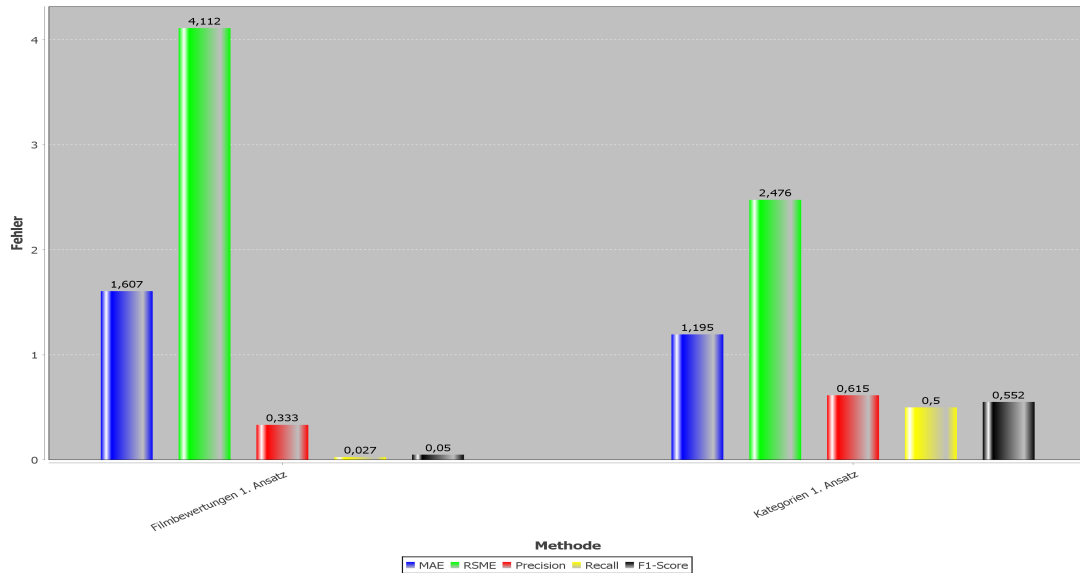


Abbildung 4.11.: Ergebnisse des ersten Ansatzes

Bei der ersten Methode (Filmbewertungen) ist der durchschnittliche Fehler (MAE) 1.61 und der durchschnittliche quadrierte Fehler (RSME) 4.11. Der Wert für precision ist 0.33, was bedeutet, dass nur jeder dritte vorgeschlagene Film auch korrekt ist. Die Werte für recall und daraus resultierend auch für F1-Score sind bei 0.03 und 0.05. Dies bedeutet, dass fast keine der Filme, welche den Benutzern gefallen haben, diesen auch vorgeschlagen wurden.

Bei der zweiten Methode (Kategorien) ist der durchschnittliche Fehler (MAE) 1.12 und der durchschnittliche quadrierte Fehler (RSME) 2.48. Der Wert für precision ist 0.62, was bedeutet, dass mehr als die Hälfte der vorgeschlagenen Filme korrekt sind. Die Werte für recall und daraus resultierend auch für F1-Score sind bei 0.50 und 0.55. Dies bedeutet, dass genau die Hälfte der Filme, welche den Benutzern gefallen haben, diesen auch vorgeschlagen wurden. Werden die beiden Methoden verglichen wird deutlich, dass die erste Methode schlechtere Ergebnisse erzielt als die zweite. Die Betrachtung aller Kategorien der **IMDb** und deren Bewertungen aus der Selbstauskunft führt zu ausreichend verwertbaren Ergebnissen.

Unter Berücksichtigung der Erkenntnisse aus Kapitel 4.3 können die im Rahmen des zweiten Experiments entwickelten Methoden verbessert werden.

4.4.2. Auswertung des zweiten Ansatzes

Im zweiten Experiment wurden die Methoden auf folgende Weise verbessert:

- Reduzierung der Kategorien auf eine Hauptkategorie,
- Einbeziehung des Herkunftslandes und
- feste Bewertungen für bestimmte Herkunftsländer.

Diese Änderungen führten zu folgenden Ergebnissen:

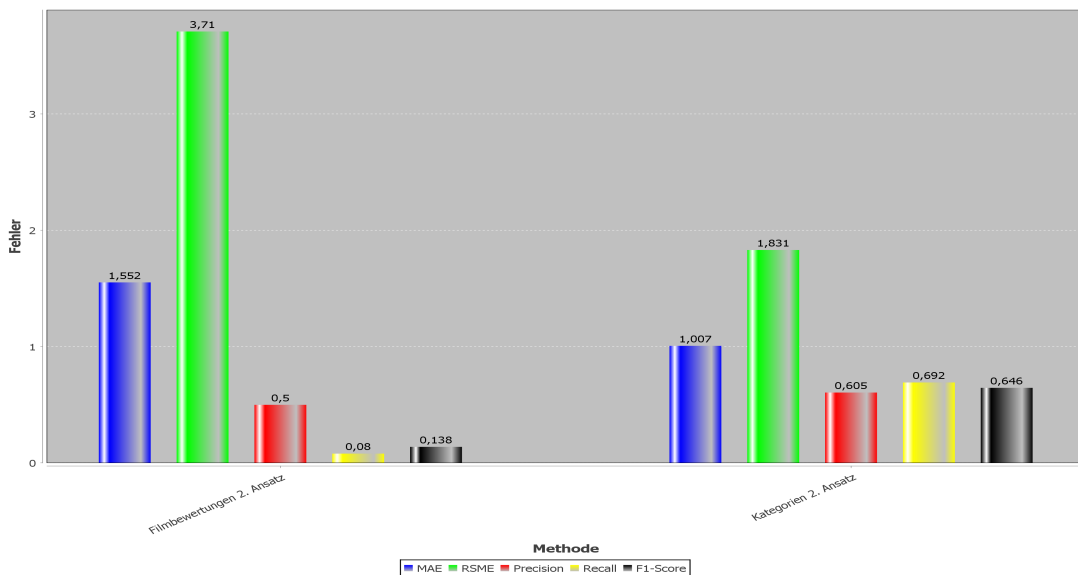


Abbildung 4.12.: Ergebnisse des zweiten Ansatzes

Bei der ersten Methode (Filmbewertungen) ist der durchschnittliche Fehler (MAE) 1.55 und der durchschnittliche quadrierte Fehler (RSME) 3.71. Der Wert für precision ist 0.50, was bedeutet, dass die Hälfte der vorgeschlagenen Filme auch korrekt ist. Die Werte für recall und daraus resultierend auch für F1-Score sind bei 0.08 und 0.14. Dies bedeutet, dass fast keine der Filme, welche den Benutzern gefallen haben, diesen auch vorgeschlagen wurden.

