Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Analyse der Wahrnehmung von Stimmung in Softwareprojekten durch explorative Datenanalyse

Analyzing the Perception of Sentiments in Software Projects Using Exploratory Data Analysis

Masterarbeit

im Studiengang Informatik

von

Marc Herrmann

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Ann-Christin Klünder
Betreuer: Dr. rer. nat. Jil Ann-Christin Klünder

Hannover, 17.04.2023

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 17.04.2023

Marc Herrmann

Zusammenfassung

Aufgrund der steigenden Komplexität in Softwareprojekten gewinnen soziale Aspekte, wie die Kommunikation zwischen den Entwicklern, zunehmend an Bedeutung. Während sich bestärkende Kommunikation positiv auf die Produktivität auswirkt, vermindert inadäquate Kommunikation die Produktivität und das Wohlbefinden der Entwickler. Für Projektleiter ist es daher von Interesse, die Kommunikation innerhalb ihrer Entwicklungsteams mit Stimmungsanalyse zu beaufsichtigen. Ein Hindernis ist jedoch, dass Stimmungsanalysetools bislang nicht in der Lage sind, die unterschiedlichen Wahrnehmungen der Stimmung zwischen einzelnen Entwicklern zu differenzieren. So zeigt eine kürzliche Studie starke Diskrepanzen zwischen der Wahrnehmung von Informatikern und den wissenschaftlichen Autoren von Datensätzen, die zur Anwendung der Stimmungsanalyse im Software Engineering genutzt werden, auf. Daraus folgt, dass die Stimmungsanalysetools für eine industrielle Anwendung auf die Wahrnehmung der Entwickler kalibriert werden müssen. Im Rahmen dieser Arbeit sollen die verschiedenen Wahrnehmungen der Stimmung von 94 Informatikern mittels explorativer Datenanalyse erforscht werden, um erste Anhaltspunkte dafür zu finden, wie eine solche Kalibrierung aussehen könnte. Dabei werden zwei Gruppen von Informatikern identifiziert, die eine signifikant unterschiedliche Wahrnehmung der Stimmung von Aussagen, aus der Domäne der kollaborativen Softwareentwicklung, aufweisen. Diese Ergebnisse können genutzt werden, um in einer Folgestudie zu untersuchen, was die unterschiedlichen Wahrnehmungen beeinflusst. Somit soll langfristig die industrielle Anwendbarkeit der Stimmungsanalyse im Software Engineering gefördert werden.

Abstract

Analyzing the Perception of Sentiments in Software Projects Using Exploratory Data Analysis

Since software projects are becoming progressively complex, social aspects such as the communication between developers are becoming increasingly important. While encouraging communication has a positive effect on productivity, inadequate communication can diminish the productivity and well-being of developers. Therefore, it is of interest for project managers to monitor communication within their development teams using sentiment analysis. However, one obstacle is that sentiment analysis tools have been unable to distinguish between individual developers' perceptions of sentiments. For instance, a recent study revealed strong discrepancies between the perceptions of computer scientists and the scientific authors of datasets used to apply sentiment analysis in software engineering. Therefore, sentiment analysis tools must be calibrated to the individual developers' perceptions for an industrial application. In this thesis, the different perceptions of sentiments of 94 computer scientists will be investigated using exploratory data analysis to find first clues on how such a calibration could look like. Using this approach, two groups of computer scientists were identified that showed significantly different perceptions of sentiments regarding statements from the collaborative software development domain. These results can be used to investigate what influences these different perceptions in a follow-up study. Thus, the industrial applicability of sentiment analysis in software engineering shall be supported.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Durch die stetig steigende Komplexität von Softwareprojekten [7] und dem damit verbundenen Aufwand steigen auch die Anforderungen an die Koordination und Zusammenarbeit der Entwickler untereinander [78]. Neben der technischen Komplexität haben auch die sozialen Aspekte [113] inmitten der Entwickler einen Einfluss auf den Erfolg eines Softwareprojektes [168]. Ein entscheidender sozialer Aspekt ist dabei die Kommunikation [94]. Dies betrifft sowohl die verbale Kommunikation in Meetings [77], als auch die textuelle Kommunikation mittels kollaborativer Software [98], zwischen den Entwicklern [81]. So können bestärkende Aussagen in Meetings positive gruppendynamische Affekte hervorrufen und die allgemeine Stimmung innerhalb eines Entwicklungsteams verbessern [136]. Gut gelaunte Entwickler sind wiederum in der Lage, Problemstellung besser zu lösen, und so das Projekt voranzutreiben [42]. Wenn die Entwickler hingegen schlecht gelaunt sind, z. B. weil durch unangemessene Kommunikation negative Affekte hervorgerufen worden sind, wird ihre Produktivität beeinträchtigt [43]. Inadäquate Kommunikation im Software Engineering kann zudem weitreichende Folgen wie einen Burn-out der betreffenden Entwickler verursachen [153]. Aufgrund dessen wird in der Forschung im Bereich des Software Engineerings sowohl Kommunikation innerhalb von Meetings [76], als auch textuelle Kommunikation [81], wie sie in der dezentralen kollaborativen Softwareentwicklung üblich ist [45], analysiert.

1.1 Motivation

Stimmungsanalyse ist ein Verfahren mit dem Ziel, Textdaten automatisiert nach ihrer Sentiment-Polarität (*negativ*, *neutral* oder *positiv*) zu klassifizieren [95]. Neben einer Vielzahl von weiteren Anwendungsfällen [95], wird die Stimmungsanalyse auch zunehmen im Bereich des Software Engineering angewendet [112]. Als Quellen der Textdaten, welche dem Software Engineering zugeordnet werden können, dienen dabei Aussagen, aus der Domäne

der kollaborativen Softwareentwicklung, von Plattformen wie *Jira* [64, 113], *GitHub* [20, 46], oder *Stack Overflow* [11, 108], sowie Kommunikationsdaten aus internen Textchats [74]. Damit Stimmungsanalysetools in der Lage sind, unbekannte Daten möglichst akkurat zu klassifizieren, werden entweder maschinelle Lernverfahren [142], oder lexikonbasierte Verfahren [63] angewendet. Bei dem Vergleich der verschiedenen Ansätze erreichen Tools, welche maschinelle Lernverfahren zur Klassifikation nutzen, die höchste Genauigkeit [110, 169]. Einen wichtigen Beitrag zur Klassifikationsgenauigkeit dieser Tools leisten dabei manuell annotierte Datensätze von wissenschaftlichen Autoren [109], die für das Training und die Evaluation maschineller Lernverfahren eingesetzt werden [37].

Da Kommunikation die Stimmung und Produktivität von Entwicklern sowohl positiv [42] als auch negativ [43] beeinflussen kann, ist eine automatisierte Klassifikation der Kommunikation, wie durch Stimmungsanalysetools ermöglicht, besonders für Projektleiter von Interesse. Das Potenzial liegt dabei in der Anwendung von Echtzeit-Stimmungsanalyse auf textuelle Kommunikation innerhalb von Entwicklungsteams [137]. Auch eine Anwendung der Stimmungsanalyse auf verbale Kommunikation in Meetings ist denkbar [53]. Falls Kommunikationsprobleme auftreten, haben Projektleiter so die Möglichkeit frühzeitig zu intervenieren und negative Auswirkungen auf den Erfolg des Projektes zu verhindern [137]. Für ein solches Szenario ist eine zu hohe Menge an Fehlklassifikationen der Aussagen allerdings nicht akzeptabel. Genau darin besteht aber auch ein grundlegendes Problem: Die Stimmungsanalysetools können nur so konsistente und genaue Ergebnisse hervorbringen, wie die Daten, auf denen sie trainiert wurden, ermöglichen [36]. Wenn die Datensätze, welche zum Training der Stimmungsanalysetools genutzt werden, voreingenommen sind, klassifizieren die Tools zwar genauso wie sie es gelernt haben, jedoch sind diese Klassifikationen nur im subjektiven Sinne der Autoren korrekt [54]. Eine kürzlich veröffentlichte systematische Literaturrecherche von Obaidi und Klünder [112] zeigt auf, dass viele Autoren von Datensätzen und Stimmungsanalysetools von Schwierigkeiten bei der manuellen Annotation der Daten aufgrund von Subjektivität berichten [22, 61, 154]. Die komplexe Subjektivität liegt dabei in der Natur der Aufgabe, den Aussagen die Sentiment-Polaritäten zuzuordnen, und führt nicht selten zu negativen Ergebnissen [68, 88].

1.2 Problemstellung

Eine kürzliche Studie von Herrmann et al. [54] untersuchte die Subjektivität der Wahrnehmung von Stimmung für Aussagen der Plattformen *Stack Overflow* und *GitHub*, welche zwei in der Forschung etablierten Datensätzen [84, 107] entsprangen. Dafür wurden Informatiker befragt, je 100 Aussagen aus den Datensätzen entsprechend ihrer persönlichen

Wahrnehmung mit einer der Sentiment-Polaritäten *negativ*, *neutral* oder *positiv* zu annotieren [54]. Eine Analyse der Umfrageergebnisse zeigte anschließend, dass die Annotationen der Studienteilnehmer durchschnittlich nur zu 62.5 % mit den vorgegebenen Annotationen der wissenschaftlichen Autoren, in den Datensätzen selbst, übereinstimmten [54]. Diese enorme Diskrepanz zeigt auf, dass die selbst die Autoren der Datensätze nicht in der Lage sind, mit ihren Annotationen die Wahrnehmung der Stimmung von potenziellen Mitgliedern eines Entwicklungsteams widerzuspiegeln [54]. Dies gilt insbesondere für die Wahrnehmung von jedem einzelnen Entwickler: Während manche Teilnehmer eine gute Übereinstimmung mit den Annotationen der Autoren erzielten, wiesen andere Teilnehmer eine erhebliche Uneinigkeit mit den Autoren auf [54]. Damit ein Stimmungsanalysetool die Kommunikation in einem bestimmten Entwicklungsteam so erfassen kann wie von den Entwicklern selbst wahrgenommen, bedarf es folglich einer Kalibrierung der Datensätze und Tools auf das Team [54]. Wie eine solche Kalibrierung aussehen könnte, ist jedoch derzeit noch unklar.

1.3 Lösungsansatz

Um eine Kalibrierung von Stimmungsanalysetools auf ein spezifisches Entwicklungsteam zu ermöglichen, sollen nun die Umfrageergebnisse [111] aus der Studie von Herrmann et al. [54] mittels explorativer Datenanalyse [152] untersucht werden. Die Methodiken der explorativen Datenanalyse dienen dabei zur Identifikation von korrelativen Strukturen innerhalb der Daten [71]. Falls es mehrere Gruppen von Studienteilnehmern gibt, die eine ähnliche Wahrnehmung der Stimmung aufweisen, so können diese Gruppen mit Verfahren wie der Clusteranalyse identifiziert werden [71]. Anschließend können diese Teilnehmergruppen wiederum auf Unterschiede in der Wahrnehmung ihrer Stimmung analysiert werden, um diese zu charakterisieren. So kann Aufschluss darüber gewonnen werden, wie viele unterschiedliche Wahrnehmungen für die Kalibrierung eines Stimmungsanalysetools berücksichtigt werden müssen, und was diese unterscheidet. Eine weitere Möglichkeit auf den Ergebnissen der Clusteranalyse aufzubauen, ergibt sich dadurch, dass die Unterschiede der demografischen Merkmale zwischen den Studienteilnehmern der verschiedenen Teilnehmergruppen untersucht werden können. Auch wenn in der Studie von Herrmann et al. [54] nur wenige demografische Merkmale erfasst wurden, könnten somit erste Unterschiede oder Gemeinsamkeiten der Teilnehmergruppen identifiziert werden, welche für die Kalibrierung von Stimmungsanalysetools relevant sind.

1.4 Zielsetzung der Arbeit

Eine explorative Datenanalyse dient nicht dem Zweck, vordefinierte Forschungsfragen zu beantworten, sondern dazu, ein allgemein besseres Verständnis über eine Datenbasis zu erlangen [152]. Im Rahmen dieser Arbeit

sollen auf diese Art und Weise Erkenntnisse über die Wahrnehmung verschiedener potenzieller Mitglieder eines Entwicklungsteams gewonnen werden. Die Wahl spezifischer Analysemethoden muss dabei im Verlauf der explorativen Datenanalyse basierend auf Zwischenergebnissen angepasst werden (z. B. für die Anzahl der identifizierten Teilnehmergruppen gemäß den Ergebnissen der Clusteranalyse). Ziel dieser Arbeit ist es also, durch eine explorative Datenanalyse der Umfrageergebnisse [111] aus der Studie von Herrmann et al. [54] Anhaltspunkte dafür zu finden, ob es Gruppen von Entwicklern gibt, die eine ähnliche Wahrnehmung der Stimmung aufweisen. Falls sich die Studienteilnehmer der Datenbasis zu Gruppen mit einer ähnlichen Wahrnehmung der Stimmung zusammenfassen lassen, kann anschließend analysiert werden, wie die Wahrnehmungen dieser Gruppen voneinander abweichen. Zudem können statistische Testverfahren angewendet werden, um zu überprüfen, ob es Unterschiede in den demografischen Merkmalen zwischen diesen Teilnehmergruppen gibt. Langfristig sollen die so gewonnenen Erkenntnisse eingesetzt werden, um die Genauigkeit der Klassifikationen eines Stimmungsanalysetools im Vergleich zu der subjektiven Wahrnehmung eines spezifischen Entwicklers oder Entwicklungsteams zu erhöhen. Somit kann in Zukunft eine industrielle Anwendung der Stimmungsanalyse im Software Engineering erfolgen.

1.5 Struktur der Arbeit

Diese Arbeit ist wie folgt strukturiert. In Kapitel 2 werden einige der im Rahmen dieser Arbeit verwendeten Methodiken erläutert, um dem Leser das Verständnis im weiteren Verlauf dieser Arbeit zu erleichtern. Kapitel 3 betrachtet verwandte Arbeiten im Bereich der Stimmungsanalyse und Clusteranalyse in Softwareprojekten, von denen diese Arbeit abgegrenzt wird. Eine Beschreibung der Datenbasis dieser Arbeit, sowie eine detaillierte Erläuterung des Forschungsdesigns, befindet sich in Kapitel 4. Im Anschluss werden die Ergebnisse der angewendeten Verfahren in Kapitel 5 präsentiert. Die Interpretation der Ergebnisse, sowie die Diskussion über deren Einschränkungen, findet sich in Kapitel 6 wieder. Beendet wird diese Arbeit mit einer Zusammenfassung sowie einem Ausblick in Kapitel 7. Ergänzende Informationen zum Hauptteil dieser Arbeit sind in Anhang A angefügt.

Kapitel 2

Grundlagen

In diesem Kapitel werden die methodischen Verfahren erklärt, die für das Verständnis der nachfolgenden Kapitel relevant sind. Dazu gehören die Imputation, die Korrelation, die hierarchische Clusteranalyse, die Dimensionsreduktion und die logistische Regressionsanalyse.

2.1 Imputation

Fehlende Daten sind seit jeher eine Herausforderung für die Anwendung statistischer Analysen [87]. Auch in der Forschung im Software Engineering muss daher auf sogenannte Imputationsverfahren (d. h. Verfahren zur Vervollständigung fehlender Daten) zurückgegriffen werden [103]. Um die im folgenden erläuterten Imputationsverfahren und ihre jeweiligen Stärken und Schwächen verstehen zu können, ist es zunächst notwendig, sich mit den unterschiedlichen Mechanismen fehlender Daten auseinanderzusetzen.

2.1.1 Mechanismen fehlender Daten

Die von Rubin [125] definierten Mechanismen fehlender Daten werden im Folgenden erläutert. Die formale Notation wurde dabei aus Kapitel 2.2.3 des Buches „*Flexible Imputation of Missing Data*“ [155] von Stef van Buuren übernommen und für ein besseres Verständnis angepasst.

Sei Y eine $n \times p$ Datenmatrix der Stichprobengröße n mit p beobachteten Variablen. Sei R eine $n \times p$ Antwortmatrix. Die Elemente der Matrizen Y und R können jeweils als y_{ij} und r_{ij} notiert werden, wobei $i \in \{1, 2, \ldots, n\}$ und $j \in \{1, 2, \ldots, p\}$ gilt. Wenn y_{ij} in Y observiert wurde, dann sei $r_{ij} = 1$, und wenn y_{ij} in Y fehlt, dann sei $r_{ij} = 0$. Sei die Gesamtheit der observierten Daten Y_{obs}, dann enthält Y_{obs} alle Elemente y_{ij} für die $r_{ij} = 1$ gilt. Sei die Gesamtheit der fehlenden Daten Y_{mis}, dann enthält Y_{mis} alle Elemente y_{ij} für die $r_{ij} = 0$ gilt. Y_{mis} enthält somit die echten Werte der fehlenden Daten, welche unbekannt sind. Damit ist die hypothetisch vollständige Datenmatrix $\{Y_{obs}, Y_{mis}\}$. Die Verteilung der Antwortmatrix R hängt also von $\{Y_{obs}, Y_{mis}\}$

ab, diese Beziehung wird durch $P(R \mid Y_{obs}, Y_{mis})$ beschrieben. Rubin [125] definierte 1976 die folgenden drei Mechanismen fehlender Daten.

MCAR (engl. *Missing Completely at Random*): Die Daten von Y fehlen komplett zufällig, wenn die Gleichung

$$\mathrm{P}(R = 0 \mid Y_{obs}, Y_{mis}) = \mathrm{P}(R = 0) \tag{2.1}$$

erfüllt ist. Das bedeutet, dass die Verteilung der fehlenden Variablen weder von den fehlenden Variablen selbst, noch von den beobachteten Variablen abhängig ist. Beispiel: „*Ein Studienteilnehmer hat die Angabe einer Antwort vergessen.*“ Im Beispiel wurde das Fehlen der Antwort rein zufällig verursacht und es gibt keinen Zusammenhang zu den beobachteten Variablen oder der fehlenden Variable selbst.

MAR (engl. *Missing at Random*): Die Daten von Y fehlen zufällig, wenn

$$\mathrm{P}(R = 0 \mid Y_{obs}, Y_{mis}) = \mathrm{P}(R = 0 \mid Y_{obs}) \tag{2.2}$$

gilt. Das bedeutet, dass die Verteilung der fehlenden Variablen nicht von den fehlenden Variablen selbst, aber von den beobachteten Variablen abhängig ist. Beispiel: „*Weibliche Studienteilnehmer geben seltener ihr Alter an.*“ Bei diesem Mechanismus hängt das Fehlen der Variable also von einer anderen beobachteten Variable ab. Im Beispiel hängt das Fehlen des Alters vom Geschlecht der Studienteilnehmer ab.

MNAR (engl. *Missing Not at Random*): Die Daten von Y fehlen nicht zufällig, genau dann, wenn ausschließlich

$$\mathrm{P}(R = 0 \mid Y_{obs}, Y_{mis}) \tag{2.3}$$

gilt. Wenn weder 2.1 noch 2.2 erfüllt sind, dann hängt die Verteilung der fehlenden Variablen sowohl von den fehlenden Variablen selbst, als auch von den beobachteten Variablen ab. Beispiel: „*Ältere weibliche Studienteilnehmer geben seltener ihr Alter an.*“ Im Beispiel ist das Fehlen des Alters neben dem Geschlecht auch vom Alter selbst abhängig.

Auch wenn Verfahren wie Little's Test [82, 85], welche prüfen sollen, ob Daten komplett zufällig fehlen (MCAR) oder nur zufällig fehlen (MAR) in der Vergangenheit vorgeschlagen wurden, ist die Anwendung dieser Verfahren in der Literatur nicht anerkannt [155]. Zudem ist ein Test, ob die Daten nicht zufällig fehlen (MNAR) prinzipiell unmöglich, da die notwendigen Informationen, die für die Durchführung eines solchen Testes nötig wären, fehlen [155]. Stattdessen wird in der Literatur empfohlen abzuwägen, welche der vorgestellten Mechanismen (MCAR, MAR und MNAR) für das Fehlen von Daten in einer Datenbasis infrage kommen [155]. Darauf basierend kann dann eine Entscheidung darüber getroffen werden, wie mit den fehlenden Daten weiter zu verfahren ist [155].

2.1.2 Imputationsverfahren

Im Allgemeinen unterscheidet man Imputationsverfahren zwischen den Arten singulärer oder multipler sowie univariater oder multivariater Imputation. Im Folgenden werden diese Arten von Imputationsverfahren anhand ihrer Unterschiede voneinander abgegrenzt.

Singuläre Imputation beschreibt Imputationsverfahren, welche fehlende Variablen einer vorliegenden Datenbasis durch eine einzelne Berechnung imputieren [86]. Das heißt, dass nicht mehrere alternativ mögliche Imputationen berechnet werden, und die Imputationen auch nicht iterativ verändert werden [86]. Ein Beispiel für eine singuläre Imputation wäre das Imputieren einer fehlenden Variable in einem Datenpunkt mit dem Mittelwert dieser Variable über alle Datenpunkte, da dafür nur einmalig das arithmetische Mittel der beobachteten Werte dieser Variable gebildet werden muss.

Multiple Imputation beschreibt im Gegensatz dazu Imputationsverfahren, die mehrere Versionen (dazu zählt auch eine iterative Veränderung) an imputierten Daten erstellen, welche anschließend zu einem Ergebnis zusammengefasst werden [126]. Multiple Imputationsverfahren sind unter optimalen Bedingungen in der Lage, die korrekten statistischen Eigenschaften der unvollständigen Daten zu erhalten, also keine Verzerrung zu erzeugen [155]. Ein Beispiel für eine multiple Imputation wäre das Erstellen von mehreren zufällig imputierten Datensätzen, die anschließend per Mehrheitsentscheid zusammengefasst werden.

Univariate Imputation beschreibt Imputationsverfahren, welche sich für die Berechnung fehlender Daten nur auf die fehlende Variable selbst stützen [156]. Dies ist unabhängig davon, ob das Imputationsverfahren singulär oder multipel ist. Ein Beispiel für eine univariate Imputation ist das Imputieren fehlender Altersdaten einzelner Teilnehmer in einem Datensatz mit dem Durchschnitt der beobachteten Alterswerte, da die Berechnung des Durchschnitts nur von der Variable selbst, also dem Alter abhängt.

Multivariate Imputation beschreibt im Gegensatz dazu Imputationsverfahren, die auch andere Variablen zur Berechnung der imputierten Daten heranziehen, oftmals sogar die gesamte Datenbasis [156]. Daraus folgt, dass die Imputation auch die Abhängigkeit der fehlende Variable zu anderen Variablen der Datenbasis berücksichtigt [156]. Das fehlende Alter eines Studienteilnehmers in einem Datensatz mit dem durchschnittlichen Alter von Teilnehmern desselben Geschlechts zu imputieren, wäre ein Beispiel für eine multivariate Imputation. Somit wäre die Berechnung der imputierten Daten zusätzlich auch vom Geschlecht, anstatt nur vom Alter der anderen Teilnehmer abhängig.

2.1.3 MICE

MICE (engl. *Multivariate Imputation by Chained Equations*) ist ein multiples multivariates Imputationsverfahren, welches in der Lage ist, komplett zufällig fehlende Daten (MCAR) und zufällig fehlende Daten (MAR) [125] zu imputieren [157]. Das folgende Beispiel erläutert den Algorithmus von MICE, indem eine Iteration des Verfahrens für eine fiktive Datenmatrix mit fehlenden Variablen durchgeführt wird.

Sei Y die folgende vollständige 6×3 Datenmatrix von sechs Datenpunkten mit je drei fiktiven Variablen. Die Einträge von Y und allen nachfolgenden Abwandlungen Y_n werden mit y_{11} bis y_{63} referenziert. Durch das Löschen der drei Einträge y_{61}, y_{12} und y_{23}, mit den Werten 35, 1 und 80 wird die unvollständige Matrix Y_{obs} gewonnen (vgl. 2.4).

$$Y = \begin{bmatrix} 25 & 1 & 50 \\ 27 & 3 & 80 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 35 & 11 & 200 \end{bmatrix} \Rightarrow \begin{bmatrix} 25 & & 50 \\ 27 & 3 & \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ & 11 & 200 \end{bmatrix} = Y_{obs} \tag{2.4}$$

Nun beginnt der Algorithmus von MICE unter Eingabe von Y_{obs}. Zunächst wird eine initiale vollständige Matrix Y_0 aufgestellt, welche die fehlenden Variablen aus Y_{obs} anhand von Mittelwert Imputation (MI) ersetzt (vgl. 2.5). MICE merkt sich jedoch für den weiteren Verlauf des Verfahrens stets die Positionen y_{61}, y_{12} und y_{23} der fehlenden Einträge aus Y_{obs}.

$$Y_{obs} = \begin{bmatrix} 25 & & 50 \\ 27 & 3 & \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ & 11 & 200 \end{bmatrix} \overset{\text{MI}}{\Longrightarrow} \begin{bmatrix} 25 & 7 & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 29 & 11 & 200 \end{bmatrix} = Y_0 \tag{2.5}$$

Anschließend wird der zuvor imputierte Mittelwert 29 an Position y_{61} wieder entfernt, während die anderen beiden Mittelwerte 7 und 134 an den Positionen y_{12} und y_{23} erhalten bleiben (vgl. 2.6). Nun wird das einzige fehlende Element y_{61} mittels linearer Regression [44] (LR) aus den vollständigen ersten fünf Zeilen von Y_0 errechnet. Somit ergibt sich neuer Wert von 36.25, anstatt des vorherigen Mittelwertes 29 an Position y_{61}.

$$Y_0 = \begin{bmatrix} 25 & 7 & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 29 & 11 & 200 \end{bmatrix} \Rightarrow \begin{bmatrix} 25 & 7 & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ & 11 & 200 \end{bmatrix} \overset{\text{LR}}{\Longrightarrow} \begin{bmatrix} 25 & 7 & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 36.25 & 11 & 200 \end{bmatrix} \tag{2.6}$$

Auf dieselbe Weise wird nachfolgend der initiale Mittelwert 7 an Position y_{12} entfernt, während der neue berechnete Wert 36.25 an Position y_{61} sowie der initiale Mittelwert 134 an Position y_{23} erhalten bleiben. Anschließend werden die letzten fünf Zeilen der Matrix genutzt, den neuen Wert von y_{12} mittels linearer Regression zu berechnen (vgl. 2.7). Dadurch ergibt sich an Position y_{12} ein neuer Wert von 1.85 anstatt des vorherigen Mittelwertes 7.

$$\Rightarrow \begin{bmatrix} 25 & & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 36.25 & 11 & 200 \end{bmatrix} \overset{\text{LR}}{\Longrightarrow} \begin{bmatrix} 25 & 1.85 & 50 \\ 27 & 3 & 134 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 36.25 & 11 & 200 \end{bmatrix} \tag{2.7}$$

Die zwei neuen Werte 36.25 und 1.85 an den Positionen y_{61} und y_{12} dienen wiederum als Grundlage für die lineare Regression von y_{23} durch die anderen fünf vollständigen Zeilen der Matrix (vgl. 2.8). Somit ergibt sich ein neuer Wert von 72.77 für y_{23} anstatt des vorherigen Mittelwertes von 134.

$$\Rightarrow \begin{bmatrix} 25 & 1.85 & 50 \\ 27 & 3 & \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 36.25 & 11 & 200 \end{bmatrix} \overset{\text{LR}}{\Longrightarrow} \begin{bmatrix} 25 & 1.85 & 50 \\ 27 & 3 & 72.77 \\ 29 & 5 & 110 \\ 31 & 7 & 140 \\ 33 & 9 & 170 \\ 36.25 & 11 & 200 \end{bmatrix} = Y_1 \tag{2.8}$$

Nachdem jedes der fehlenden Elemente aus Y_{obs} einmal per linearer Regression unter Verwendung aller anderen Zeilen bestimmt wurde, ist die erste Iteration von MICE abgeschlossen (vgl. 2.8). Die neu entstandene Matrix Y_1 liegt bereits nach einer einzelnen Iteration deutlich näher an den tatsächlichen Werten 35, 1, und 80 der fehlenden Einträge y_{61}, y_{12} und y_{23} aus Y. Dieses Beispiel wurde allerdings auch bewusst so gewählt, dass es einen

tatsächlichen linearen Zusammenhang der drei Variablen von Y gibt. Für echte Variablen, wo diese Zusammenhänge schwächer sind, benötigt MICE mehrere Iterationen für eine genaue Imputation. Für eine zweite Iteration von MICE würde nun als Nächstes der Wert 36.25 in Y_1 an Position y_{61} erneut gelöscht und mittels linearer Regression der ersten fünf Zeilen aus Y_1 durch einen neuen Wert ersetzt werden. Die Schritte der Gleichungen 2.6 bis 2.8 werden also sukzessiv erneut mit der Matrix Y_1 durchlaufen, um die Matrix Y_2 zu erhalten. Iterativ lässt sich so aus jeder Matrix Y_n die nächste Matrix Y_{n+1} errechnen. Die imputierten Werte der Matrix stabilisieren sich auf diese Weise recht schnell, d. h. der Betrag $|\, Y_n - Y_{n-1} \,|$ konvergiert gegen null. Zudem ist es möglich für MICE eine maximale Anzahl an Iterationen anzugeben, nach welcher das Verfahren abgebrochen wird.

Da die gesamte vorhandene Datenbasis genutzt wird, um die fehlenden Variablen abzuschätzen, folgt somit, dass MICE in der Lage ist, sowohl komplett zufällig fehlende Variablen (MCAR), als auch zufällig fehlenden Variablen (MAR) zu imputieren [157]. Eine Imputation von nicht zufällig fehlenden Variablen (MNAR) ist mit MICE zwar möglich, jedoch können so entstandene Verzerrungen nicht ohne einen externen Vergleich mit den fehlenden Daten selbst abgeschätzt werden [157]. Für mindestens zufällig fehlenden Variablen (MAR) ist MICE jedoch in der Lage bis zu 90 % fehlender Daten ohne eine Verzerrung der statistischen Eigenschaften zu imputieren [91]. Eine Analyse der vollständigen Datenpunkte, auch CCA (engl. *Complete Case Analysis*), hätte bei zufällig fehlenden Variablen (MAR) hingegen zur Folge, dass eine Verzerrung erzeugt wird [155]. Ein Beispiel dafür wäre, dass mehr weibliche Studienteilnehmer aus der Datenbasis entfernt werden, falls diese öfter nicht ihr Alter angeben. Somit wäre die Stichprobe bei einer Analyse der vollständigen Datenpunkte verzerrt, mit Imputation der unvollständigen Datenpunkte durch MICE jedoch nicht [155]. Neben der vorgestellten Variante von MICE, gibt es vielzählige Variationen, welche sich primär anhand der verwendeten Regressionsverfahren unterscheiden.

2.1.4 MICE-RF (Random Forest Regression)

MICE-Random-Forest [138], kurz MICE-RF, ist ein auf MICE basierendes Verfahren, welches zur Imputation fehlender Daten einen Random Forest [6] nutzt. Ein Random Forest ist ein maschinelles Lernverfahren, welches ein Ensemble von zufällig generierten Entscheidungsbäumen [119] für die Klassifikation von Daten nutzt [6]. Um die Klasse einer abhängigen Variable Y vorherzusagen, wird eine Menge von beobachteten Variablen V, $Y \notin V$ einer Datenbasis D genutzt, um variierende Entscheidungsbäume (für die Abhängigkeit von Y zu V) zu generieren. Im Folgenden wird der Ablauf einer Random Forest Regression vereinfacht in vier Schritten erläutert.

Schritt 1: Aus der Datenbasis D werden zufällig Datenpunkte ausgewählt, um eine modifizierte Datenbasis $\tilde{D}$ mit derselben Anzahl an Datenpunkten und Variablen zu erstellen (engl. *Bagging* [5]). Hierbei werden einzelne Datenpunkte aus D mehrfach für $\tilde{D}$ ausgewählt [6]. Somit sind nicht alle Datenpunkte aus D auch in $\tilde{D}$ enthalten.

Schritt 2: Aus $\tilde{D}$ wird eine zufällige Teilmenge an beobachteten Variablen $\tilde{V} \subset V$ ausgewählt. Die Variable aus $\tilde{V}$, welche die Datenpunkte in $\tilde{D}$ im Sinne der abhängigen Variable Y am besten separiert, wird als Wurzel eines neuen Entscheidungsbaumes ausgewählt [6].

Schritt 3: Analog zum vorherigen Schritt wird für jeden weiteren Knoten im Entscheidungsbaum eine zufällige Teilmenge der Variablen $\tilde{V} \subset V$ aus $\tilde{D}$ bestimmt und für jeden Knoten die Variable gewählt, welche $\tilde{D}$ am besten in die vorherzusagende Klasse Y separiert (unter Kenntnis über den bisherigen Pfad des Entscheidungsbaumes). Dies wird wiederholt, bis eine bestimmte Tiefe im Entscheidungsbaum erreicht ist. Die Blätter des Entscheidungsbaumes repräsentieren dann jeweils die Klassen der abhängigen Variable Y [6].

Schritt 4: Abschließend werden die Schritte 1 bis 3 erneut durchlaufen, um weitere zufällige Entscheidungsbäume zu generieren. Dies geschieht so lange, bis eine bestimmte Anzahl an Entscheidungsbäumen generiert wurde. Durch die zufällige Auswahl von $\tilde{D}$ für jeden Entscheidungsbaum und $\tilde{V}$ für jeden Knoten entstehen Entscheidungsbäume mit großer Variation untereinander [6].

Nachdem die obigen Schritte durchlaufen wurden, ist ein Random Forest, also sinngemäß ein Wald aus zufälligen Entscheidungsbäumen generiert worden [6]. Um nun eine unbekannte Reihe von beobachteten Variablen zu klassifizieren, wird jeder der generierten Entscheidungsbäume entsprechend der Belegung der Variablen durchlaufen. Das Ergebnis Y von jedem einzelnen Entscheidungsbaum wird anschließend erfasst [17]. Die finale Vorhersage von Y wird anschließend aus dem Mehrheitsentscheid (engl. *Aggregation* [5]) der einzelnen Entscheidungsbäume gewonnen [17]. Eine Evaluation des Random Forest ist möglich durch die Klassifikation der Datenpunkte aus D, welche nicht als Teil der modifizierten Datenbasis $\tilde{D}$ ausgewählt worden sind (engl. *Out of Bag Evaluation*) [17]. Dadurch können Parameter wie die Tiefe der Entscheidungsbäume und die Anzahl an berücksichtigten Variablen pro Schritt optimiert werden [6]. MICE-RF [138] nutzt den Algorithmus von MICE (vgl. Kapitel 2.1.3) unter Verwendung einer Random Forest Regression anstelle von einer linearen Regression. Somit besteht die Möglichkeit MICE-RF auch zur Imputation von ordinal- und nominalskalierte Variablen zu nutzen [138]. Eine Python Implementation des Imputationsverfahrens MICE-RF ist im Paket *miceforest* enthalten.

2.2 Korrelation

Die Korrelation ist eine Maßzahl, welche den Zusammenhang von zwei Merkmalen quantifiziert [160]. Für eine Reihe von Datenpaaren $\{x_i, y_i\}$ kann durch einen Korrelationskoeffizienten der Zusammenhang und die Stärke der Beziehung von x und y berechnet werden [160]. Sind x und y stark korreliert, so kann man einen Wert mithilfe des anderen vorhersagen [160]. Im Folgenden werden zwei Korrelationskoeffizienten vorgestellt.