Bei der zweiten Methode (Kategorien) ist der durchschnittliche Fehler (MAE) 1.01 und der

durchschnittliche quadrierte Fehler (RSME) 1.83 . Der Wert für precision ist 0.61, was bedeutet, dass mehr als die Hälfte der vorgeschlagenen Filme korrekt sind. Die Werte für recall und daraus resultierend auch für F1-Score sind bei 0.70 und 0.65. Dies bedeutet, dass mehr als die Hälfte der Filme, welche den Benutzern gefallen haben, diesen auch vorgeschlagen wurden. Die Verwendung weniger Kategorien und die Einbeziehung des Herkunftslandes führten zu einer geringen Verbesserung der Ergebnisse. Dennoch ist Wert für recall mit 0.08 viel zu niedrig, da weniger als zehn Prozent der Filme vorgeschlagen wurden, welche den Nutzern gefallen. Bei der zweiten Methode hingegen verbesserten sich die meisten Werte. Nur die precision wurde geringfügig schlechter. Dafür verbesserte sich recall um fast 20 Prozent von 0.5 auf 0.69. Somit werden zwei von drei "richtigen" Filmen vorgeschlagen. Die Änderungen des zweiten Ansatzes führten somit zu verbesserten Ergebnissen. Diese sind vor allem bei der zweiten Methode signifikant.

4.4.3. Auswertung von Collaborative Filtering mit k-nearest

Eine Möglichkeit des CF ist die Betrachtung der ähnlichsten Benutzer und derer Bewertungen. Die in dieser Arbeit verwendete Methode des CF vergleicht die Bewertungen der Benutzer vom ersten Voting. Die unterschiedlichen Methoden unterscheiden sich in der Menge der betrachteten Benutzer. So wird beim „k-nearest 1“ nur der ähnlichste Benutzer betrachtet und dessen Bewertung direkt übernommen. Beim „k-nearest 2“ werden die beiden ähnlichsten Benutzer betrachtet und ein Mittelwert derer Bewertungen gebildet. Bei „k-nearest 3“ die nächsten drei usw. Abbildung 4.13 zeigt eine Übersicht über die Ergebnisse der unterschiedlichen Varianten von „k-nearest“.

4. Auswertung

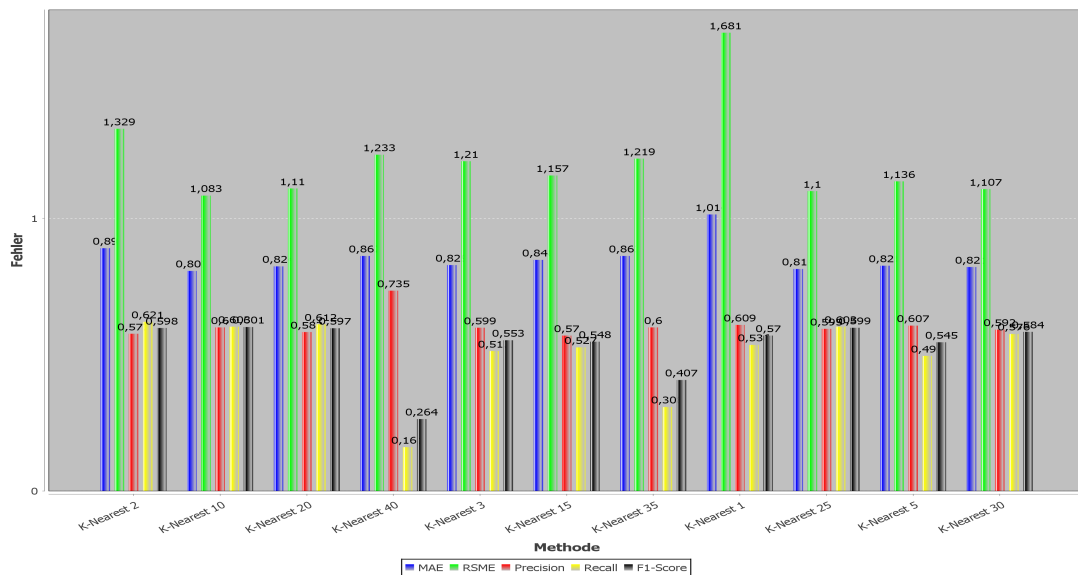


Abbildung 4.13.: Ergebnisse von CF mit k-nearest

Die Ergebnisse der Methode „k-nearest 1“ sind die schlechtesten. Es ist die einzige Methode, bei welcher der durchschnittliche Fehler über 1 ist. Der hohe Wert von 1,68 für RMSE deutet darauf hin, dass einige Berechnungen stark von der tatsächlichen Bewertung abweichen.

Werden nicht nur der ähnlichste sondern mehr Benutzer betrachtet, werden die Ergebnisse deutlich besser. Bis hin zu der Betrachtung von 10 ähnlichen Benutzern. Hier ist der quadrierte Fehler unter 1,1 und sowohl der Wert für recall als auch der für precision bei 0,6. Dies bedeutet, dass 60% der gefundenen Filme auch korrekt und nur 40% der gefundenen Filme nicht korrekt sind.

Werden mehr als 10 ähnliche Benutzer betrachtet, ändern sich die Werte nur sehr gering. Die Bewertungen werden langsam schlechter, bis bei 40 ähnlichen Benutzern sich der Wert für die precision um 13,5% verbessert auf Kosten des recall. Der Wert für recall fällt auf 0,16. Somit ist zwar mehr als $\frac{2}{3}$ der gefundenen Filme korrekt, es wird aber nur noch jeder achte Film gefunden.

Die gerade beschriebenen Ergebnisse, der in dieser Arbeit verwendeten Methoden für die Vorhersage, werden im nächsten Kapitel mit einander verglichen. Von den „k-nearest“-Methoden wird „k-nearest 10“ verwendet, da diese die besten Ergebnisse erreicht hat.