2.2.1 Pearson's r

Der Pearson'sche Korrelationskoeffizient, kurz Pearson's r ist der am weitesten verbreitete Koeffizient zur Berechnung der Korrelation von numerischen Merkmalen [104]. Für eine Reihe von n Datenpaaren $\{x_i, y_i\}$ mit den Mittelwerten $\overline{x}$, $\overline{y}$ berechnet sich Pearson's r wie folgt [131, 160].

$$r_{xy} = \frac{\frac{1}{n}\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\frac{1}{n}\sum(x_i - \overline{x})^2} \cdot \sqrt{\frac{1}{n}\sum(y_i - \overline{y})^2}} \tag{2.9}$$

Durch die Division der Kovarianz von x und y durch die Standardabweichung von x und y wird eine auf das Intervall $[-1, 1]$ normierte Maßzahl berechnet [56]. Damit ist Pearson's r unabhängig von der Skalierung der betrachteten Merkmale [160]. Zudem folgt, dass Pearson's r einen linearen Zusammenhang zwischen den Merkmalen misst. Für die Berechnung von Pearson's r werden daher mindestens intervallskalierte Daten benötigt [160].

2.2.2 Spearman's ϱ

Liegen stattdessen ordinalsklalierte Daten vor, kann stattdessen Spearman's Rangkorrelationskoeffizient [145], kurz Spearman's ϱ, für die Berechnung der Korrelation herangezogen werden [160]. Für die Berechnung von Spearman's ϱ werden zunächst die Werte von x und y zunächst nach ihrer aufsteigenden Größe sortiert [160]. Entsprechend dieser Sortierung werden den Werten die Rangzahlen $R(x_i)$, $R(y_i) \in \{1, 2, \ldots, n\}$ zugeordnet [160]. Die Berechnung von Spearman's ϱ ergibt sich anschließend wie folgt [160].

$$\varrho_{xy} = r_{R(x)R(y)} = \frac{\frac{1}{n}\sum(R(x_i) - \overline{R}_x)(R(y_i) - \overline{R}_y)}{\sqrt{\frac{1}{n}\sum(R(x_i) - \overline{R}_x)^2} \cdot \sqrt{\frac{1}{n}\sum(R(y_i) - \overline{R}_y)^2}} \tag{2.10}$$

Dabei sind $\overline{R}_x$ und $\overline{R}_y$ als Mittelwerte der Rangzahlen von x und y definiert. Wie die Gleichung 2.10 aufzeigt, berechnet Spearman's ϱ also Pearson's r auf den Rangzahlen der Datenpaare $\{x_i, y_i\}$ [160]. Da die Berechnung der Rangzahlen nur erfordert, dass die Werte nach Größe sortiert werden können, ist eine ordinale Skalierung der Eingabedaten für die Berechnung von Spearman's ϱ ausreichend [160].

2.2.3 Vergleich von Pearson's r und Spearman's ϱ

Abbildung 2.1 visualisiert die Unterschiede zwischen Pearson's r und Spearman's ϱ für zwei Reihen von Datenpunkten $\{x_i, y_i\}$. Für einen maximalen linearen Zusammenhang ($y = x$), welcher auf der linken Seite von Abbildung 2.1 dargestellt ist, nehmen sowohl Pearson's r als auch Spearman's ϱ einen Wert von eins, also einer maximal positiven Korrelation von x und y, an. Sobald kein linearer Zusammenhang zwischen den beiden Variablen mehr besteht, sind die berechneten Werte beider Korrelationskoeffizienten jedoch nicht mehr identisch. Auf der rechten Seite von Abbildung 2.1 ist ein nicht linearer Zusammenhang ($y = x^9$) der Datenpunkte dargestellt. Während der Wert von Pearson's r auf 0.8565 absinkt, gibt Spearman's ϱ weiterhin eine maximal positive Korrelation von eins an. Warum das so ist, wird durch Gleichung 2.10 verdeutlicht: Transformiert man alle Datenpunkte $\{x_i, y_i\}$ in ihre jeweiligen Rangzahlen $\{R(x_i), R(y_i)\}$, so erhält man denselben linearen Zusammenhang, der auf der linken Seite von Abbildung 2.1 dargestellt ist. Folglich misst Spearman's ϱ also nur das monotone Wachstum zwischen den Variablen, wobei die Abstände, in denen dieses Wachstum stattfinden, nicht konstant sein müssen. So wird verdeutlicht, warum Spearman's ϱ für die Berechnung von Korrelationen auf ordinalskalierten Daten geeignet ist, da für diese Daten nur die Reihenfolge und nicht die Abstände bekannt sind. Daher kann Spearman's ϱ beispielsweise genutzt werden, um die Zusammenhänge zwischen den Antworten von Likert-Skalen-Fragebögen zu messen. Dabei kann die Korrelation der Antworten auch als Maß der Interrater-Reliabilität interpretiert werden [18].

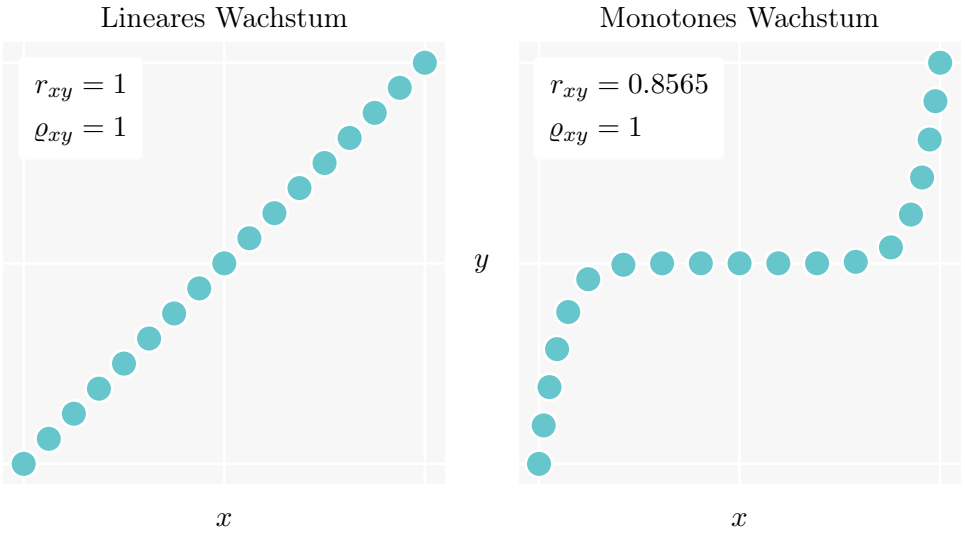

Abbildung 2.1: Vergleich von Pearson's r und Spearman's ϱ für lineares Wachstum ($y = x$) und monotones Wachstum ($y = x^9$).

2.3 Hierarchische Clusteranalyse

Bei der hierarchischen Clusteranalyse handelt es sich, um ein unüberwachtes maschinelles Lernverfahren, mit dem Ziel zusammenhängende Strukturen innerhalb einer Datenbasis erkennbar zu machen [106]. Im Gegensatz zu partitionierenden Clusterverfahren (wie dem k-Means-Algorithmus) ist für das hierarchische Clustering keine Angabe der Anzahl an Clustern, in welche die Eingabedaten partitioniert werden sollen, notwendig. Stattdessen werden beim hierarchischen Clustering Partitionen entsprechend jeder möglichen Anzahl von Clustern ausgegeben. Dabei unterscheidet man zwischen agglomerativem und divisivem hierarchischen Clustering [49].

Agglomeratives hierarchisches Clustering verfolgt einen Bottom-Up-Ansatz [49]. Zu Beginn befindet sich dafür jedes zu gruppierende Objekt innerhalb seines eigenen Clusters [49]. Im Folgenden werden dann iterativ einzelne und bereits gruppierte Objekte zu Clustern fusioniert, indem in jedem Iterationsschritt die kostengünstigste Fusion vollzogen wird [49].

Divisives hierarchisches Clustering verfolgt im Gegensatz dazu ein Top-Down-Ansatz [49]. Dabei befinden sich zu Beginn alle Objekte innerhalb eines einzelnen Clusters [49]. Anschließend wird die Anzahl der Cluster in jedem Iterationsschritt inkrementiert, solange, bis sich jedes Objekt in einem eigenen Cluster befindet [49].

2.3.1 Distanzmaße

Neben der Wahl zwischen dem agglomerativem und divisivem Ansatz, sind für die Durchführung einer hierarchischen Clusteranalyse zusätzlich ein die Wahl eines Distanzmaßes und ein Fusionierungsalgorithmus notwendig. Gängige Distanzmaße sind dabei die euklidische Distanz oder die Manhattan-Distanz [2]. Auch Korrelationskoeffizienten, welche die Ähnlichkeit zweier Objekte angeben, können zu einem Distanzmaß transformiert werden [99]. Für eine Datenbasis mit n Objekten kann anschließend die Distanz d_{ij} für alle Objektpaare berechnet werden. Dadurch entsteht eine symmetrische $n \times n$ Distanzmatrix D [128]. Auf der Hauptdiagonalen von D befinden sich dabei Nullen, da die Distanz von jedem Objekt zu sich selbst Null entspricht [128].

2.3.2 Fusionierungsalgorithmen

Um das hierarchische Clustering durchführen zu können, muss zuletzt noch ein Fusionierungsalgorithmus (engl. *Linkage*) [106] festgelegt werden. Dieser bestimmt auf Basis von D, welche Cluster in jedem iterativen Schritt miteinander fusioniert (oder bei divisivem hierarchischen Clustern aufgetrennt) werden [106]. Bei den Fusionierungsalgorithmen handelt es sich in der Regel um Greedy-Algorithmen [148], da diese jeweils nur die günstigste Fusion innerhalb einer Iteration ermitteln. Um nachfolgend einige

gängige Fusionierungsalgorithmen vorzustellen, seien $a \in A$ die Objekte eines Clusters A und $b \in B$ die Objekte eines Clusters B. Dann sind die Fusionierungsalgorithmen Single-Linkage, Complete-Linkage, und Average-Linkage wie folgt definiert [106].

$$\textbf{Single-Linkage:} \quad \min_{a \in A,\, b \in B} d(a,\, b) \tag{2.11}$$

$$\textbf{Complete-Linkage:} \quad \max_{a \in A,\, b \in B} d(a,\, b) \tag{2.12}$$

$$\textbf{Average-Linkage:} \quad \frac{1}{|A| \cdot |B|} \sum_{a \in A,\, b \in B} d(a,\, b) \tag{2.13}$$

Die Fusionierungsalgorithmen Single-Linkage und Complete-Linkage ermitteln also jeweils den minimalen und maximalen Abstand zwischen allen Objekten beider Cluster. Im Gegensatz dazu wird für die Average-Linkage der durchschnittliche Abstand zwischen den Objektpaaren $\{a, b\}$ der beiden Cluster bestimmt. Diese drei Fusionierungsalgorithmen wenden einfache Kriterien zur Bestimmung der zu fusionierenden Cluster an. Ward's minimales Varianzkriterium [162] (engl. *Ward-Linkage*) ist ein weiterer Fusionierungsalgorithmus, welcher in jedem Iterationsschritt die zwei Cluster fusioniert, für welche das fusionierte Cluster $A \cup B$ die niedrigste Varianz aufweist [106]. Dafür werden in jedem Fusionsschritt die Cluster A und B fusioniert, welche das folgende Kriterium minimieren [106].

$$\textbf{Ward's minimales Varianzkriterium:} \quad \sum_{x \in A \cup B} d(x,\, \overline{A \cup B})\,, \;\; \overline{A \cup B} = \frac{1}{|A \cup B|} \sum_{x \in A \cup B} x \tag{2.14}$$

Dabei ist $\overline{A \cup B}$ als Zentroid von $A \cup B$ durch Gleichung 2.14 definiert. Für die Berechnung von Ward's minimalem Varianzkriterium wird also die Summer der Abstände aller Objekte der beiden Cluster A und B zum Zentroid des fusionierten Clusters $\overline{A \cup B}$ bestimmt. Diese Berechnung wird für jede mögliche Kombination von zwei Clustern A und B aus der Menge der bestehenden Cluster durchgeführt. Anschließend werden die beiden Cluster, die nach der Fusion die Abstände ihrer Objekte zum Zentroid minimieren, miteinander fusioniert [106].

2.3.3 Vergleich verschiedener Fusionierungsalgorithmen

Die Wahl des Fusionierungsalgorithmus beeinflusst maßgeblich die Ergebnisse des hierarchischen Clusterings [106], wie Abbildung 2.2 (auf der umliegenden Seite) veranschaulicht. Abbildung 2.2 vergleicht die Clustering-Ergebnisse der zuvor vorgestellten Fusionierungsalgorithmen für drei fiktive Datenverteilungen. Die Beispieldaten wurden dem Python-Paket *scikit-learn* [117] entnommen. Die Single-Linkage neigt zur Kettenbildung, da immer die Cluster mit den am nächsten benachbarten Objekten fusioniert

werden. Daher kann die Single-Linkage als einziger Fusionierungsalgorithmus die konkaven Formen im ersten Beispiel von 2.2 korrekt voneinander abgrenzen. Sind die Strukturen der Objekte in den Daten dicht und deutlich voneinander abgegrenzt, so können alle Verfahren die Cluster korrekt identifizieren, wie das zweite Beispiel in Abbildung 2.2 zeigt. Im letzten Beispiel von Abbildung 2.2 sind die Strukturen der Objekte weniger deutlich voneinander abgegrenzt. Zudem weist die große Struktur im Zentrum eine niedrigere Dichte auf, als die zwei äußeren Strukturen, und es sind mehrere Ausreißer vorhanden. Die Single-Linkage schneidet bei diesem Beispiel am schlechtesten ab. Nur durch Ward's minimales Varianzkriterium kann das kleine dichte Cluster am rechten Rand der Datenverteilung in Grün korrekt vom größeren blauen Cluster im Zentrum abgegrenzt werden. Der Fusionierungsalgorithmus Average-Linkage schneidet am zweitbesten ab, ordnet jedoch drei Objekte dem falschen Cluster zu (grün statt blau). Abbildung 2.2 zeigt auf, dass jeder Fusionierungsalgorithmus seine Vor- und Nachteile bietet. Für die Auswahl eines Verfahrens ist daher nötig abzuwägen, welche der Vor- und Nachteile für die eigene Datenbasis den größten Mehrwert bieten.

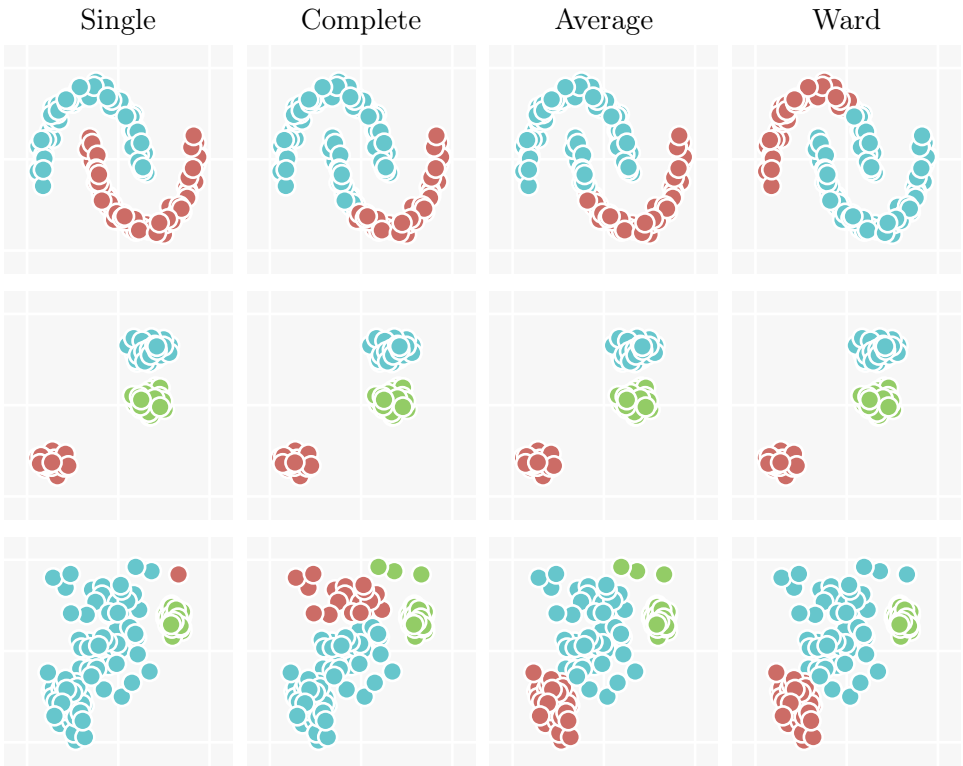

Abbildung 2.2: Vergleich der Fusionierungsalgorithmen Single-Linkage, Complete-Linkage, Average-Linkage und Ward's minimalem Varianzkriterium. Es sind je drei Beispiele (von oben nach unten) getestet worden.

2.4 Dimensionsreduktion

Die Möglichkeiten große Mengen an Daten zu sammeln, und zu speichern, steigen stetig [35]. Jedoch sind die Ressourcen und Möglichkeiten diese Daten auszuwerten, um daraus sinnvolle Informationen zu gewinnen, oft sehr beschränkt [35]. Im Forschungsbereich der Data Science beschäftigt man sich daher damit, die Dimensionalität (d. h. die Anzahl der Variablen pro Datenpunkt) in Datensätzen so zu reduzieren, dass wichtige Informationen der Daten dennoch erhalten bleiben [35]. Dies ermöglicht unter anderem die Anwendung von maschinellen Lernverfahren, deren Berechnung auf den ursprünglichen hochdimensionalen Daten zu aufwendig gewesen wäre [35]. Im Folgenden werden hochdimensionale Daten definiert sowie mit der Hauptkomponentenanalyse [66] und der linearen Diskriminanzanalyse [32] zwei traditionelle Dimensionsreduktionsverfahren vorgestellt.