4.4.4. Vergleich mehrerer Methoden

Folgende Methoden wurden für den Vergleich verwendet, Methoden mit der Verwendung von:

- Kategoriebewertungen,
- Filmbewertungen,
- k-nearest 10 und
- immer 2 (Eine Methode, welche immer den Wert 2 als Bewertung zurückliefert).

Abbildung 4.14 zeigt den Vergleich dieser vier Methoden.

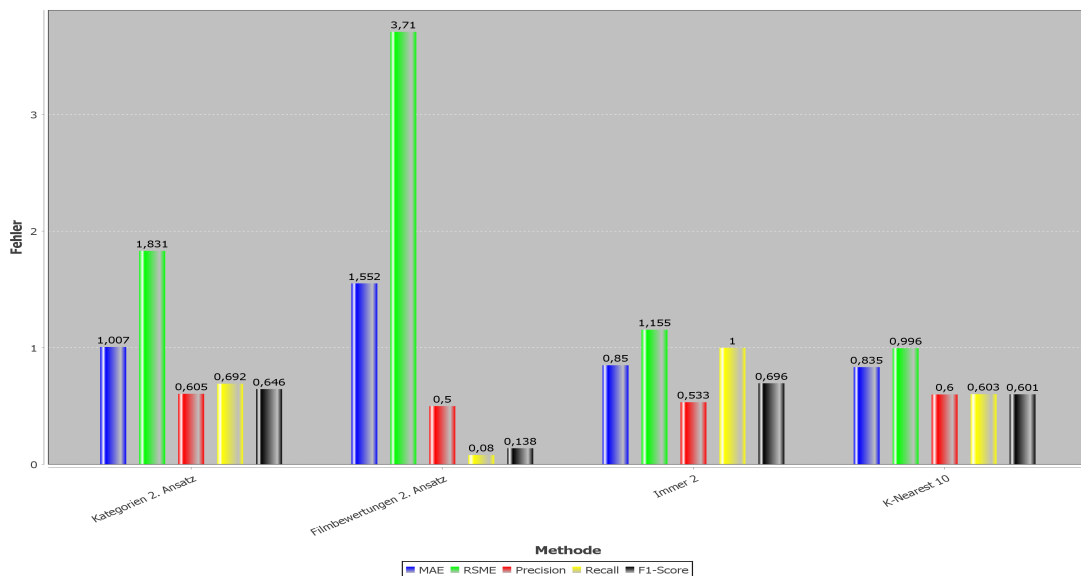


Abbildung 4.14.: Vergleich mehrerer Methoden

Bei den Kategorie- und Filmbewertungen wurden die Methoden des zweiten Ansatzes verwendet, da diese bessere Ergebnisse erzielten, als die des ersten Ansatzes.

Die Methode „Immer 2“ liefert immer eine Bewertung von 2 zurück. Daraus resultiert ein Wert von 1 für recall. Wenn alle Filme vorgeschlagen werden, werden auch alle richtigen Filme vorgeschlagen. Bei der Methode Kategoriebewertungen ist der Wert mit 0,7 ebenfalls recht hoch. Die Methode k-nearest kommt nur auf 0,6 und die Methode mit den Filmbewertungen nur auf sehr schwache 0,1.

Der Wert von precision mit 0,53 für „immer 2“ bedeutet, dass etwas mehr als die Hälfte der

Filme auch tatsächlich für gut befunden wurden. Dieser Wert ist nur minimal größer als der bei den Filmbewertungen. Bei der Kategoriebewertungen und der k-nearest Methode sind die Werte für precision mit 0,6 leicht besser.

Werden der durchschnittliche Fehler und der quadrierte durchschnittliche Fehler betrachtet, ist die Methode „k-nearest“ knapp am genauesten, dicht gefolgt von „immer 2“.

4.5. Auswertung des gesamten Experiments

Die Teilnehmer des Experiments sind hauptsächlich Studenten, die zwischen 21 und 30 Jahre alt sind. Da der große Anteil der Studenten aus dem Informatikbereich kommt, könnte dies eine Erklärung für das gute Abschneiden der Kategorie Science-Fiction sein. Somit kann die These aufgestellt werden, dass Informatikstudenten aufgrund der großen Technikaffinität auch eine Vorliebe für Science-Fiction haben.

Werden die Filmbewertungen betrachtet, fällt auf, dass neben den Kategorien auch Attribute, wie Budget und Herkunftsland eine wesentliche Rolle spielen. Somit scheint es ausgeschlossen, Vorhersagen nur anhand der Kategorien zu treffen.

Die Bewertungen der Filme des zweiten Experiments sind im Verhältnis zu den Bewertungen des ersten Experiments deutlich besser. Dies lässt darauf schließen, dass die Filme für das zweite Experiment ungünstig gewählt worden sind. Für ein aussagekräftigeres Experiment, wären auch Filme mit negativer Bewertung erforderlich gewesen.

Bei der Betrachtung der Ergebnisse der Methoden fällt auf, dass die Methode, die immer 2 als Bewertung liefert, relativ gut abschneidet. Dies ist in der gerade erwähnten Auswahl der Filme für das zweite Voting begründet. Da diese im Durchschnitt mit 2 bewertet wurden. Das gute Abschneiden der Methode des CF war zu erwarten, da sie sich im Bereich der RS bewährt hat. Das gute Abschneiden der Methode unter Verwendung der Kategorien liegt unter anderem an der Verbesserung des zweiten Ansatzes. Das Einbeziehen des Herkunftslandes führte zu einer deutlichen Verbesserung.

Der Vorteil der beiden Methoden, die im zweiten Experiment entwickelt wurden, ist, dass sie nicht unter dem Kaltstartproblem leiden. Im Gegensatz dazu ist es der Methode „k-nearest“ nicht möglich, Bewertungen für einen Film zu generieren, der noch nicht bewertet wurde.

Im Großen und Ganzen lässt sich sagen, dass eine Verwendung der Kategorien für die Generierung der Bewertungen eine valide Option ist. Zumindest deuten die Ergebnisse bei dem in dieser Arbeit erstellten Test- und Trainingsdatensatz auf ein gutes Ergebnis für diese Methode. Für eine Validierung dieser These müssen weitere Trainings- und Testdatensätze erstellt und weitere Experimente durchgeführt werden.

4. Auswertung

Für weitere Experimente könnte ein **RS** entwickelt werden, welches die Methode der Kategorien zum Umgehen des Kaltstartproblems verwendet. Für Filme mit genügend Bewertungen ist eher die Verwendung eines **RS** mit **CF** zu Empfehlen.

5. Fazit

5.1. Zusammenfassung

In dieser Masterarbeit wurde die Frage behandelt, ob es möglich ist, mithilfe von Selbstauskunft Filmvorschläge zu generieren. Dafür wurden zwei Experimente durchgeführt. Die Teilnehmer beantworteten Fragen zur Person, Bewerteten 22 Kategorien und insgesamt 60 Filme. Dabei wurden ein Test- und ein Trainingsdatensatz erstellt, die von einem parallel entwickelt **RS** verwendet wurden. Abschließend wurden die Ergebnisse des **RS** evaluiert.

Im **1.** Kapitel wurden Ziel und Thema der Arbeit vorgestellt. In Kapitel **2** wurden **RS**, die Erstellung einer Votingplattform und der **KDD** Prozess analysiert. Die beiden Experimente und deren Ziel wurden in Kapitel **3** betrachtet. Darauf folgte eine Betrachtung des im Kontext dieser Arbeit entwickelten **RS**. In Kapitel **4** wurden die Informationen über die Teilnehmer, die Kategoriebewertungen, die Filmbewertungen und die Ergebnisse des **RS** ausgewertet. In diesem, dem **5.** und letzten Kapitel, werden die Erkenntnisse der Arbeit zusammengefasst. Diese Masterarbeit endet mit einem Ausblick.

Im Rahmen dieser Arbeit war es möglich, mithilfe von Selbstauskunft Empfehlungen zu generieren. Es wurde eine Methode gezeigt, welche das Kaltstartproblem löst. Diese Methode verwendet die Kategoriebewertungen und bezieht das Herkunftsland mit ein. Der erste Ansatz dieser Methode, welcher nur die Kategoriebewertungen berücksichtigt, lieferte zwar weniger präzise, aber immer noch akzeptable Ergebnisse.

Die in dieser Masterarbeit durchgeführten Experimente hatten zwei große Probleme. Zum einen war die Anzahl der Teilnehmer relativ klein, um die Qualität der Methoden nachzuweisen. Zum anderen hat sich herausgestellt, dass die Auswahl der Filme des zweiten Experiments ungünstig getroffen wurde. Es wurden nur Filme ausgewählt, welche überdurchschnittlich gut bewertet wurden. Dies erschwert die Bewertung der Qualität ebenfalls.

5.2. Ausblick

Die im Kontext dieser Arbeit entwickelten Methoden können weiterentwickelt werden. Hierfür bieten sich folgende Möglichkeiten:

5. Fazit

- Es können weitere Attribute, wie z.B. Budget für die Vorhersage verwendet werden.
- Es kann eine Gewichtung der Attribute eingeführt werden.
- Die Bildung der Cluster kann verbessert werden.
- Die Einbeziehung der Cluster kann optimiert werden.

Um die Methoden weiter zu evaluieren, können sie auf größere Datensätze angewendet werden. So werden auch die Bewertungen für eher „schlechte“ Filme berechnet.

Eine weitere Möglichkeit ist die Befragung der Teilnehmer nach den Gründen für bestimmte Bewertungen. Die Erkenntnisse aus einer solchen Befragung können für eine Optimierung der Methoden verwendet werden.

Es ist zu erwarten, dass die Bedeutung der **RS** für Onlinedienste weiter zunehmen wird. Somit stellt sich die Frage nach der Zukunft der **RS**. Hector Garcia-Molina beantwortet diese Frage bei der RecSys 2014, mit folgenden drei Punkten:

- mehr Kontrolle über die Strategie,
- Einbeziehung von Freunden und
- Integration von Suche und Empfehlungen.

Aber da Vorhersagen nicht immer leicht zu treffen und nicht immer zutreffend sind, bleibt abzuwarten, wie sich die Zukunft der **RS** entwickeln wird. Eins aber ist mit Sicherheit zu erwarten: die **RSs** werden eine immer größere Rolle spielen.

A. Anhang

A.1. Fragen zur Person

Die Fragen und Antwortmöglichkeiten, welche den Teilnehmern zu ihrer Person gestellt wurden.

- Wie alt bist du?

unter 21

21-30

31-40

41-50

51-60

61-70

über 70

- Bist du männlich oder weiblich?

männlich

weiblich

- Wie schaust du Filme? (Mehrfachauswahl möglich)

VHS

DVD

Internet

Kino

Tablets, etc.

- Wie oft schaust du Filme?

mehrere pro Woche

einen pro Woche

einen pro Monat

seltener als einmal pro Monat

A.2. Bewertungen der Kategorie

Folgende Kategorien wurden bewertet. Die Teilnehmer konnten diese mit einer Wertung von -3 bis 3 bewerten.

- Action
- Abenteuerfilme
- Animationsfilme
- Biografien
- Komödien
- Krimis
- Dokumentationen
- Dramen
- Familienfilme
- Fantasy
- Film Noir
- Geschichtsfilm
- Horrorfilme
- Musikfilme
- Musicals
- Mysteryfilme
- Liebesfilme

- Science Fiction
- Sportfilme
- Thriller
- Kriegsfilm
- Western

A.3. Filme des ersten Experiments

Folgende Filme wurden bewertet (sortiert nach der Kategorie, welche diese beim Voting zugeordnet waren). Die Teilnehmer konnten diese mit einer Wertung von -3 bis 3 bewerten.