2.4.1 Hochdimensionale Daten

Ob Daten als hochdimensional gelten, hängt von dem Verhältnis von Datenpunkten, zu beobachteten Variablen pro Datenpunkt ab [135]. Ist die Anzahl an Variablen nahe der Anzahl an Datenpunkten (oder übersteigt diese sogar), kann von hochdimensionalen Daten gesprochen werden [135]. Dies wäre z. B. der Fall, falls im Rahmen einer Studie 90 Messwerte von 100 Studienteilnehmern gesammelt wurden: Die Anzahl der Variablen (90) ist nahe der Anzahl an Datenpunkten (100). Aufgrund des „*Fluches der Dimensionalität*“ [4] nimmt die räumliche Dichte der Daten exponentiell mit der Anzahl an Dimensionen ab. Dadurch geht der Bezug benachbarten Datenpunkten im hochdimensionalen Raum verloren [4].

2.4.2 Hauptkomponentenanalyse

Bei der Hauptkomponentenanalyse (engl. *Principal Component Analysis*) [66] handelt es sich, um ein unüberwachtes Dimensionsreduktionsverfahren, mit dem Ziel, die maximale Varianz der ursprünglichen Datenbasis zu erhalten [35]. Dabei wird die Anzahl der Variablen V in der Datenbasis verringert. Zunächst wird für jedes Paar $x, y \in V$ von Variablen die Kovarianz $\sigma_{xy} = \frac{1}{n}\sum(x_i - \overline{x})(y_i - \overline{y})$ über alle n Datenpunkten berechnet. Zudem wird die Varianz ${\sigma_x}^2 = \frac{1}{n}\sum(x_i - \overline{x})^2$ jeder einzelnen Variable $x \in V$ berechnet. Anschließend wird die symmetrische $p \times p$ Kovarianzmatrix $\mathrm{Cov}(V)$ aller Variablen aufgestellt, die sich für p Variablen $\{V_1, V_2, \ldots, V_p\}$ folgendermaßen ergibt [170].

$$\mathrm{Cov}(V) = \begin{bmatrix} {\sigma_1}^2 & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & {\sigma_2}^2 & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & {\sigma_p}^2 \end{bmatrix} \tag{2.15}$$

Um aus der Kovarianzmatrix $\mathrm{Cov}(V)$ die Hauptkomponenten (engl. *Principal Components*) abzuleiten, müssen zunächst alle Eigenwerte und Eigenvektoren [149] der Kovarianzmatrix berechnet werden. Dafür wird zunächst das charakteristische Polynom $\det(\mathrm{Cov}(V) - \lambda \cdot I_p)$ berechnet, wobei I_p die $p \times p$ Identitätsmatrix bezeichnet. Das charakteristische Polynom kann beispielsweise mit dem Gaußschen Eliminationsverfahren [58] berechnet werden, indem die Determinante aufgelöst wird. Anschließend können die Eigenwerte λ_i von $\mathrm{Cov}(V)$ als Nullstellen des charakteristischen Polynoms berechnet werden [151]. Daraufhin kann zu jedem Eigenwert λ_i von $\mathrm{Cov}(P)$ ein zugehöriger Eigenvektor $\vec{x_i}$ durch Lösung von $(\mathrm{Cov}(P) - \lambda_i \cdot I_p) \cdot \vec{x_i} = 0$ berechnet werden [151]. Die erste Hauptkomponente (PC1) der Daten ergibt sich anschließend aus dem Eigenvektor $\vec{x_i}$ mit dem größten Eigenwert λ_i, die zweite Hauptkomponenten (PC2) ergibt sich folglich aus dem Eigenvektor mit dem zweitgrößten Eigenwert, et cetera [35]. Die Varianz, der ursprünglichen Daten, die durch eine Hauptkomponente erklärt werden kann, ergibt sich aus dem Verhältnis des Eigenwertes der jeweiligen Hauptkomponente zu der Summe aller Eigenwerte von $\mathrm{Cov}(P)$ [35]. Abbildung 2.3 veranschaulicht beispielhaft die ersten beiden Hauptkomponenten (PC1 und PC2) einer multivariaten Datenverteilung mit zwei ursprünglichen Dimensionen. Die Vektoren der beiden Hauptkomponenten verlaufen orthogonal zueinander und bilden die gesamte Varianz der Datenverteilung ab. Anhand der ersten Hauptkomponente können jedoch bereits 95.08 % der ursprünglichen Varianz dieser Datenverteilung abgebildet werden (und bei deiner Dimensionsreduktion der Datenpunkte in den eindimensionalen Raum erhalten bleiben).

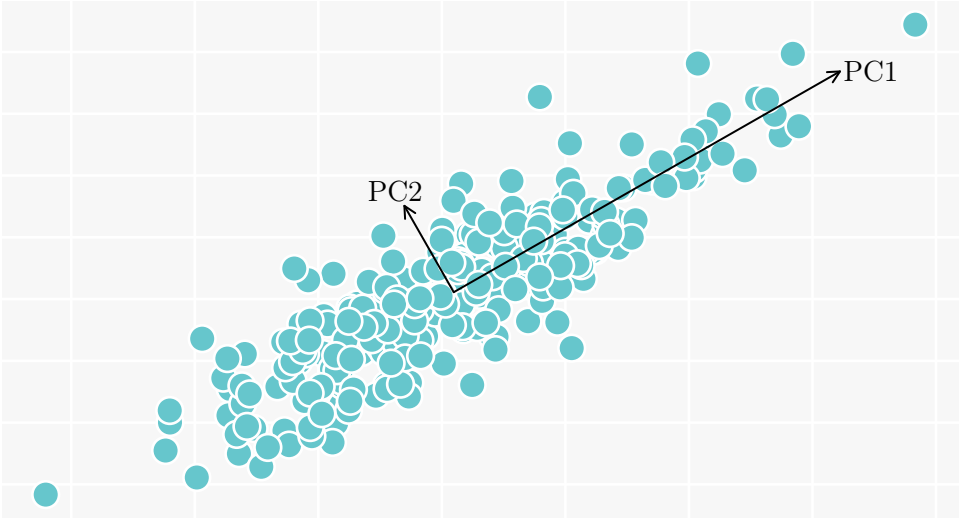

Abbildung 2.3: Die ersten beiden Hauptkomponenten (PC1 und PC2) einer multivariaten Datenverteilung. PC1 bildet 95.08 % der ursprünglichen Varianz ab und PC2 die restlichen 4.92 %.

Um die ursprünglichen hochdimensionalen Daten nun in den niedrigerdimensionalen Raum zu transformieren, ist es nötig, das Skalarprodukt aus den Datenpunkten und der jeweiligen Hauptkomponente zu bilden [35]. Die erste Hauptkomponente berechnet dabei die x-Koordinate im niedrigerdimensionalen Raum, und die zweite Hauptkomponente die y-Koordinate [35]. So können auch Daten, die ursprünglich viele Hunderte Variablen beinhalteten, als Punkte in einem Raum mit nur zwei Dimensionen abgebildet werden. Dabei wird jedoch immer nur ein Teil der Varianz der Daten beibehalten, je mehr Variablen im hochdimensionalen Raum allerdings hohe Kovarianzen aufweisen, desto mehr Varianz kann nach der Dimensionsreduktion durch die Hauptkomponentenanalyse weiterhin abgebildet werden [35].

2.4.3 Lineare Diskriminanzanalyse

Die lineare Diskriminanzanalyse [32] ist ein überwachtes Dimensionsreduktionsverfahren [167]. Zusätzlich zu den hochdimensionalen Daten ist für die Durchführung der linearen Diskriminanzanalyse also auch eine Zuordnung der Datenpunkte in Klassen notwendig [167]. Ziel der linearen Diskriminanzanalyse ist es, einen niedrigerdimensionalen Raum zu finden, welcher die Separation der Datenpunkte unterschiedlicher Klassen maximiert [167]. Abbildung 2.4 zeigt dieselbe multivariate Datenverteilung wie Abbildung 2.3 jedoch ist jeder Datenpunkt zusätzlich einer von zwei Klassen zugeordnet. Weiterhin zeigt Abbildung 2.4, dass die erste lineare Diskriminante (LD1) orthogonal zu der unbekannten Entscheidungsgrenze zwischen beiden Klassen verläuft, und nicht entlang der größten Varianz.

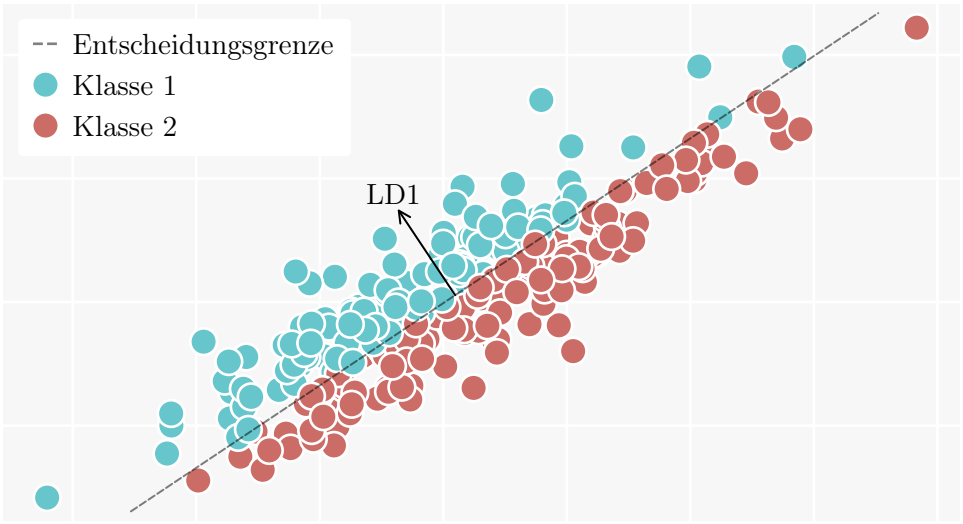

Abbildung 2.4: Die erste lineare Diskriminante (LD1) einer multivariaten Datenverteilung mit zwei Klassen von Daten. LD1 verläuft orthogonal zu der unbekannten Entscheidungsgrenze zwischen beiden Klassen.

Dadurch, dass LD1 orthogonal zur Entscheidungsgrenze verläuft, wird durch eine Transformation der Daten in den eindimensionalen Raum, die Separation der beiden Klassen entlang der neuen Koordinatenachse maximiert. Die maximale Anzahl an Dimensionen, welche durch die lineare Diskriminanzanalyse bestimmt werden können, ist dabei immer niedriger als die Anzahl an Klassen der Daten [35]. Daher kann für die zwei Klassen in Abbildung 2.4 nur eine einzige lineare Diskriminante bestimmt werden. Für die Durchführung der linearen Diskriminanzanalyse sind, wie zuvor bei der Hauptkomponentenanalyse, Verfahren der linearen Algebra notwendig [167]. Im Folgenden wird die Berechnung der linearen Diskriminante für eine Datenverteilung mit zwei Klassen A und B beschrieben, die formale Notation basiert dabei auf Xanthopoulos et al. [167]. Zunächst seien die p-dimensionalen Zentroide über alle Datenpunkte innerhalb Klasse A und B durch $\overline{A}$ und $\overline{B}$ folgendermaßen definiert.

$$\overline{A} = \frac{1}{|A|} \sum_{a \in A} a \,, \;\; \overline{B} = \frac{1}{|B|} \sum_{b \in B} b \tag{2.16}$$

Dabei entsprechen $|A|$ und $|B|$ der Anzahl an Datenpunkten $a \in A$ und $b \in B$ in den Klassen A und B [167]. Für die Durchführung der linearen Diskriminanzanalyse werden anschließend die positiv semidefiniten Streuungsmatrizen S_A und S_B (engl. *Scatter Matrix*) wie folgt berechnet [167].

$$S_A = \sum_{a \in A} (a - \overline{A})(a - \overline{A})^\top \,, \;\; S_B = \sum_{b \in B} (b - \overline{B})(b - \overline{B})^\top \tag{2.17}$$

Die Matrizen S_A und S_B repräsentieren dabei die Streuung der Datenpunkte $a \in A$ und $b \in B$ innerhalb der jeweiligen Klassen von A und B [167]. Die Streuungsmatrix S_{AB} gibt hingegen die Streuung der Datenpunkte zwischen den Klassen A und B an, und wird folgendermaßen berechnet [167].

$$S_{AB} = (\overline{A} - \overline{B})(\overline{A} - \overline{B})^\top \tag{2.18}$$

Ein Kriterium der linearen Diskriminanzanalyse ist es, dass die Datenpunkte innerhalb ihrer eigenen Klasse im niedrigerdimensionalen Raum eine möglichst geringe Streuung aufweisen [167]. Formal muss dafür eine Hyperebene, welche durch einen Vektor ϕ definiert ist, gefunden werden, die das folgende Kriterium minimiert [167].

$$\textbf{Erstes Kriterium:} \;\; \min_{\phi} \;\; \phi^\top (S_A + S_B) \phi \tag{2.19}$$

Zugleich soll die Streuung zwischen den Datenpunkten der beiden Klassen maximiert werden [167]. Formal soll ϕ also das folgende zweite Kriterium der linearen Diskriminanzanalyse maximieren [167].

$$\textbf{Zweites Kriterium:} \;\; \max_{\phi} \;\; \phi^\top S_{AB} \phi \tag{2.20}$$

Fisher's Kriterium [32] maximiert die Streuung der Datenpunkte zwischen den Klassen und minimiert zugleich die Streuung der Datenpunkte innerhalb der Klassen. Dafür wird sowohl das erste als auch das zweite Kriterium der linearen Diskriminanzanalyse wie folgt berücksichtigt [167].

$$\textbf{Fisher's Kriterium:} \max_{\phi} \frac{\phi^\top S_{AB} \phi}{\phi^\top (S_A + S_B) \phi} \tag{2.21}$$

Um Fisher's Kriterium zu erfüllen, muss der Eigenvektor ϕ mit dem größten Eigenwert λ für das folgende Eigenwertproblem bestimmt werden [167].

$$S_{AB}\phi = \lambda(S_A + S_B)\phi \tag{2.22}$$

Die Lösung des Eigenwertproblems wurde dabei bereits in Kapitel 2.4.2 im Rahmen der Hauptkomponentenanalyse beschrieben, und kann bei der linearen Diskriminanzanalyse ebenso angewendet werden. Der resultierende Eigenvektor ϕ mit dem größten Eigenwert λ wird die erste lineare Diskriminante (LD1) genannt [65]. Die lineare Diskriminante kann wie die Hauptkomponenten der Hauptkomponentenanalyse genutzt werden, um die Datenpunkte in den niedrigerdimensionalen Raum zu transformieren (vgl. Kapitel 2.3). Durch diese Transformation der ursprünglichen Datenpunkte mit LD1 werden zuvor festgelegten Kriterien für eine maximale räumliche Separation der Klassen A und B erfüllt.

2.5 Logistische Regressionsanalyse

Während bei einer linearen Regressionsanalyse eine abhängige numerische Variable vorhergesagt werden soll (z. B. das Alter einer Person anhand der Körpergröße und des Geschlechts), geht es bei der logistischen Regressionsanalyse um die Vorhersage einer abhängigen Variable Y, welche nur zwei diskrete Ausprägungen besitzt (z. B. ob eine Person einer Risikogruppe angehört oder nicht, aufgrund des Alters und der Vorerkrankungen) [3].

2.5.1 Linearer Prädiktor

Wie bei einer linearen Regressionsanalyse müssen auch für die logistische Regressionsanalyse zunächst die Prädiktorvariablen $X = \{x_1, x_2, \ldots, x_m\}$ festgelegt werden. Diese sind eine Teilmenge $m < p$ von p unabhängigen Variablen einer Datenbasis [3]. Um die Informationen der Prädiktorvariablen zu kombinieren, wird jeder Prädiktorvariable zunächst ein Regressionskoeffizient in $\beta = \{\beta_0, \beta_1, \beta_2, \ldots, \beta_m\}$ durch die logistische Regressionsanalyse zugeordnet [134]. Dadurch ergibt sich das folgende Produkt [3].

$$\omega = \beta_0 + \sum_{i=1}^{m} \beta_i x_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m \tag{2.23}$$

Dabei wird ω als linearer Prädiktor bezeichnet, welcher die Prädiktorvariablen x_i anhand der Regressionskoeffizienten β_i auf einen einzelnen numerischen Wert abbildet [134]. β_0 justiert dabei den Wert des linearen Prädiktors für den Fall $\{x_1, x_2, \ldots, x_m\} = 0$.

2.5.2 Vorhersage der abhängigen Variable

Die Werte des linearen Prädiktors liegen im Intervall $[-\infty, \infty]$. Um eine Wahrscheinlichkeit der abhängigen Variable $P(Y = 1 \mid X)$ im Intervall [0, 1] vorherzusagen, wird ω durch den nachstehenden Term normiert [134].

$$P(Y = 1 \mid X) = \frac{e^{\omega}}{1 + e^{\omega}} \tag{2.24}$$

Somit kann die Wahrscheinlichkeit, dass die abhängige Variable Y für einen Datenpunkt ausgeprägt ist $Y = 1$ unter Kenntnis der Werte von X für diesen Datenpunkt, durch $P(Y = 1 \mid X)$ angegeben werden [3]. Die Gegenwahrscheinlichkeit ergibt sich folglich als $P(Y = 0 \mid X) = 1 - P(Y = 1 \mid X)$, und gibt die Wahrscheinlichkeit an, dass die abhängige Variable Y für einen Datenpunkt nicht ausgeprägt ist. Für $P \geq 0.5$ wird folglich $Y = 1$ vorhergesagt, während für $P < 0.5$ stattdessen $Y = 0$ vorhergesagt wird [3]. Das Ziel der logistischen Regressionsanalyse ist es also, die Regressionskoeffizienten β_i so zu wählen, die abhängige Variable Y unter Kenntnis der Prädiktorvariablen X korrekt vorhergesagt wird [3]. Für Datenpunkte, deren Ausprägungen der abhängigen Variable Y bekannt sind, kann so zudem geprüft werden, ob eine vollständige Separation der Datenpunkte anhand der Prädiktorvariablen X möglich ist. Das bedeutet, es kann geprüft werden, ob die in X enthaltenen Informationen ausreichend sind, um jeden Datenpunkt seiner korrekten Ausprägung $Y = 0$ oder $Y = 1$ zuordnen zu können.

Kapitel 3

Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten zusammengefasst, welche sich thematisch oder methodisch mit den Inhalten dieser Masterarbeit vergleichen lassen. Zunächst werden verwandte Arbeiten im Bereich der Stimmungsanalyse in Softwareprojekten beschrieben. Anschließend wird auf verwandte Arbeiten eingegangen, welche ebenfalls eine Clusteranalyse, mit dem Ziel Softwareentwickler nach bestimmten Kriterien zu gruppieren, anwenden. Abschließend wird diese Arbeit von den verwandten Arbeiten abgegrenzt.

3.1 Stimmungsanalyse in Softwareprojekten

Das Wirth'sche Gesetz beschreibt das Phänomen, dass Software schneller langsamer wird, als Hardware schneller wird [165]. Dies unter anderem dem Fakt geschuldet, dass moderne Softwaresysteme immer mehr essenzielle Komplexität beinhalten, da immer mehr Funktionalität vom Kunden erwartet wird [7]. Im Zuge dessen stößt das Software Engineering an seine Grenzen, da Softwaresysteme schneller größer und komplexer werden, als neue Methoden zur Bewältigung der Komplexität entwickelt werden [7]. Die nachfolgenden verwandten Arbeiten beschäftigen sich mit der Anwendung der Stimmungsanalyse in Softwareprojekten und erläutern zudem, warum diese ein vielversprechender Ansatz ist, um die steigende Komplexität der Softwareprojekte zu bewältigen.

In ihrem Fachartikel „*Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts*“ analysieren Murgia et al. [101], ob Artefakte aus der professionellen Softwareentwicklung, wie Problemberichte, Emotionen transportieren. Die Motivation für diese Analysen ist trivial: Dort, wo Menschen miteinander zusammenarbeiten, werden unweigerlich auch Emotionen hervorgerufen, welche sich positiv oder negativ auf die Zusammenarbeit auswirken können [101]. So sind unter anderem hitzige Diskussion in der Mailing-Liste der Linux-Kernel-Entwickler keine Seltenheit [101]. Daher ist es notwendig, diese Emotionen zu betrachten, um

vollständig zu verstehen, wie Entwickler innerhalb von Softwareprojekten agieren [101]. Eine rein rationale Betrachtung ist nicht ausreichend, so Murgia et al. [101]. Die Psychologie zeigt beispielsweise, dass eine positivere Stimmung zu mehr Kreativität führt [34]. Kreativität ist wiederum essenziell für das erfolgreiche Design und die Entwicklung von Software [7]. Unabhängig vom Talent eines Entwicklers gilt hingegen auch, dass dieser wahrscheinlich ein Projekt verlassen wird, sofern er mit seiner Umgebung oder seinen Kollegen unzufrieden ist [101]. Dies gibt einen Anreiz für Projektleiter, sich mit den Emotionen ihrer Entwickler auseinanderzusetzen [101]. In einem ersten Schritt zur Ermöglichung einer solchen Technologie führen Murgia et al. [101] eine Studie mit Master-Studierenden und Doktoranden der Informatik zweier Universitäten durch. Die Aufgabe der Studienteilnehmer bestand darin, Fehlerberichte aus dem Fehlersystem der *Apache Software Foundation* mit zutreffenden Emotionen zu annotieren. Dafür standen die sechs Kategorien von primären Emotionen aus dem Framework von Shaver et al. [140] zur Verfügung, namentlich *Liebe*, *Freude*, *Überraschung*, *Wut*, *Trauer* und *Angst*. Die erste Forschungsfrage diente dabei dazu, festzustellen, ob die Studienteilnehmer in der Lage sind über die An- oder Abwesenheit von Emotionen in den Fehlerberichten einig zu sein [101]. Dabei kommen Murgia et al. [101] zu dem Schluss, dass dies für die Emotionen *Liebe (durch die sekundäre Emotion Dankbarkeit)*, *Freude* und *Trauer* der Fall ist. Die zweite Forschungsfrage untersuchte, ob die Ergebnisse der ersten Forschungsfrage durch die zusätzliche Präsentation des Kontextes der jeweiligen Fehlerberichte verbessert werden können [101]. Hierbei berichten Murgia et al. [101] von einem interessanten Ergebnis: Durch den zusätzlichen Kontext eines Problemberichtes sind sich die Studienteilnehmer weniger über die transportierten Emotionen einig als zuvor [101]. Dennoch, durch die Beantwortung ihrer ersten Forschungsfrage zeigen Murgia et al. [101], dass Artefakte der professionellen Softwareentwicklung Emotionen transportieren, und potenzielle Mitglieder eines Entwicklungsteams in der Lage sind, diese Emotionen kohärent zu identifizieren.

Islam und Zibran [62] untersuchten in ihrem Fachartikel „*Towards Understanding and Exploiting Developers' Emotional Variations in Software Engineering*“ die unterschiedlichen Emotionen von Entwicklern in Open-Source-Softwareprojekten. Dafür wurden die 50 größten Open-Source-Softwareprojekte auf *GitHub* analysiert und deren insgesamt mehr als 490.000 Commit-Nachrichten analysiert [62]. Islam und Zibran [62] untersuchten ihre Datenbasis mit dem Stimmungsanalysetool *SentiStrength* [150] auf transportierte Emotionen, wobei sie zusätzlich verschiedenen äußere Umstände betrachteten. Zu diesen Umständen zählten dabei unterschiedliche Arten von Entwicklungsaktivitäten (z. B. Fehlerbehebung, Implementierung neuer Funktionen und Refactoring), sowie verschiedene Wochentage und Tageszeiten [62]. Dabei fanden Islam und Zibran [62] heraus, dass die

Entwickler bei der Fehlerbehebung und dem Refactoring signifikant öfter *positive* Emotionen ausdrücken. Im Gegensatz dazu enthielten die Commit-Nachrichten der Entwickler während der Implementation neuer Funktion einen signifikant höheren Anteil an *negativen* Emotionen [62]. Durch eine hierarchische Clusteranalyse konnten Islam und Zibran [62] zudem drei Gruppen von Entwicklern identifizieren, die unterschiedliche Emotionen während der Fehlerbehebung ausdrücken (überwiegend *negativ*, überwiegend *positiv* und gleichermaßen *negativ* und *positiv*). Die Clusteranalyse stand dabei jedoch nicht im Vordergrund der Arbeit, und wurde nur beiläufig erörtert [62]. Zwischen allen Paaren von je zwei verschieden Wochentagen fanden Islam und Zibran [62] keine signifikanten Unterschiede in der Verteilung der *negativen* und *positiven* Emotionen. Selbiges gilt auch für die betrachteten Kategorien von Tageszeiten (vor 9:00 Uhr, zwischen 9:00 und 17:00 Uhr, und nach 17:00 Uhr) [62]. Letztlich untersuchten Islam und Zibran [62] den Zusammenhang zwischen der Länge einer Commit-Nachricht und der emotionalen Aktivität der Entwickler. Als emotionale Aktivität galt dabei der Ausdruck von sowohl stark *positiven* als auch stark *negativen* Emotionen im Rahmen der Commit-Nachricht [62]. Dabei fanden Islam und Zibran [62] heraus, dass Entwickler deutlich längere Commit-Nachrichten verfassen, wenn sie gefühlsmäßig aktiv sind, als sonst [62].

3.2 Clusteranalyse in Softwareprojekten

In ihrem Fachartikel „*Characterizing Software Developers by Perceptions of Productivity*“ untersuchten Meyer et al. [97] die Wahrnehmung von Produktivität verschiedener Entwickler durch eine Studie mit anschließender Clusteranalyse. Obwohl es viele verschiedene Maße zur Messung der Produktivität gibt (z. B. *Lines of Code* oder das *Function-Point-Verfahren* [67]), gibt es wenig Wissen über die Variationen und Gemeinsamkeiten der Wahrnehmung von Produktivität durch die betreffenden Entwickler selbst [97]. Um dies zu ändern, befragten Meyer et al. [97] die teilnehmenden Entwickler nach kurzen Definitionen von produktiven und unproduktiven Arbeitstagen. Anschließend wurde die Übereinstimmung der teilnehmenden Entwickler mit vordefinierten Aussagen über die Produktivität (z. B. „*Ich fühle mich produktiv, wenn ich programmiere.*“) durch eine fünfstufige Likert-Skala gemessen [97]. Abschließend wurden die teilnehmenden Entwickler dazu befragt, welche Metriken ihnen bei der reflektierenden Selbsteinschätzung der Produktivität einer vergangenen Fünftagewoche helfen würden [97]. Dafür sollten ebenfalls vordefinierte Metriken mit einer fünfstufige Likert-Skala bewertet werden [97]. Unter den einzuschätzenden Metriken waren beispielsweise „*Wie viele Stunden ich programmiert habe.*“ oder „*Wie viele E-Mails ich versendet habe.*“ vertreten [97]. An der Umfrage nahmen insgesamt 413 professionelle Softwareentwickler von *Microsoft* teil, die durchschnittlich mehr als neun Jahre Berufserfahrung hatten [97]. Durch eine Clusteranalyse

auf Basis der Übereinstimmung der teilnehmenden Entwickler mit den verschiedenen Aussagen über die Produktivität fanden Meyer et al. [97] sechs Cluster von Entwicklern. Die gefundenen Cluster wurden anschließend mithilfe der anderen Umfrageantworten charakterisiert [97]. Unter den sechs verschiedenen Arten von Entwicklern fanden Meyer et al. [97] so unter anderem den *einsamen Entwickler*, welcher Gespräche, Meetings und E-Mails als Störungen empfindet und sich am produktivsten fühlt, wenn er allein und ununterbrochen programmieren kann. Umgekehrt verhält es sich mit dem *sozialen Entwickler*, welcher sich am produktivsten fühlt, wenn er Kollegen helfen kann oder mit ihnen zusammenarbeitet (z. B. bei Code Reviews). Ein wichtiges Ergebnis von Meyer et al. [97] ist, dass alle diese unterschiedlichen Typen von Entwicklern, auch unterschiedliche Metriken als Feedback bezüglich ihrer Produktivität bevorzugen. So bevorzugt der *einsame Entwickler* Feedback darüber, wie oft und wie lange er jeweils von seiner Arbeit unterbrochen wurde, während der *soziale Entwickler* die Zeit, die er mit dem Programmieren verbrachte, als Metrik bevorzugt. Abschließend gaben Meyer et al. [97] diverse Empfehlungen für Projektleiter, basierend auf ihren Ergebnissen. So ist es wichtig einen abgetrennten ruhigen Bereich in einem Büro zur Verfügung zu stellen, damit sich die *einsamen Entwickler* produktiv fühlen können, während für die *sozialen Entwickler* offene Büroflächen besser geeignet sind.

Di Bella et al. [21] untersuchten mittels einer Clusteranalyse das Verhalten und die Beiträge von Entwicklern in Open-Source-Projekten. Die Motivation bestand dabei darin, zu verstehen, welche Mitglieder in Entwicklungsteams wichtiger sind als andere und wer die eigentlichen „*Hauptentwickler*“ sind [21]. Da viele Unternehmen auf Open-Source-Projekten aufbauen, ist das Verhalten dieser Entwickler für sie von Interessen. [21]. Die Identifikation solcher Entwickler in Open-Source-Softwareprojekten eines Unternehmens kann z. B. sinnvoll sein, um diese Entwickler fest anzustellen [21]. Di Bella et al. [21] betrachteten daher zehn Open-Source-Softwareprojekte, wobei diese in Domäne, Programmiersprache und Größe variierten. Anschließend wurden für alle Entwickler innerhalb eines jeden Projektes eine Vielzahl von Metriken berechnet [21]. Diese umfassten etwa die Anzahl an Commits zu dem Projekt und die durchschnittliche Zeit zwischen Commits [21]. Anhand dieser Metriken wurden die Entwickler in den Projekten anschließend geclustert, wobei vier Gruppen von Entwicklern identifiziert werden konnten [21]. Dazu gehörten unter anderem die *gelegentlichen Entwickler*, die einen Großteil der Entwickler ausmachen und nur sporadisch etwas zu ihren Projekten beitragen [21]. Hauptsächlich fokussieren sie sich dabei zudem auf einzelne Dateien [21]. Im Gegensatz dazu stehen die *zentralen Entwickler*: Sie tragen die Open-Source-Projekte maßgeblich auf ihren Schultern und liefern die wichtigsten Beiträge, allerdings machen sie auch nur einen kleinen Teil der Entwickler aus [21]. Di Bella et al. [21] nutzten zudem

die Hauptkomponentenanalyse [66], um die Ergebnisse des Clusterings zu visualisieren. Dabei erkannten sie, dass die *zentralen Entwickler* ein deutlich abgegrenztes Cluster im Vergleich zu den anderen drei Gruppen von Entwicklern bildeten [21]. Innerhalb aller zehn unterschiedlichen Open-Source-Projekte konnten die vier Arten von Entwicklern identifiziert werden, wobei es jedoch deutliche Unterschiede in deren Häufigkeitsverteilungen gab [21]. Di Bella et al. [21] argumentierten, dass eine Überwachung der Arten von Entwicklern Aufschluss über den Zustand eines Projektes, die potenzielle Zukunft des Projektes, und die Frage, ob eine Investition in das Projekt (in Form von Zeit oder Geld) sinnvoll ist, geben kann [21]. Zusätzlich ist diese Methodik sinnvoll, um die mögliche Anzahl an Kombinationen von Interaktionen, die im Rahmen einer sozialen Netzwerkanalyse [96] beobachtet werden müssen, zu reduzieren, indem die weniger wichtigen Entwickler für die Berechnungen außen vor gelassen werden. [21].

3.3 Abgrenzung dieser Arbeit

Obwohl sich in dieser Arbeit viele der einzelnen Aspekte der zuvor beschriebenen verwandten Arbeiten wiederfinden, verknüpft diese Arbeit die methodischen Aspekte, wie die Clusteranalyse, mit der Motivation für die Anwendung von Stimmungsanalyse in Softwareprojekten, was so in noch keiner Arbeit durchgeführt wurde. Durch die Arbeiten von Murgia et al. [101], sowie Islam und Zibran [62] ist bekannt, dass Emotionen eine entscheidende Rolle für die Zusammenarbeit in Entwicklungsteams spielen, diese Emotionen von Entwicklern identifizierbar sind, und zwischen verschiedenen Entwicklungsaktivitäten variieren. Im Rahmen dieser Arbeit soll nun aber ein Perspektivwechsel auf die persönliche Ebene der einzelnen Entwickler vollzogen werden: Anstatt den Einfluss von äußeren Umständen zu untersuchen, sollen die allgemeinen Unterschiede in der Wahrnehmung der Stimmung von Aussagen zwischen verschiedenen Entwicklern untersucht werden. Damit ähnelt diese Arbeit methodisch derer von Meyer et al. [97], welche jedoch die Wahrnehmung der Produktivität der Entwickler anstelle von der Wahrnehmung der Stimmung untersuchte. Mit Methodiken, wie der Dimensionsreduktion und Clusteranalyse, die unter anderem auch von Di Bella et al. [21] angewendet wurden, soll die von Entwicklern wahrgenommene Stimmung zu verschiedenen Aussagen, aus der Domäne der kollaborativen Softwareentwicklung, untersucht werden. Falls sich dabei unterschiedliche Gruppierungen von Entwicklern mit einer ähnlichen Wahrnehmung der Stimmung identifizieren lassen, sollen diese zudem so weit wie möglich charakterisiert werden. Diese Arbeit vereint also die bisherigen Motivationen und Kenntnisse über die Stimmungsanalyse im Software Engineering mit einer zuvor so noch nicht in diesem Bereich angewendeten Methodik, um Aufschluss über die Unterschiede in der Wahrnehmung der Stimmung zwischen einzelnen Entwicklern zu erhalten.

Kapitel 4

Forschungsdesign

In diesem Kapitel werden die Rohdaten, die Datenbasis, sowie Forschungsmethoden, welche im Rahmen dieser Arbeit auf die Datenbasis angewendet wurden, beschrieben. Abbildung 4.1 gibt einen Überblick über die wichtigsten dieser Forschungsmethoden, und ihre Abhängigkeiten untereinander.