- Action
 - Kaththi (2014)
 - John Wick (2014)
 - Redirected (2014)
 - Waar (2013)
 - Inception (2010)
 - Nenokkadine (2014)
 - Gangs of Wasseypur (2012)
 - The Dark Knight Rises (2012)
 - Guardians of the Galaxy (2014)
 - 021 (2014)
- Biografien
 - Ziemlich beste Freunde (2011)
 - The Wolf of the Wall Street (2013)
 - Rush - alles für den Sieg (2013)
 - 12 Years a Slave (2013)
 - The Imitation Game (2014)
 - The King's Speech (2010)

Dallas Buyers Club (2013)

Foxcatcher (2014)

Captain Phillips (2013)

The Fighter (2010)

- Liebesfilme

Haider (2014)

Barfi (2012)

Das Schicksal ist ein mieser Verräter (2014)

Vielleicht lieber morgen (2012)

Before Midnight (2013)

My Name is Khan (2010)

Blau ist eine warme Farbe (2013)

Silver Linings (2012)

The Best Offer (2013)

Liebe (2012)

- Science Fiction

Inception (2010)

Guardians of the Galaxy (2014)

X-Men: Days of Future Past (2014)

Marvel's the Avengers (2012)

Planet der Affen: Revolution (2014)

Edge of Tomorrow (2014)

Gravity (2013)

The Return of the First Avenger (2014)

Star Trek: Into Darkness (2013)

X-Men: Erste Entscheidung (2011)

- Sportfilme
 - Warrior (2011)
 - Rush (2013)
 - Foxcatcher
 - The Fighter (2010)
 - Moneyball (2011)
 - 42 - Die wahre Geschichte einer Sportlegende (2013)
 - United (2011)
 - Secretariat (2010)
 - Win win (2011)
 - Mavericks (2012)

A.4. Filme des zweiten Experiments

- Mad Max: Fury Road (2015)
- Fast & Furious 7 (2015)
- Blind Side ? Die große Chance
- Die Entdeckung der Unendlichkeit
- Tanu Weds Manu Returns (2015)
- Kein Ort Ohne Dich (2015)
- Marvel's The Avengers 2: Age of Ultron (2015)
- Guardians of the Galaxy (2014)
- Unbroken (2014)
- City of McFarland (2015)

A.5. Filmbewertungen des zweiten Experiments nach Herkunftsland

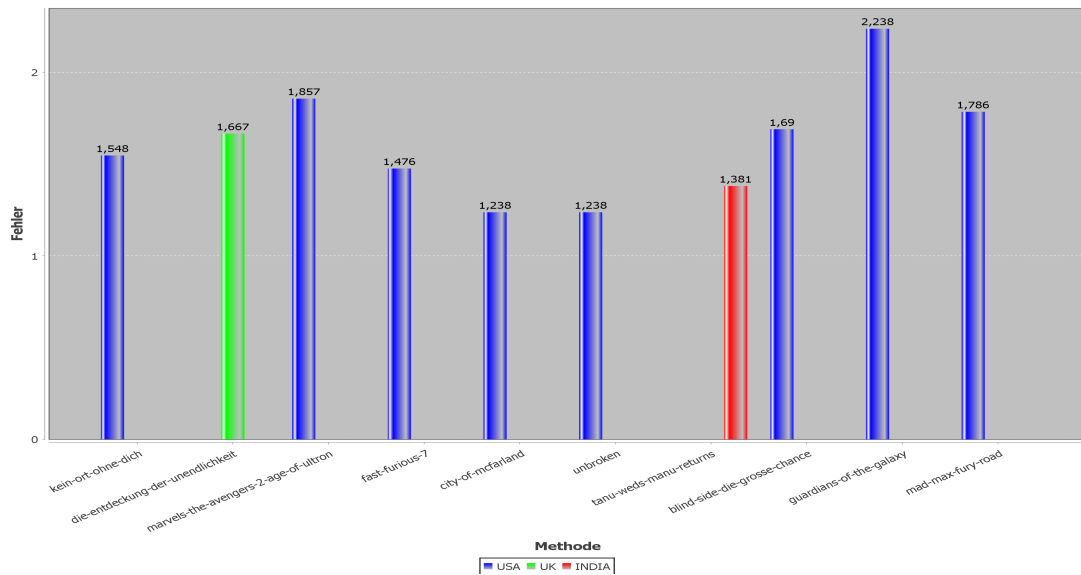


Abbildung A.1.: Filmbewertungen des zweiten Experiments mit Betrachtung des Herkunftsland

A.6. Informationstext des ersten Votings

INFO

Hallo,

vielen Dank fürs Mitvoten.

Da ich tendenziell lange Texte nicht so gerne lese, die wichtigsten Stichpunkte gefolgt von einer ausführlichen Beschreibung:

- Es geht um die Frage: "Ist es möglich – mit der Hilfe von Selbstauskunft – automatisch Filmvorschläge zu generieren?"
- Das Ganze ist Teil meiner Masterarbeit.
- Die Umfrage ist **bis zum 31.01.2015 verlängert** und die Auswertung ist anonymisiert.
- Sie muss nicht an einem Stück gemacht werden und kann jederzeit fortgesetzt werden.
- In ca. drei bis vier Monaten werden noch einmal 5-10 Filme bewertet, darüber werdet ihr per Mail informiert.
- Die Seite ist hauptsächlich in deutsch. Trailer, die ich nicht auf Deutsch gefunden habe, sind in einer anderen Sprache.
- Die Beschreibungen der Kategorien sind die englischen aus IMDB und deren Übersetzung.
- Es werden 22 Kategorien und 50 Filme bewertet.
- Es kann der selbe Film in unterschiedlichen Kategorien vorkommen.
- Ebenfalls werden ein paar Angaben zu der eigenen Person gemacht (Altersgruppe, Geschlecht).
- Die Bewertung geht von -3 (sehr schlecht) bis 3 (sehr gut).
- Unter den Teilnehmern wird ein 20 Euro Gutschein von Amazon verlost.

Abbildung A.2.: Informationstext des ersten Votings

Etwas ausführlicher:

Hallo ich bin Bastian Probst. Im Rahmen meines Masters mache ich eine Studie. Ist es möglich anhand von ein paar Informationen über Personen und ein paar Bewertungen von Filmen, Filmvorschläge zu generieren?

Bisher habe ich mich mit den Fragen beschäftigt, was Benutzermodelle sind, wie diese erstellt werden und wie diese für Vorhersagen verwendet werden können. (Bisherige Arbeiten zu diesem Thema: [diese](#) und [jene](#))

Nun möchte ich ein Programm schreiben, welches eure Bewertungen vorhersagt, mit der Hilfe der Angaben, die ihr hier macht. Daher brauche ich nun eure Hilfe. Alles was ihr tun müsst ist, dass ihr ein paar Fragen zu euch beantwortet und die Kategorien und Filme hier bewertet. Dann kommt in ca. drei bis vier Monaten eine neue Umfrage. Bei dieser werdet ihr dann 5-10 Filme bewerten. Meine Aufgabe ist es ein Programm zu schreiben, welches versucht eure Bewertungen automatisch vorherzusagen.