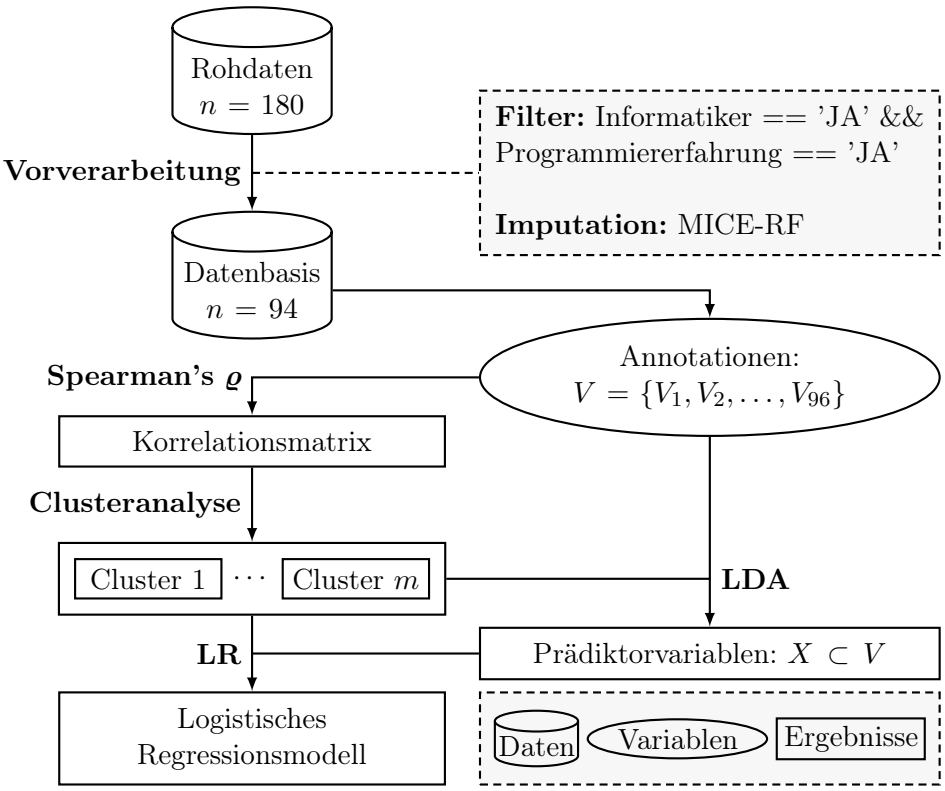

Abbildung 4.1: Überblick über die wichtigsten Methoden des Forschungsdesigns, sowie deren Abhängigkeiten untereinander.

Wie Abbildung 4.1 zeigt, werden zunächst in Kapitel 4.1 die Rohdaten dieser Arbeit [111] beschrieben, wobei auf den Aufbau der Umfrage und die Erhebung der Antworten eingegangen wird. Kapitel 4.2 beschreibt, wie der Rohdatensatz vorverarbeitet wurde, um die im weiteren Verlauf dieser Arbeit betrachtete Datenbasis von $n = 94$ Studienteilnehmern zu erhalten. Auf Basis der Korrelationen (vgl. Kapitel 4.3.1) zwischen den Annotationen der 96 Aussagen V_1 bis V_{96} aus der Datenbasis werden die Studienteilnehmer in Kapitel 4.3.2 geclustert. Die dabei entstandenen Teilnehmergruppen (Cluster 1 bis Cluster m) werden in Kapitel 4.3.3 zusammen mit den Annotationen der Aussagen V genutzt, um durch die lineare Diskriminanzanalyse (LDA) eine Teilmenge $X \subset V$ von Aussagen zu bestimmen, deren Annotationen zwischen den Teilnehmergruppen die größten Unterschiede aufweisen. Diese Aussagen X dienen letztlich als Prädiktorvariablen für die logistische Regressionsanalyse (LR), welche in Kapitel 4.3.4 genutzt wird, um ein logistisches Regressionsmodell zur Vorhersage der Teilnehmergruppe eines jeden Studienteilnehmers zu berechnen. Zu den Verfahren, die nicht in Abbildung 4.1 dargestellt sind, gehört die Hauptkomponentenanalyse (vgl. Kapitel 4.3.3), welche als Dimensionsreduktionsverfahren auf die Ergebnisse der Clusteranalyse angewendet wird. Im Anschluss an die Verfahren in Abbildung 4.1 werden zudem in Kapitel 4.3.5 alle im Rahmen der Umfrage erfassten Merkmale der Studienteilnehmer auf Unterschiede zwischen den verschiedenen Teilnehmergruppen untersucht. Dafür werden aufgrund der unterschiedlichen Skalenniveaus und Vorbedingungen verschiedener Merkmalswerte auch verschiedene Testverfahren benötigt, welche am Ende von Kapitel 4.3.5 beschrieben werden.

4.1 Rohdaten

Der verwendete Rohdatensatz aus der Studie von Herrmann et al. [54] ist im Forschungsdatenrepositorium *Zenodo* verfügbar (vgl. Obaidi et al. [111]). Aus diesen Rohdaten wurde die vorverarbeitete Datenbasis dieser Arbeit bestimmt. Im Folgenden wird auf das Erhebungsdesign der zugehörigen Umfrage, die Auswahl der Referenz-Aussagen, sowie die Datenerhebung des Rohdatensatzes eingegangen. Dieser Abschnitt überschneidet sich daher mit den Abschnitten 3.2 und 3.3 aus dem Fachartikel von Herrmann et al. [54].

4.1.1 Erhebungsdesign

Das Erhebungsdesign der Umfrage ist im Anhang dieser Arbeit in Tabelle A.1 als deutschsprachige Übersetzung vom englischsprachigen Original [54] tabellarisch dargestellt. Insgesamt wurden die Studienteilnehmer der Umfrage zu fünf Kategorien befragt, wobei alle Fragen, sowie die zu annotierenden Aussagen selbst auf Englisch präsentiert wurden [54]. Die erste Kategorie (vgl. Tabelle A.1a) umfasste Fragen zu den demografischen Merkmalen der Teilnehmer. Die Teilnehmer wurden nach Angaben zu ihrem Alter

und ihrem Geschlecht gebeten [54]. Anschließend sollten die Teilnehmer angeben, ob Englisch ihre Muttersprache ist, und wie häufig sie auf Englisch kommunizieren [54]. In der zweiten Kategorie (vgl. Tabelle A.1b) wurde festgestellt, ob und inwiefern die Teilnehmer zur Informatik zugehörig sind. Die Teilnehmer sollten angeben, ob sie sich selbst als Informatiker identifizieren, und wie ihr beruflicher Status lautet [54]. Die dritte Kategorie (vgl. Tabelle A.1c) umfasste Fragen zur Berufserfahrung der Teilnehmer. Zunächst wurden die Teilnehmer befragt, ob sie im Allgemeinen Programmiererfahrung haben und wie sie ihre Programmierkenntnisse einschätzen würden [54]. Anschließend wurden die Teilnehmer gefragt, wie viele Jahre sie an professioneller Erfahrung als Entwickler (sowohl mit als auch ohne Entwicklungsteam) haben, und wie familiär sie darin sind als Entwickler in einem Team zu arbeiten [54]. Anschließend folgte die Annotation der Aussagen (vgl. Tabelle A.1d). Hierfür wurden die Teilnehmer darum gebeten, zehn Blöcke von je zehn Aussagen mit den Sentiment-Polaritäten, welche sie bei den jeweiligen Aussagen wahrnehmen, zu annotieren [54]. Sowohl die Reihenfolge der zehn Aussagen-Blöcke als auch die der zehn Aussagen innerhalb der jeweiligen Blöcke wurde für jeden Teilnehmer zufällig festgelegt [54]. Es wurde vorab keine Anweisung dafür gegeben, wie die Teilnehmer die Annotation durchführen sollten [54]. Es blieb den Teilnehmern also freigestellt, woran sie die Sentiment-Polaritäten der einzelnen Aussagen festmachten [54]. Im Anschluss an die Annotation der Aussagen wurden die Teilnehmer dazu befragt, nach welchen Kriterien sie die Annotation vorgenommen haben (vgl. Tabelle A.1e). Als Auswahlmöglichkeiten standen dafür sowohl der Inhalt der Aussage (z. B. Berichte über negative Ereignisse im Sinne der Softwareentwicklung), als auch die Tonalität (z. B. abwertende oder unangemessene Formulierungen) zur Verfügung, sowie ein Feld für Freitext-Antworten [54]. Einschließlich einiger gespeicherter Metadaten (z. B. Datum, Uhrzeit, Dauer der Umfrage etc.), wurden 126 abhängige und unabhängige Variablen [133] pro Studienteilnehmer erfasst [111].

4.1.2 Selektion der Aussagen für das Erhebungsdesign

Die Auswahl der insgesamt 100 Aussagen, welche von jedem Teilnehmer der Umfrage [111] annotiert werden sollten, ergab sich wie im Folgenden beschrieben. Die einzelnen Aussagen entstammten je einem von zwei verschiedenen Datensätzen [84, 107], die von wissenschaftlichen Autoren annotiert und publiziert worden sind [54]. Dabei wurde ein Datensatz ausgewählt, für dessen Erstellung die Autoren angaben, ein Emotions-Framework verwendet zu haben, und ein Datensatz, für welchen stattdessen eine ad-hoc Annotation erfolgte. Ein Emotions-Framework dient bei der Erstellung des Datensatzes als eine Richtlinie, nach welchen Kriterien die Aussagen mit den Sentiment-Polaritäten *negativ*, *neutral* und *positiv* (oder feingranulareren Emotionen) zu annotieren sind. Dieses Vorgehen verspricht eine höhere Konsistenz des Datensatzes, wenn mehrere Personen die manuelle

Annotation anhand des Frameworks vornehmen [109]. Durch die Verwendung beider Datensätze in der aktuellen Forschung über Stimmungsanalyse im Software Engineering [12, 166, 169] sind diese eine repräsentative Datenquelle für die ursprüngliche Umfrage [111] aus der Studie von Herrmann et al. [54], sowie für die weiterführenden Untersuchungen, welche im Rahmen dieser Arbeit angewendet werden.

Emotions-Framework-Datensatz

Beim ersten Datensatz handelt es sich um den von Novielli et al. [107] erstellten *GitHub-„Goldstandard"*-Datensatz [107]. Dieser Datensatz wurde aus der Mehrheitsentscheidung [115] von drei Autoren unter Verwendung des Emotions-Frameworks von Shaver et al. [140] annotiert. Das Shaver-Framework [140] gab den Autoren dabei klare Richtlinien, die besagten, wann eine Aussage den Emotionen *Liebe*, *Freude*, *Trauer*, *Wut*, *Angst* und *Überraschung* zuzuordnen war. Die so festgelegten Emotionen der Aussagen wurden dann je einer der drei Sentiment-Polaritäten *positiv*, *neutral*, *negativ* zugeordnet. Dabei wurden *Liebe* und *Freude* je der Kategorie *positiv* zugeordnet, während *Trauer*, *Wut*, und *Angst* der Kategorie *negativ* zugeordnet wurden [107]. Die Emotion *Überraschung* konnte nicht eindeutig einer Sentiment-Polarität zugeordnet werden und wurde daher verworfen [107]. Alle Aussagen, die keiner der sechs Emotionen aus dem Shaver-Framework [140] zugeordnet werden konnten, erhielten die Sentiment-Polarität *neutral* [107]. Der so entstandene Datensatz von Novielli et al. [107] umfasst 7122 manuell annotierte Aussagen der Plattform *GitHub*.

Ad-hoc-Datensatz

Für den zweiten Datensatz wurden von Lin et al. [84] Aussagen von der Plattform *Stack Overflow* extrahiert und manuell mit den drei Sentiment-Polaritäten *positiv*, *neutral*, *negativ* annotiert. Im Gegensatz zu Novielli et al. [107] nutzen die Autoren dafür keine festgelegten Richtlinien, sondern annotierten die Aussagen ad-hoc [54]. Insgesamt wurden 1500 Aussagen von den Autoren annotiert, bevor der Datensatz veröffentlicht wurde [84].

Selektion der Aussagen aus den Referenz-Datensätzen

Aus jedem der beiden Referenz-Datensätze [84, 107] wurden je 16 *negative*, *neutrale*, und *positive* Aussagen zufällig ausgewählt [54]. Dadurch ergeben sich insgesamt 48 zufällige Aussagen pro Datensatz und 96 zufällige Aussagen insgesamt. Aus den bereits ausgewählten 48 Aussagen des GitHub Datensatzes [107] wurden zufällig zwei *positive*, und je eine *neutrale* und *negative* Aussage ausgewählt, welche doppelt in die Umfrage eingefügt wurden, um die Intrarater-Reliabilität [47] der Teilnehmer zu überprüfen [54]. Dadurch ergeben sich insgesamt 100 Aussagen mit ausgewogenen Verhältnissen der drei Sentiment-Polaritäten (lt. den Annotationen der Autoren der Datensätze). Im weiteren Verlauf dieser Arbeit werden sich alle Datenanalysen auf die 96 verschiedenen Aussagen ohne die vier Duplikate beschränken.

4.1.3 Datenerhebung

Die Umfrage wurde mittels *LimeSurvey* auf dem Server des Fachgebiets Software Engineering der Leibniz Universität Hannover veröffentlicht [54]. Die Einladungen an der Umfrage teilzunehmen wurden primär über E-Mails verschickt [54]. Bachelor- und Masterstudierende im Studiengang Informatik wurden über die E-Mail-Verteiler der Veranstaltungen des Fachgebiets Software Engineering gebeten, an der Umfrage teilzunehmen [54]. Die E-Mail-Adressen von Doktoranden, Postdoktoranden und wissenschaftliche Hilfskräfte im Bereich der Informatik wurden von institutionellen Webseiten extrahiert, um diese ebenfalls auf die Umfrage aufmerksam zu machen [54]. Des Weiteren wurden Autoren von Publikationen im Bereich der Stimmungsanalyse im Software Engineering über die in ihren Publikationen angegebenen E-Mail-Adressen kontaktiert und darum gebeten, die Umfrage in ihrem Forschungsnetzwerk zu verbreiten [54]. Die relevanten Autoren wurden dabei aus der systematischen Literaturrecherche von Obaidi und Klünder [112] extrahiert [54]. Zusätzlich wurde die Umfrage in den sozialen Netzwerken *Twitter*, *Facebook*, *LinkedIn*, und *XING* verbreitet [54]. In der Einladung wurde ausdrücklich darauf hingewiesen, dass sich die Umfrage nur an Informatiker mit Programmiererfahrung richtet [54]. Zusätzlich wurde in der Einladung beschrieben, dass diese Umfrage dazu dient, die Wahrnehmung von Informatikern bezüglich der Stimmung in Aussagen, aus der Domäne der kollaborativen Softwareentwicklung, besser zu verstehen. Eine Teilnahme an der Umfrage war von April 2021 bis November 2021 möglich [54]. Insgesamt umfasst der Rohdatensatz der Umfrageergebnisse 180 Datenpunkte [111].

4.2 Datenvorverarbeitung

Um den Rohdatensatz von Obaidi et al. [111] aus der Studie von Herrmann von et al. [54] für die Forschungsmethoden im weiteren Verlauf dieser Arbeit sinnvoll nutzen zu können, müssen die Rohdaten zunächst vorverarbeitet werden. Im Folgenden werden die Auswahl der Strichprobe aus den Rohdaten, sowie die Imputation fehlender Annotationen beschrieben, welche dazu dienten, die Datenbasis dieser Arbeit zu bestimmen.

4.2.1 Selektion der Stichprobe

Da diese Arbeit die Wahrnehmung der Stimmung in Softwareprojekten untersucht, sollte die untersuchte Stichprobe auch nur potenzielle Mitglieder eines Entwicklungsteams enthalten. Deshalb mussten zunächst Studienteilnehmer aus dem Rohdatensatz [111] entfernt werden, die aufgrund ihrer Antworten in der Umfrage nicht als potenzielle Mitglieder eines Entwicklungsteams angesehen werden konnten. Dafür wurden aus den Rohdaten im ersten Schritt nur die Studienteilnehmer selektiert, welche die beiden

Fragen „*Identifizieren Sie sich als Informatiker?*“ und „*Haben Sie Erfahrung mit Programmierung?*“ mit *Ja* beantwortet haben. Dadurch reduzierte sich die Anzahl der berücksichtigten Studienteilnehmer um 50 Teilnehmer von 180 auf 130. Um die Wahrnehmung der Stimmung der verbleibenden Teilnehmer zu untersuchen, ist es notwendig, dass diese Teilnehmer die Wahrnehmung ihrer Stimmung angegeben haben. Daher wurden im nächsten Schritt der Datenvorverarbeitung alle die Studienteilnehmer entfernt, welche keine einzige der 96 Aussagen annotiert haben. Dadurch reduzierte sich die Anzahl der berücksichtigten Teilnehmer um 36 Teilnehmer von 130 auf 94. Für den weiteren Verlauf dieser Arbeit wird genau diese Stichprobe von $n = 94$ Studienteilnehmern betrachtet.

4.2.2 Imputation fehlender Annotationen

Von den insgesamt $94 \cdot 96 = 9024$ möglichen Annotationen der Stichprobe fehlen 2339 Annotation, was einem Anteil von 25.92 % entspricht. Im Folgenden werden mögliche Erklärungen präsentiert, die zum Fehlen dieser Annotationen geführt haben könnten. Anhand dieser Überlegungen wird die Wahl eines geeigneten Imputationsverfahrens begründet und durch einen anschließenden Vergleich verschiedener Verfahren zum Umgang mit den fehlenden Annotationen validiert.

Bezug zu den Mechanismen fehlender Daten

Die drei Mechanismen fehlender Daten (kurz MCAR, MAR und MNAR) welche von Rubin [125] definiert wurden (vgl. Kapitel 2.1.1), werden nachfolgend auf die fehlenden 2339 Annotationen der Studienteilnehmer bezogen. Dies dient dazu, festzustellen, welche Mechanismen fehlender Daten für die fehlenden Annotationen plausibel sind.

MCAR: Dass diese Annotationen komplett zufällig fehlen, wäre denkbar, wenn die jeweiligen Annotationen von betreffenden Studienteilnehmern schlichtweg vergessen worden sind. Ein komplett zufälliges Fehlen der Daten wäre auch dann gegeben, wenn die Umfrage von den betreffenden Studienteilnehmern bewusst vor Abschluss abgebrochen wurde, da die Reihenfolge der zu annotierenden Aussagen für jeden Studienteilnehmer zufällig festgelegt wurde (vgl. Kapitel 4.1.1).

MAR: Ein zufälliges Fehlen dieser Annotationen wäre gegeben, wenn das Fehlen der Annotationen eines Studienteilnehmers mit anderen beobachteten Variablen zusammenhängt. Zu den beobachteten Variablen, die dafür infrage kommen, gehören die Berufserfahrung, die Programmierkenntnisse, und Familiarität mit der Arbeit in Entwicklungsteams. Es wäre etwa denkbar, dass ein Studienteilnehmer aufgrund mangelnder Berufserfahrung nicht in der Lage war, eine präsentierte Aussage ausreichen einzuschätzen und daher keine Annotation vornahm.

MNAR: Da es sich bei Wahrnehmung einer präsentierten Aussage nicht um eine sensible Information handelt, kann ausgeschlossen werden, dass diese Annotation nicht zufällig fehlen. Es wird nicht davon ausgegangen, dass ein Studienteilnehmer eine Annotation verweigert hat, weil er seine Wahrnehmung nicht mitteilen wollte, da die Einladung zur Umfrage eindeutig darauf hingewiesen hat, dass die Annotation der Aussagen Kern der Befragung ist (vgl. Kapitel 4.1.3).

Wahl eines geeigneten Imputationsverfahrens

Es kann also ausgeschlossen werden, dass der Mechanismus MNAR für das Fehlen der Annotationen verantwortlich ist. Es kann jedoch nicht abgegrenzt werden, welcher der verbleibenden Mechanismen (MCAR oder MAR) für das Fehlen der Annotationen verantwortlich sind. Sollten die Annotationen nur zufällig (MAR) und nicht komplett zufällig fehlen (MCAR), dann könnte das Ausschließen unvollständiger Datenpunkte, kurz CCA (engl. *Complete Case Analysis*), die so entstehende Datenbasis verzerren [155]. Aufgrund der Ungewissheit darüber, welcher der beiden Mechanismen (MCAR oder MAR) in den Rohdaten vorliegt, sollten daher auch die unvollständigen Datenpunkte berücksichtigt werden. Für die Anwendung der im folgenden in Kapitel 4.3 beschriebenen Forschungsmethoden können jedoch keine unvollständigen Datenpunkte genutzt werden. Daher ist es nötig, die fehlenden Annotationen zu imputieren. Um die fehlenden Annotationen bestmöglich zu imputieren, wurde das in Kapitel 2.1.4 beschriebene Verfahren MICE-RF [138] ausgewählt. Dieses eignet sich für die Imputation von ordinalen Daten und verspricht die statistische Inferenz der Daten zu erhalten [70]. Damit ist MICE-RF auch für die Imputation der fehlenden Annotation der vielversprechendste Ansatz. Dies hat zudem den Vorteil, dass nachfolgend alle $n = 94$ Datenpunkte der selektierten Stichprobe genutzt werden können und Verfahren wie die Clusteranalyse auf weniger spärlichen Daten bessere Ergebnisse erzielen können [93].

Abschätzung der Güte der Imputation

Um die Güte der Imputation abzuschätzen, werden in der Literatur oftmals Lagemaße wie Mittelwerte und Varianzen der einzelnen Variablen einer Datenbasis vor und nach der Imputation verglichen [171]. Dadurch wird dann etwa deutlich, dass eine Mittelwert Imputation den Mittelwert zwar nicht verändert, aber die Varianz der Daten verzerrt, da immer nur derselbe Wert imputiert wird [171]. Da es sich um bei den Annotationen der Studienteilnehmer um ordinal skalierte Daten handelt, ist eine Berechnung von Mittelwerten und Varianzen nicht möglich. Daher wurde entschieden, stattdessen die relativen Häufigkeiten der Sentiment-Polaritäten *negativ*, *neutral* und *positiv* als Lagemaße für jede Aussage zu betrachten. Als Grundlage für den anschließenden Vergleich wurden diese relativen Häufigkeiten aller 96 Aussagen zunächst anhand der verfügbaren Daten (engl. *Available*

Case Analysis), berechnet. Für jede Aussage V_i, $i \in \{1, 2, \ldots, 96\}$ wurde die relative Häufigkeit $h_i(P)$ einer Sentiment-Polarität P dabei folgendermaßen berechnet [132].

$$h_i(P) = \frac{|P(V_i)|}{|V_i|} \text{ , } P \in \{negativ,\ neutral,\ positiv\} \tag{4.1}$$

$$h_i(negativ) + h_i(neutral) + h_i(positiv) = 1 \tag{4.2}$$

Dabei gibt $|P(V_i)|$ die absolute Häufigkeit der Annotationen der Sentiment-Polarität P für V_i an, während $|V_i|$ die absolute Häufigkeit aller Annotationen von V_i angibt. Um die Verzerrung der ursprünglichen Häufigkeitsverteilungen zu quantifizieren, wurden neben den drei relativen Häufigkeiten $h_i(P)$ jeder Aussage V_i, $i \in \{1, 2, \ldots, 96\}$ der unvollständigen Datenpunkte auch die veränderten relativen Häufigkeiten $\tilde{h_i}(P)$ für jedes der folgenden Verfahren berechnet.

CCA (engl. Complete Case Analysis): Der Datenpunkt eines jeden Studienteilnehmers S_i, $i \in \{1, 2, \ldots 94\}$, welcher nicht alle V_i annotiert hat, wird entfernt. Dadurch sinkt die Anzahl der Datenpunkte auf $n = 45$. Die Annotationen unvollständiger Datenpunkte gehen somit verloren.

Median Imputation: Alle fehlenden Annotationen einer Aussage V_i werden mit dem Median der Sentiment-Polaritäten für diese Aussage imputiert [89]. Folglich entsprechen alle imputierten Annotationen von V_i derselben Sentiment-Polarität P [89].

Zufällige Imputation: Für jede einzelne der 2339 fehlenden Annotationen wird eine Sentiment-Polarität $P \in \{negativ, neutral, positiv\}$ zufällig ausgewählt. Die Imputation einer fehlenden Annotation ist also weder von der Aussage V_i noch dem Teilnehmer S_i abhängig.

MICE-RF: Die fehlenden Annotationen werden unter Berücksichtigung der Annotationen aller Aussagen V_i und aller Studienteilnehmer S_i durch das Verfahren MICE-RF [138] imputiert (vgl. Kapitel 2.1.4).

Nach der Anwendung von jedem dieser vier Verfahren gibt es folglich eine veränderte Verteilung der relativen Häufigkeiten $\tilde{h_i}(P)$, für jede Aussage V_i und jede Sentiment-Polarität P. Um diese mit den ursprünglichen relativen Häufigkeiten $h_i(P)$ zu vergleichen, wurde die Wurzel des mittleren quadratischen Fehlers, kurz RMSE (engl. *Root-Mean-Square Error*) [141], für jede Sentiment-Polarität P wie folgt berechnet.

$$\text{RMSE} = \sqrt{\frac{1}{96} \sum_i \left(\tilde{h_i}(P) - h_i(P)\right)^2} \tag{4.3}$$

Tabelle 4.1 zeigt die resultierenden Fehler in den veränderte Verteilungen der *negativen*, *neutralen* und *positiven* Annotationen, sowie den durchschnittlichen Fehler ø über alle Sentiment-Polaritäten P für die vier zuvor vorgestellten Verfahren. Die Balkendiagramme sind auf den maximalen Fehler der Median Imputation für die Sentiment-Polarität *neutral* normiert.

Tabelle 4.1: Vergleich zwischen den resultierenden Fehlern für die Analyse vollständiger Datenpunkte (CCA), sowie die Verfahren Median Imputation, zufälliger Imputation und MICE-RF.

Verfahren	**RMSE**			
	Negativ	**Neutral**	**Positiv**	**ø**
CCA	0.0372	0.0400	0.0314	0.0364
Median	0.0649	0.1019	0.0601	0.0779
Zufällig	0.0638	0.0586	0.0621	0.0616
MICE-RF	0.0255	0.0438	0.0308	0.0342

Wie Tabelle 4.1 zu entnehmen ist, wird durch die Median Imputation, mit 7.79 % RMSE-Abweichung, die größte Verzerrung der Sentiment-Polaritäten hervorgerufen. Insbesondere die Häufigkeit der Sentiment-Polarität *neutral* wird durch dieses Verfahren um mehr als 10 % verzerrt. Die zufällige Imputation schneidet mit 6.16 % Abweichung am zweitschlechtesten ab. Die CCA ist das zweitbeste Verfahren und schafft es in der Häufigkeit der Sentiment-Polarität *neutral* den niedrigsten Fehler von 4 % RMSE-Abweichung zu verursachen. Durchschnittlich liegt die Abweichung zu ursprünglichen relativen Häufigkeiten der Sentiment-Polaritäten durch die CCA bei 3.64 %, was eine deutliche Verbesserung zu zufälligen Imputation und Median Imputation darstellt. MICE-RF erreicht das beste Ergebnis mit einer durchschnittlichen RMSE-Abweichung von 3.42 %. Insbesondere für die Sentiment-Polarität *negativ* erreicht MICE-RF eine besonders niedrige Abweichung von nur 2.55 % und kann damit einen deutlichen Vorteil zur CCA aufweisen. Insgesamt erreicht MICE-RF für den Durchschnitt und die Sentiment-Polaritäten *negativ* und *positiv* das beste Ergebnis. Tabelle 4.1 lässt darauf schließen, dass die Imputation der fehlenden Annotationen durch MICE-RF eine mindestens genauso valide Verfahrensweise ist, wie der Ausschluss von unvollständigen Datenpunkten (CCA). Dies gilt insbesondere, da die *negativen* und *positiven* Annotationen im weiteren Verlauf dieser Arbeit von höherem Interesse sind und MICE-RF deren Häufigkeitsverteilungen am wenigsten verzerrt. Daher bilden die Datenpunkte der verbleibenden 94 Studienteilnehmer S_i nach der Imputation durch MICE-RF die vollständige Datenbasis dieser Arbeit.

4.3 Datenanalyse

Im Folgenden werden alle Forschungsmethoden, welche auf die zuvor beschriebene Datenbasis der 94 Studienteilnehmer angewendet wurden, beschrieben. Alle Methoden dienen dabei dem Ziel, die verschiedenen Wahrnehmungen der Stimmung von Kommunikation im Software Engineering besser zu verstehen. Die wichtigsten dieser Forschungsmethoden und ihre Abhängigkeiten untereinander wurden in Abbildung 4.1 visualisiert.

4.3.1 Korrelationsanalyse der Wahrnehmung

Um zu quantifizieren, wie stark die Wahrnehmung der Studienteilnehmer im Allgemeinen zusammenhängt, wurde eine Korrelationsanalyse zwischen allen Studienteilnehmer auf Basis ihrer Annotationen durchgeführt. Für die Berechnung der Korrelation wurde Spearman's Rangkorrelationskoeffizient, kurz Spearman's ϱ [145] verwendet (vgl. Kapitel 2.2.2). Spearman's ϱ eignet sich durch die Berücksichtigung der ordinalen Rangfolge der Sentiment-Polaritäten, sowie der Interpretierbarkeit (identisch zu Pearson's r [160]) für den Vergleich der Zusammenhänge in der Wahrnehmung der Stimmung. Ein Maß für die Interrater-Reliabilität wie Cohen's κ [13] ist für die folgenden Analysen ungeeignet, da es von der Verteilung der Annotations-Klassen (in der Datenbasis die Sentiment-Polaritäten *negativ*, *neutral*, und *positiv*) abhängig ist. Das heißt, dass zwei Paare von Studienteilnehmern unterschiedliche κ-Werte erzeugen können, obwohl für beide dieselbe Anzahl an Annotationen übereinstimmt, was zu paradoxen Ergebnissen führen kann [9, 29]. Aufgrund dessen wird Cohen's κ in der Literatur oftmals als unzuverlässig kritisiert [90, 143]. Spearman's ϱ eignet sich zudem ebenfalls als Maß der Interrater-Reliabilität für ordinalskalierte Variablen [18], sowie als Ähnlichkeitsmaß für die im weiteren Verlauf folgende Clusteranalyse [79].

Berechnung der Korrelationsmatrix aller Studienteilnehmer

Um korrelative Strukturen innerhalb der Datenbasis aufzudecken, wird zunächst eine Korrelationsmatrix zwischen den insgesamt 94 Studienteilnehmern S_i, $i \in \{1, 2, \ldots, 94\}$ berechnet. Die Korrelationen über alle 96 Annotationen zwischen jedem Paar von Studienteilnehmern S_i und S_j kann durch $\varrho(S_i, S_j)$ beschrieben werden. Die Korrelationsmatrix aller Studienteilnehmer in der Datenbasis ist folglich durch die nachstehende untere Dreiecksmatrix $\mathrm{Corr}(S)$ gegeben.

$$\mathrm{Corr}(S) = \begin{bmatrix} \varrho(S_1, S_2) & & & & \\ \varrho(S_1, S_3) & \varrho(S_2, S_3) & & & \\ \vdots & \vdots & \ddots & & \\ \varrho(S_1, S_{93}) & \varrho(S_2, S_{93}) & \cdots & \varrho(S_{92}, S_{93}) & \\ \varrho(S_1, S_{94}) & \varrho(S_2, S_{94}) & \cdots & \varrho(S_{92}, S_{94}) & \varrho(S_{93}, S_{94}) \end{bmatrix} \tag{4.4}$$

Die Einträge der Dreiecksmatrix $\mathrm{Corr}(S)$ repräsentierten die Korrelationen nach Spearman's ϱ für jedes mögliche Paar aus den 94 Studienteilnehmern. Insgesamt enthält $\mathrm{Corr}(S)$ also $94 \cdot (94-1)/2 = 4371$ Korrelationskoeffizienten. Die restlichen kombinatorischen Möglichkeiten müssen nicht betrachtet werden, da die Korrelation kommutativ ist, d. h. dass $\varrho(S_i, S_j) = \varrho(S_j, S_i)\, \forall i, j \in \{1, 2, \ldots, 94\}$ gilt. Die Korrelation eines Teilnehmers mit sich selbst ist zudem auch uninteressant, da diese immer maximal positiv ist, also $\varrho(S_i, S_i) = 1$ gilt. Für die 4371 Einträge von $\mathrm{Corr}(S)$ nimmt Spearman's ϱ jeweils einen Wert im Intervall $[-1, 1]$ an [56]. Ein positives ϱ gibt einen positiven Zusammenhang zwischen der Wahrnehmung der Teilnehmer an (z. B. übereinstimmende *positive* und *negative* Annotationen für dieselben Aussagen). Ein negatives ϱ gibt hingegen einen negativen Zusammenhang der Wahrnehmung zweier Studienteilnehmer an (z. B. je eine *positive* und eine *negative* Annotation für mehrere Aussagen). Der Betrag $|\varrho|$ gibt dabei an, wie stark die Wahrnehmung der Studienteilnehmer zusammenhängt, unabhängig davon, ob dieser Zusammenhang positiv oder negativ ist [56]. Um die Korrelationsmatrix $\mathrm{Corr}(S)$ leichter interpretieren zu können, wurden die 4371 Werte von Spearman's ϱ mit einem divergierenden Farbverlauf [100] visualisiert, um positiv, negativ, und gar nicht miteinander korrelierende Studienteilnehmer optisch voneinander abzugrenzen zu können.

Hypothesentests für Zusammenhänge der Wahrnehmung

Um die statistische Signifikanz der Zusammenhänge in der Wahrnehmung der Stimmung zwischen den Studienteilnehmern zu prüfen, wurden mehrere Nullhypothesen aufgestellt, die in Tabelle 4.2 definiert sind.

Tabelle 4.2: Definition der 94 übergeordneten Nullhypothesen $\mathrm{H1}(S_i)_0$ und 4371 untergeordneten Nullhypothesen $\mathrm{H1}(S_i, S_j)_0$.

	Definition der Nullhypothesen
$\mathrm{H1}(S_i)_0$	Es gibt keinen Zusammenhang zwischen den Annotationen von Studienteilnehmer S_i und den Annotationen der anderen Studienteilnehmer $\{S_1, S_2, \ldots, S_{94}\} \setminus \{S_i\}$.
$\mathrm{H1}(S_i, S_j)_0$	Es gibt keinen Zusammenhang zwischen den Annotationen der Studienteilnehmer S_i und S_j, wobei $i, j \in \{1, 2, \ldots, 94\}$ und $i < j$ gilt.

Die 94 übergeordneten Nullhypothesen $\mathrm{H1}(S_i)_0$ können dabei anhand der 4371 untergeordneten Nullhypothesen $\mathrm{H1}(S_i, S_j)_0$ überprüft werden. Sobald $\mathrm{H1}(S_i, S_j)_0$ für die Studienteilnehmer S_i und S_j abgelehnt werden kann, werden auch $\mathrm{H1}(S_i)_0$ und $\mathrm{H1}(S_j)_0$ abgelehnt, da beide Studienteilnehmer mindestens einen signifikanten Zusammenhang zu den Annotationen des

jeweils anderen aufweisen. Dabei können die 4371 untergeordneten Nullhypothesen $\mathrm{H1}(S_i, S_j)_0$ anhand der Korrelationen $\varrho(S_j, S_i)$ der Studienteilnehmer überprüft werden. Dafür wird zu jedem ϱ-Wert aus der Korrelationsmatrix $\mathrm{Corr}(S)$ die zugehörige Wahrscheinlichkeit der Korrelation $P(>|\varrho(S_j, S_i)|)$ berechnet. Da jeder Studienteilnehmer S_i für die Beantwortung von $\mathrm{H1}(S_i)_0$ mit jedem anderen Teilnehmer $\{S_1, S_2, \ldots, S_{94}\}\backslash\{S_i\}$ verglichen wird, wurde das Signifikanzniveau α durch die Bonferroni-Korrektur [127] (von 0.05 nach Fisher [33]) auf $\alpha = 0.05/93 \approx 0.000538$ abgesenkt. Um $\mathrm{H1}(S_i, S_j)_0$ abzulehnen, muss also $P(>|\varrho(S_j, S_i)|) < \alpha$ gelten. Folglich kann $\mathrm{H1}(S_i)_0$ für jeden Studienteilnehmer S_i abgelehnt werden, der Teil mindestens einer signifikanten Korrelation $P(>|\varrho(S_j, S_i)|) < \alpha$ ist.