Abbildung A.3.: Informationstext des ersten Votings

A.7. Informationstext des zweiten Votings

INFORMATIONEN

Vielen Dank für die Teilnahme am zweiten Voting!

- 10 Filme werden bewertet.
- Ein Film aus dem ersten Voting ist dabei.
- Das Voting geht bis zum 18. Juli 2015.
- Den Projektbericht des ersten Votings findest du [hier](#).

 BEARBEITEN

Abbildung A.4.: Informationstext des zweiten Votings

Literaturverzeichnis

- [Albanese u. a. 2013] ALBANESE, Massimiliano ; D'ACIERNO, Antonio ; MOSCATO, Vincenzo ; PERSIA, Fabio ; PICARIELLO, Antonio: A Multimedia Recommender System. In: *ACM Trans. Internet Technol.* 13 (2013), November, Nr. 1, S. 3:1–3:32. – URL <http://doi.acm.org/10.1145/2532640>. – ISSN 1533-5399
- [Amatriain 2013] AMATRIAIN, Xavier: Big & Personal: Data and Models Behind Netflix Recommendations. In: *Proceedings of the 2Nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. New York, NY, USA : ACM, 2013 (BigMine '13), S. 1–6. – URL <http://doi.acm.org/10.1145/2501221.2501222>. – ISBN 978-1-4503-2324-6
- [Baltrunas 2008] BALTRUNAS, Linas: Exploiting Contextual Information in Recommender Systems. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2008 (RecSys '08), S. 295–298. – URL <http://doi.acm.org/10.1145/1454008.1454056>. – ISBN 978-1-60558-093-7
- [Barjasteh u. a. 2015] BARJASTEH, Iman ; FORSATI, Rana ; MASROUR, Farzan ; ESFAHANIAN, Abdol-Hossein ; RADHA, Hayder: Cold-Start Item and User Recommendation with Decoupled Completion and Transduction. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15), S. 91–98. – URL <http://doi.acm.org/10.1145/2792838.2800196>. – ISBN 978-1-4503-3692-5
- [Berry u. a. 2010] BERRY, Steve ; FAZZIO, Steven ; ZHOU, Yongyi ; SCOTT, Bethany ; FRANCISCO-REVILLA, Luis: Netflix Recommendations for Groups. In: *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47*. Silver Springs, MD, USA : American Society for Information Science, 2010 (ASIS&T '10), S. 150:1–150:3. – URL <http://dl.acm.org/citation.cfm?id=1920331.1920522>
- [Biancalana u. a. 2011] BIANCALANA, Claudio ; GASPARETTI, Fabio ; MICARELLI, Alessandro ; MIOLA, Alfonso ; SANSONETTI, Giuseppe: Context-aware Movie Recommendation Based on

- Signal Processing and Machine Learning. In: *Proceedings of the 2Nd Challenge on Context-Aware Movie Recommendation*. New York, NY, USA : ACM, 2011 (CAMRa '11), S. 5–10. – URL <http://doi.acm.org/10.1145/2096112.2096114>. – ISBN 978-1-4503-0825-0
- [Borthakur 2013] BORTHAKUR, Dhruba: Petabyte Scale Databases and Storage Systems at Facebook. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA : ACM, 2013 (SIGMOD '13), S. 1267–1268. – URL <http://doi.acm.org/10.1145/2463676.2463713>. – ISBN 978-1-4503-2037-5
- [Bu u. a. 2010] BU, Jiajun ; TAN, Shulong ; CHEN, Chun ; WANG, Can ; WU, Hao ; ZHANG, Lijun ; HE, Xiaofei: Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content. In: *Proceedings of the 18th ACM International Conference on Multimedia*. New York, NY, USA : ACM, 2010 (MM '10), S. 391–400. – URL <http://doi.acm.org/10.1145/1873951.1874005>. – ISBN 978-1-60558-933-6
- [Burke 2002] BURKE, Robin: Hybrid Recommender Systems: Survey and Experiments. In: *User Modeling and User-Adapted Interaction* 12 (2002), November, Nr. 4, S. 331–370. – URL <http://dx.doi.org/10.1023/A:1021240730564>. – ISSN 0924-1868
- [Burke 2007] BURKE, Robin: The Adaptive Web. Berlin, Heidelberg : Springer-Verlag, 2007, Kap. Hybrid Web Recommender Systems, S. 377–408. – URL <http://dl.acm.org/citation.cfm?id=1768197.1768211>. – ISBN 978-3-540-72078-2
- [Camara u. a. 2012] CAMARA, Fatoumata ; CALVARY, Gaëlle ; DEMUMIEUX, Rachel ; MANDRAN, Nadine: Where Do Facebook Intelligent Lists Come from? In: *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*. New York, NY, USA : ACM, 2012 (IUI '12), S. 289–292. – URL <http://doi.acm.org/10.1145/2166966.2167020>. – ISBN 978-1-4503-1048-2
- [Das u. a. 2007] DAS, Abhinandan S. ; DATAR, Mayur ; GARG, Ashutosh ; RAJARAM, Shyam: Google News Personalization: Scalable Online Collaborative Filtering. In: *Proceedings of the 16th International Conference on World Wide Web*. New York, NY, USA : ACM, 2007 (WWW '07), S. 271–280. – URL <http://doi.acm.org/10.1145/1242572.1242610>. – ISBN 978-1-59593-654-7
- [Deshpande und Karypis 2004] DESHPANDE, Mukund ; KARYPIS, George: Item-based top-N Recommendation Algorithms. In: *ACM Trans. Inf. Syst.* 22 (2004), Januar, Nr. 1, S. 143–177. – URL <http://doi.acm.org/10.1145/963770.963776>. – ISSN 1046-8188

- [Fayyad u. a. 1996] FAYYAD, Usama ; PIATETSKY-SHAPIO, G ; SMYTH, Padhraic: From data mining to knowledge discovery in databases. In: *AI magazine* (1996), S. 37–54. – URL <http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1230>. – ISBN 0-262-56097-6
- [Gantner u. a. 2010] GANTNER, Z. ; DRUMOND, L. ; FREUDENTHALER, C. ; RENDLE, S. ; SCHMIDT-THIEME, L.: Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, Dec 2010, S. 176–185. – ISSN 1550-4786
- [Gomez-Uribe und Hunt 2015] GOMEZ-URIBE, Carlos A. ; HUNT, Neil: The Netflix Recommender System: Algorithms, Business Value, and Innovation. In: *ACM Trans. Manage. Inf. Syst.* 6 (2015), Dezember, Nr. 4, S. 13:1–13:19. – URL <http://doi.acm.org/10.1145/2843948>. – ISSN 2158-656X
- [Herlocker u. a. 2004] HERLOCKER, Jonathan L. ; KONSTAN, Joseph A. ; TERVEEN, Loren G. ; RIEDL, John T.: Evaluating Collaborative Filtering Recommender Systems. In: *ACM Trans. Inf. Syst.* 22 (2004), Januar, Nr. 1, S. 5–53. – URL <http://doi.acm.org/10.1145/963770.963772>. – ISSN 1046-8188
- [Jain u. a. 1999] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data Clustering: A Review. In: *ACM Comput. Surv.* 31 (1999), September, Nr. 3, S. 264–323. – URL <http://doi.acm.org/10.1145/331499.331504>. – ISSN 0360-0300
- [Jannach u. a. 2015] JANNACH, Dietmar ; LERCHE, Lukas ; JUGOVAC, Michael: Adaptation and Evaluation of Recommendations for Short-term Shopping Goals. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15), S. 211–218. – URL <http://doi.acm.org/10.1145/2792838.2800176>. – ISBN 978-1-4503-3692-5
- [Joshi und Gatica-Perez 2006] JOSHI, Dhiraj ; GATICA-PEREZ, Daniel: Discovering Groups of People in Google News. In: *Proceedings of the 1st ACM International Workshop on Human-centered Multimedia*. New York, NY, USA : ACM, 2006 (HCM '06), S. 55–64. – URL <http://doi.acm.org/10.1145/1178745.1178757>. – ISBN 1-59593-500-2
- [Kaminskas u. a. 2013] KAMINSKAS, Marius ; RICCI, Francesco ; SCHEDL, Markus: Location-aware Music Recommendation Using Auto-tagging and Hybrid Matching. In: *Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2013 (RecSys

- '13), S. 17–24. – URL <http://doi.acm.org/10.1145/2507157.2507180>. – ISBN 978-1-4503-2409-0
- [Koren 2008] KOREN, Yehuda: Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA : ACM, 2008 (KDD '08), S. 426–434. – URL <http://doi.acm.org/10.1145/1401890.1401944>. – ISBN 978-1-60558-193-4
- [Lakiotaki u. a. 2008] LAKIOTAKI, Kleanthi ; TSAFARAKIS, Stelios ; MATSATSINIS, Nikolaos: UTA-Rec: A Recommender System Based on Multiple Criteria Analysis. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2008 (RecSys '08), S. 219–226. – URL <http://doi.acm.org/10.1145/1454008.1454043>. – ISBN 978-1-60558-093-7
- [Lathia u. a. 2010] LATHIA, Neal ; HAILES, Stephen ; CAPRA, Licia ; AMATRIAIN, Xavier: Temporal Diversity in Recommender Systems. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA : ACM, 2010 (SIGIR '10), S. 210–217. – URL <http://doi.acm.org/10.1145/1835449.1835486>. – ISBN 978-1-4503-0153-4
- [Leino und Rähä 2007] LEINO, Juha ; RÄIHÄ, Kari-Jouko: Case Amazon: Ratings and Reviews As Part of Recommendations. In: *Proceedings of the 2007 ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2007 (RecSys '07), S. 137–140. – URL <http://doi.acm.org/10.1145/1297231.1297255>. – ISBN 978-1-59593-730-8
- [Leporini 2011] LEPORINI, Barbara: Google News: How User-friendly is It for the Blind? In: *Proceedings of the 29th ACM International Conference on Design of Communication*. New York, NY, USA : ACM, 2011 (SIGDOC '11), S. 241–248. – URL <http://doi.acm.org/10.1145/2038476.2038523>. – ISBN 978-1-4503-0936-3
- [Li und Li 2013] LI, Lei ; LI, Tao: News Recommendation via Hypergraph Learning: Encapsulation of User Behavior and News Content. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. New York, NY, USA : ACM, 2013 (WSDM '13), S. 305–314. – URL <http://doi.acm.org/10.1145/2433396.2433436>. – ISBN 978-1-4503-1869-3
- [Lin u. a. 2013] LIN, Jovian ; SUGIYAMA, Kazunari ; KAN, Min-Yen ; CHUA, Tat-Seng: Addressing Cold-start in App Recommendation: Latent User Models Constructed from Twitter

- Followers. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA : ACM, 2013 (SIGIR '13), S. 283–292. – URL <http://doi.acm.org/10.1145/2484028.2484035>. – ISBN 978-1-4503-2034-4
- [Mejía u. a. 2012] MEJÍA, Andrés ; JUÁREZ-RAMÍREZ, Reyes ; INZUNZA, Sergio ; VALENZUELA, Rocio: Implementing Adaptive Interfaces: A User Model for the Development of Usability in Interactive Systems. In: *Proceedings of the CUBE International Information Technology Conference*. New York, NY, USA : ACM, 2012 (CUBE '12), S. 598–604. – URL <http://doi.acm.org/10.1145/2381716.2381831>. – ISBN 978-1-4503-1185-4
- [Nguyen u. a. 2014] NGUYEN, Tien T. ; HUI, Pik-Mai ; HARPER, F. M. ; TERVEEN, Loren ; KONSTAN, Joseph A.: Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In: *Proceedings of the 23rd International Conference on World Wide Web*. New York, NY, USA : ACM, 2014 (WWW '14), S. 677–686. – URL <http://doi.acm.org/10.1145/2566486.2568012>. – ISBN 978-1-4503-2744-2
- [Palankar u. a. 2008] PALANKAR, Mayur R. ; IAMNITCHI, Adriana ; RIPEANU, Matei ; GARFINKEL, Simson: Amazon S3 for Science Grids: A Viable Solution? In: *Proceedings of the 2008 International Workshop on Data-aware Distributed Computing*. New York, NY, USA : ACM, 2008 (DADC '08), S. 55–64. – URL <http://doi.acm.org/10.1145/1383519.1383526>. – ISBN 978-1-60558-154-5
- [Perisic] PERISIC, Igor: RecSys 2015 Keynote by Igor Perisic: Recommendations within a Social Network: One Step at a Time
- [Probst 2015a] PROBST, Bastian: Durchführung eines Votings zur Erstellung von Benutzerprofilen. (2015)
- [Probst 2015b] PROBST, Bastian: Durchführung eines zweiten Votings und Erstellung eines Vorhersagesystems. (2015)
- [Resnick u. a. 1994] RESNICK, Paul ; IACOVOU, Neophytos ; SUCHAK, Mitesh ; BERGSTROM, Peter ; RIEDL, John: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA : ACM, 1994 (CSCW '94), S. 175–186. – URL <http://doi.acm.org/10.1145/192844.192905>. – ISBN 0-89791-689-1

- [Resnick und Varian 1997] RESNICK, Paul ; VARIAN, Hal R.: Recommender Systems. In: *Commun. ACM* 40 (1997), März, Nr. 3, S. 56–58. – URL <http://doi.acm.org/10.1145/245108.245121>. – ISSN 0001-0782
- [Ricci u. a. 2015] RICCI, Francesco ; ROKACH, Lior ; SHAPIRA, Bracha ; KANTOR, Paul B.: *Recommender Systems Handbook*. 2nd. New York, NY, USA : Springer-Verlag New York, Inc., 2015. – ISBN 978-1-4899-7637-6
- [Said und Bellogín 2014] SAID, Alan ; BELLOGÍN, Alejandro: Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In: *Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2014 (RecSys '14), S. 129–136. – URL <http://doi.acm.org/10.1145/2645710.2645746>. – ISBN 978-1-4503-2668-1
- [Said u. a. 2010] SAID, Alan ; BERKOVSKY, Shlomo ; DE LUCA, Ernesto W.: Putting Things in Context: Challenge on Context-Aware Movie Recommendation. In: *Proceedings of the Workshop on Context-Aware Movie Recommendation*. New York, NY, USA : ACM, 2010 (CAMRa '10), S. 2–6. – URL <http://doi.acm.org/10.1145/1869652.1869665>. – ISBN 978-1-4503-0258-6
- [Said u. a. 2014] SAID, Alan ; DOOMS, Simon ; LONI, Babak ; TIKK, Domonkos: Recommender Systems Challenge 2014. In: *Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2014 (RecSys '14), S. 387–388. – URL <http://doi.acm.org/10.1145/2645710.2645779>. – ISBN 978-1-4503-2668-1
- [Sarwar u. a. 2001] SARWAR, Badrul ; KARYPIS, George ; KONSTAN, Joseph ; RIEDL, John: Item-based Collaborative Filtering Recommendation Algorithms. In: *Proceedings of the 10th International Conference on World Wide Web*. New York, NY, USA : ACM, 2001 (WWW '01), S. 285–295. – URL <http://doi.acm.org/10.1145/371920.372071>. – ISBN 1-58113-348-0
- [Shi u. a. 2010] SHI, Yue ; LARSON, Martha ; HANJALIC, Alan: Mining Mood-specific Movie Similarity with Matrix Factorization for Context-aware Recommendation. In: *Proceedings of the Workshop on Context-Aware Movie Recommendation*. New York, NY, USA : ACM, 2010 (CAMRa '10), S. 34–40. – URL <http://doi.acm.org/10.1145/1869652.1869658>. – ISBN 978-1-4503-0258-6
- [Skeels und Grudin 2009] SKEELS, Meredith M. ; GRUDIN, Jonathan: When Social Networks Cross Boundaries: A Case Study of Workplace Use of Facebook and LinkedIn. In: *Proceedings*

- of the ACM 2009 International Conference on Supporting Group Work. New York, NY, USA : ACM, 2009 (GROUP '09), S. 95–104. – URL <http://doi.acm.org/10.1145/1531674.1531689>. – ISBN 978-1-60558-500-0
- [Unkelos-Shpigel u. a. 2015] UNKELOS-SHPIGEL, Naomi ; SHERMAN, Sofia ; HADAR, Irit: Finding the Missing Link to Industry: LinkedIn Professional Groups As Facilitators of Empirical Research. In: *Proceedings of the Third International Workshop on Conducting Empirical Studies in Industry*. Piscataway, NJ, USA : IEEE Press, 2015 (CESI '15), S. 43–46. – URL <http://dl.acm.org/citation.cfm?id=2819303.2819318>
- [Vargas und Castells 2014] VARGAS, Saúl ; CASTELLS, Pablo: Improving Sales Diversity by Recommending Users to Items. In: *Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2014 (RecSys '14), S. 145–152. – URL <http://doi.acm.org/10.1145/2645710.2645744>. – ISBN 978-1-4503-2668-1
- [Verstrepfen und Goethals 2015] VERSTREPEN, Koen ; GOETHALS, Bart: Top-N Recommendation for Shared Accounts. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15), S. 59–66. – URL <http://doi.acm.org/10.1145/2792838.2800170>. – ISBN 978-1-4503-3692-5
- [Weiß u. a. 2008] WEISS, Diana ; SCHEUERER, Johannes ; WENLEDER, Michael ; ERK, Alexander ; GÜLBAHAR, Mark ; LINNHOF-POPIEN, Claudia: A User Profile-based Personalization System for Digital Multimedia Content. In: *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*. New York, NY, USA : ACM, 2008 (DIMEA '08), S. 281–288. – URL <http://doi.acm.org/10.1145/1413634.1413687>. – ISBN 978-1-60558-248-1
- [Wikipedia 2016a] WIKIPEDIA: *Parship* — *Wikipedia, Die freie Enzyklopädie*. 2016. – URL <https://de.wikipedia.org/w/index.php?title=Parship&oldid=152337859>. – [Online; Stand 25. August 2016]
- [Wikipedia 2016b] WIKIPEDIA: *WordPress* — *Wikipedia, Die freie Enzyklopädie*. 2016. – URL <https://de.wikipedia.org/w/index.php?title=WordPress&oldid=157121710>. – [Online; Stand 24. August 2016]
- [Zhao u. a. 2015] ZHAO, Qi ; ZHANG, Yi ; FRIEDMAN, Daniel ; TAN, Fangfang: E-commerce Recommendation with Personalized Promotion. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15), S. 219–226. –

URL <http://doi.acm.org/10.1145/2792838.2800178>. – ISBN 978-1-4503-3692-5

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. August 2016

Bastian Probst