4.3.2 Clusteranalyse der Studienteilnehmer

Um die Strukturen innerhalb der Datenbasis zu erforschen, wird eine Clusteranalyse auf Basis der Annotationen der Studienteilnehmer durchgeführt. Somit sollen Studienteilnehmer, die eine ähnliche Wahrnehmung der Stimmung aufweisen, gruppiert werden. Als Clusterverfahren wird dafür die hierarchische Clusteranalyse [106] verwendet (vgl. Kapitel 2.3). Das Ergebnis der hierarchischen Clusteranalyse ist eine verschachtelte Sequenz von Partitionen der Studienteilnehmer, von einem Cluster hin zu 94 einzelnen Clustern [146]. Die Anzahl der Cluster muss somit nicht im Vorfeld bestimmt werden, wie es bei partitionierenden Clusterverfahren (z. B. dem k-Means-Algorithmus) der Fall ist, sondern kann stattdessen im Nachhinein festgelegt werden [146]. Für die hierarchische Clusteranalyse wird nachfolgend das Distanzmaß definiert und der gewählte Fusionierungsalgorithmus beschrieben.

Definition des Distanzmaßes

Da die zuvor in Kapitel 4.3.1 berechneten Korrelationen angeben, wie sehr sich die Wahrnehmung zweier Studienteilnehmer ähnelt, bietet es sich an, aus der Korrelation eine Unähnlichkeit bzw. Distanz für die hierarchische Clusteranalyse abzuleiten. In der Literatur [38, 41, 99] wird die Korrelation Corr unter Verwendung des Kosinussatzes [55] zu einer geometrisch korrekten euklidischen Distanz d durch $d = \sqrt{2 \cdot (1 - \mathrm{Corr})}$ umgeformt. Für ein konkretes Paar von Studienteilnehmern S_i und S_j aus der Datenbasis, für welche der Zusammenhang ihrer Annotationen durch $\varrho(S_i, S_j)$ gemessen wird, ergibt sich die Distanz der beiden Teilnehmer d_{ij} also wie folgt.

$$d_{ij} = \sqrt{2 \cdot (1 - \varrho(S_i, S_j))} \tag{4.5}$$

Die Distanzen d_{ij} aller Studienteilnehmer S_i und S_j können folglich direkt aus den Einträgen der Korrelationsmatrix $\mathrm{Corr}(S)$ berechnet werden. Abbildung 4.2 zeigt, den Verlauf der Distanz d_{ij} in Abhängigkeit von der Korrelation $\varrho(S_i, S_j)$ der Annotationen zweier Studienteilnehmer S_i und S_j.

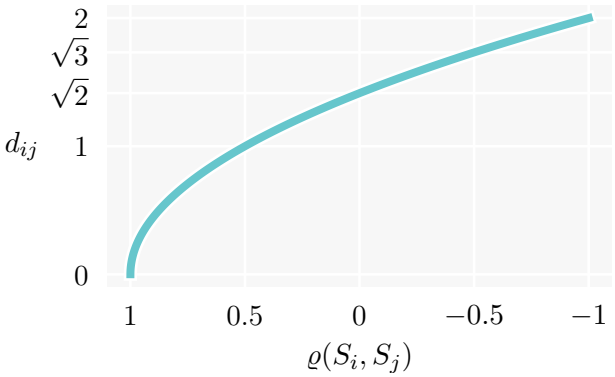

Abbildung 4.2: Die Distanz zweier Studienteilnehmer S_i und S_j ist durch d_{ij} in Abhängigkeit von der Korrelation zwischen den Annotationen $\varrho(S_i, S_j)$ der Teilnehmer gegeben.

Wie Abbildung 4.2 zu entnehmen ist, nimmt die Distanz d_{ij} einen Wert im Intervall $[0, 2]$ an. Für $\varrho(S_i, S_j) = 1$, also eine maximal positive Korrelation der Annotationen zweier Studienteilnehmer S_i und S_j beträgt die Distanz $d_{ij} = 0$. Für $\varrho(S_i, S_j) = -1$, also eine maximal negative Korrelation der Annotationen, zweier Studienteilnehmer, beträgt die Distanz $d_{ij} = 2$. Für alle anderen Werte von $\varrho(S_i, S_j)$ im Intervall $[-1, 1]$ besitzt die Distanz d_{ij} einen nicht linearen Zusammenhang zur Korrelation $\varrho(S_i, S_j)$. So nehmen zwei Studienteilnehmer, die nicht miteinander korrelieren, für $\varrho(S_i, S_j) = 0$, eine Distanz von $d_{ij} = \sqrt{2}$ an (vgl. Gleichung 4.5).

Wahl des Fusionierungsalgorithmus

Neben dem Distanzmaß muss für das hierarchische Clustering auch ein Fusionierungsalgorithmus (engl. *Linkage*) gewählt werden, der festlegt, welche Cluster in jedem hierarchischen Schritt miteinander fusioniert werden [102]. Als Fusionierungsalgorithmus für das hierarchische Clustering in dieser Arbeit wurde Ward's minimales Varianzkriterium [162] ausgewählt (vgl. Kapitel 2.3). Ward's minimales Varianzkriterium erzielt im Vergleich zu anderen Fusionierungsalgorithmen besonders gute Ergebnisse dabei, Cluster zu identifizieren, welche nicht stark voneinander separiert sind [158]. Dies wird im Rahmen des Ziels dieser Arbeit präferiert, da die Annotationen der Studienteilnehmer nur drei diskrete Werte (*negativ*, *neutral* und *positiv*) annehmen konnten und diese somit nicht stark variieren können. Durch das hierarchische Clustering werden anfänglich 94 Cluster initialisiert, wobei jeder Studienteilnehmer sich in seinem eigenen Cluster mit minimaler Distanz befindet. Dies entspricht einem agglomerativem hierarchischen Clustering, d. h. einem Bottom-Up-Verfahren, wobei schrittweise Cluster miteinander fusioniert werden, bis sich alle Teilnehmer im selben Cluster befinden (vgl. im Gegensatz dazu Kapitel 2.3 für divisives hierarchisches Clustering). Durch den Fusionierungsalgorithmus von Ward werden anschließend in jedem Schritt die zwei Cluster A und B fusioniert, deren Vereinigungsmenge $A \cup B$ von Studienteilnehmern die geringste Varianz der Annotationen aufweist [162]. Ward's minimales Varianzkriterium neigt somit dazu, Cluster mit einer hohen Dichte zu formen (vgl. Kapitel 2.3).

Visualisierung der Clustering-Ergebnisse

Um die Ergebnisse des hierarchischen Clusterings zu visualisieren, stehen eine Vielzahl von Verfahren zur Verfügung [57]. Zwei weitverbreitete Methoden sind die Darstellung der Ergebnisse durch ein Dendrogramm oder eine geordnete Korrelationsmatrix, die im Folgenden beschrieben werden.

Dendrogramm: Die Ergebnisse des hierarchischen Clusterings können durch ein Dendrogramm [57] visualisiert werden. Das Dendrogramm ist eine baumähnliche Struktur, dessen horizontale Verbindungen die iterativen Fusionen des hierarchischen Clusterings repräsentieren [57]. Die Blätter des Dendrogramms repräsentieren die 94 Studienteilnehmer und sind auf der x-Achse aufgetragen. Auf der y-Achse ist die durch den Fusionierungsalgorithmus bestimmte Distanz von Clustern aufgetragen [57]. Bei Ward's minimalem Varianzkriterium entspricht diese der Varianz innerhalb der fusionierten Cluster (vgl. Kapitel 2.3). Horizontale Verbindungen im Dendrogramm visualisieren die Fusion zweier Cluster, wobei diese auf Höhe der Distanz beider Cluster vor der Fusion eingetragen werden [57]. Da die Distanz der in jedem Schritt fusionierten Clustern wächst, sind die horizontalen Verbindungen im Dendrogramm also übereinander von der günstigsten Fusion (unten) bis zur teuersten Fusion (oben) angeordnet [57]. Das Dendrogramm kann als visuelle Entscheidungshilfe für das Abschätzen der Anzahl an tatsächlichen Clustern in den Daten genutzt werden [57].

Geordnete Korrelationsmatrix: Eine weitere Möglichkeit, die Ergebnisse des Clusterings zu visualisieren ist, die vollständige Korrelationsmatrix der Studienteilnehmer so darzustellen, dass die Reihenfolge der Zeilen und Spalten entsprechend der Clustering-Ergebnisse permutiert wird [57]. Die vollständige Korrelationsmatrix der Studienteilnehmer besteht aus der unteren Dreiecksmatrix $\mathrm{Corr}(S)$ aus Gleichung 4.4, der oberen Dreiecksmatrix $\mathrm{Corr}(S)^\top$, sowie einer Hauptdiagonale aus Einsen. Die Zeilen und Spalten der vollständigen Korrelationsmatrix werden bei diesem Verfahren so permutiert, dass Studienteilnehmer mit geringer Distanz in der Reihenfolge der Zeilen und Spalten benachbart sind [57]. Diese Permutation ist identisch zu der Reihenfolge der Studienteilnehmer in den Blättern des Dendrogramms. Die Studienteilnehmer, welche sich nach den Ergebnissen des hierarchischen Clusterings in denselben Clustern befinden, sind folglich auch in den Zeilen und Spalten der geordneten Korrelationsmatrix benachbart angeordnet [57]. Dadurch, dass die Korrelation der Studienteilnehmer im selben Cluster höher ist als zwischen den verschiedenen Clustern, entstehen in der geordneten Korrelationsmatrix sichtbare Blöcke als korrelative Strukturen, bei einer entsprechenden Visualisierung der Korrelation durch einen divergierenden Farbverlauf [100].

Das Python-Paket *Seaborn* bietet unter anderem eine Funktion namens *Clustermap* an, welche die Ergebnisse eines hierarchischen Clusterings mittels der geordneten Korrelationsmatrix und einem Dendrogramm visualisiert [163]. Dazu wird die Eigenschaft, dass die Reihenfolge der Zeilen und Spalten der geordneten Korrelationsmatrix der Reihenfolge der Blätter des Dendrogramms entspricht, ausgenutzt, indem diese miteinander verbunden werden [57]. Die Ergebnisse der hierarchischen Clusteranalyse dieser Arbeit werden mit dieser Funktion visualisiert.

Bestimmung der optimalen Anzahl an Teilnehmerclustern

Anhand rein visueller Verfahren kann es schwierig sein, die Anzahl der vorhandenen Cluster abzuschätzen. Es empfiehlt sich daher, zusätzlich eine Metrik wie den Silhouettenkoeffizienten [124] auf die Ergebnisse anzuwenden. Der Silhouettenkoeffizient gibt die Güte eines Clusterings in Bezug auf die Dichte in jedem Cluster und die Distanz der verschiedenen Cluster untereinander durch einen Wert im Intervall $[-1, 1]$ an. Um die optimale Anzahl der Cluster aus den Ergebnissen der hierarchischen Clusteranalyse abzuleiten, wird der Silhouettenkoeffizient für jede Partitionierung der Studienteilnehmer, beginnend bei zwei Clustern, hin zu 94 Clustern berechnet. Die Anzahl an Clustern, welche den höchsten zugehörigen Silhouettenkoeffizient erzielt, ist folglich die optimale Anzahl an Teilnehmerclustern [124]. Die Interpretation des Silhouettenkoeffizienten wird anhand der Literatur von Kaufman und Rousseeuw [71] durchgeführt (vgl. Tabelle 4.3).

Tabelle 4.3: Interpretation des Silhouettenkoeffizienten SK, nach der Literatur von Kaufman und Rousseeuw [71].

SK	Interpretation
$0.70 < SK \leq 1.00$	Eine starke Struktur wurde gefunden.
$0.50 < SK \leq 0.70$	Eine sinnvolle Struktur wurde gefunden.
$0.25 < SK \leq 0.50$	Eine schwache, möglicherweise künstliche Struktur wurde gefunden.
$SK \leq 0.25$	Keine substanzielle Struktur wurde gefunden.

Aufgrund des „*Fluches der Dimensionalität*“ (engl. *Curse of Dimensionality*) [4] wird auch der Silhouettenkoeffizient abgeschwächt, da jeder der 94 Teilnehmer durch seine insgesamt 96 Annotationen repräsentiert wird. Eine mögliche Gegenmaßnahme ist, den Silhouettenkoeffizient im Anschluss auf die im folgenden Kapitel 4.3.3 erläuterten Dimensionsreduktionsverfahren [159] erneut zu berechnen. Somit kann auch die Existenz möglicherweise visuell erkennbarer Cluster in der dimensionsreduzierten Datenbasis validiert werden.

Bestimmung der internen Konsistenz der Teilnehmergruppen

Um die interne Konsistenz der Teilnehmergruppen, welche durch die Ergebnisse der hierarchischen Clusteranalyse definiert werden, zu validieren, kann für die Annotationen der Studienteilnehmer einer jeden Gruppe, Cronbach's α [16] berechnet werden. Nach der Methodik von Klünder et al. [75] gibt Cronbach's α somit an, ob durch die Annotationen der Aussagen von Studienteilnehmern innerhalb einer Gruppe dieselbe latente Variable [28] gemessen wird. Die latente Variable kann dabei im Rahmen dieser Arbeit als Wahrnehmung bezeichnet werden, wobei jedoch nicht näher bekannt ist, woraus sich diese zusammensetzt. Um die resultierende Werte von Cronbach's α für die einzelnen Gruppen zu interpretieren, wird die Skala von Hair et al. [48] genutzt, welche in Tabelle 4.4 dargestellt ist.

Tabelle 4.4: Interpretation von Cronbach's α als Maß der internen Konsistenz, nach der Literatur von Hair et al. [48].

Cronbach's α	**Interne Konsistenz**
$0.90 \leq \alpha$	Exzellent
$0.80 \leq \alpha < 0.90$	Gut
$0.70 \leq \alpha < 0.80$	Akzeptabel
$0.60 \leq \alpha < 0.70$	Fragwürdig
$0.50 \leq \alpha < 0.60$	Schlecht
$\alpha < 0.50$	Inakzeptabel

4.3.3 Dimensionsreduktion der Datenbasis

Hochdimensionale Daten werden in der Fachliteratur als Daten definiert, bei denen die Anzahl der Variablen nahe oder sogar größer der Anzahl an Datenpunkten ist [135]. Da die Annotationen der 94 Studienteilnehmer der Datenbasis in einem 96-dimensionalen Raum liegen können diese folglich als hochdimensionale Daten klassifiziert werden (vgl. Kapitel 2.4.1). Da eine Visualisierung der Ergebnisse der Clusteranalyse im hochdimensionalen Raum nicht möglich ist, soll die hochdimensionale Datenbasis stattdessen mithilfe von Dimensionsreduktionsverfahren in einen niedrigerdimensionalen Raum transformiert werden [35]. Neben der Visualisierung der Clustering-Ergebnisse können somit auch weitere Erkenntnisse über die verschiedenen Wahrnehmungen der Studienteilnehmer gewonnen werden. In Kapitel 2.4 wurden zwei Dimensionsreduktionsverfahren, die Hauptkomponentenanalyse [66] und die lineare Diskriminanzanalyse [32] vorgestellt. Im Folgenden wird beschrieben, wie die Hauptkomponentenanalyse und die lineare Diskriminanzanalyse auf die Datenbasis angewendet werden.

Anwendung der Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (engl. *Principal Component Analysis*) ist ein statistisches Verfahren zur Dimensionsreduktion, welches beabsichtigt, die maximale Varianz der ursprünglich hochdimensionalen Daten zu erhalten [35]. Um die Hauptkomponentenanalyse auf die Datenbasis dieser Arbeit anzuwenden, muss zunächst die 96×96 Kovarianzmatrix der Annotationen, die von jedem Studienteilnehmer getätigt wurden, gebildet werden [66]. Um die Kovarianzen zu berechnen wurden die Ränge der ordinalen Kategorien *negativ*, *neutral*, und *positiv* verwendet.

> Für die Anwendung der Hauptkomponentenanalyse werden mindestens intervallskalierte Daten benötigt. Obwohl die Hauptkomponentenanalyse damit also methodisch streng genommen nicht auf ordinalskalierte Daten anwendbar ist, sind auf diese Weise erzielte Ergebnisse oftmals dennoch aussagekräftig (vgl. Robitzsch [122] und Sullivan et al. [147]).

Für die Anwendung der Hauptkomponentenanalyse wird also angenommen, dass die Sentiment-Polaritäten *negativ* und *positiv* denselben Abstand zu der Sentiment-Polarität *neutral* besitzen, obwohl dies nicht gesichert ist. Die Hauptkomponentenanalyse wird im Rahmen dieser Arbeit jedoch nur genutzt, um die Ergebnisse des hierarchischen Clusterings für den Leser verständlich zu visualisieren, was ohne eine Dimensionsreduktion nicht möglich gewesen wäre. Aus der 96×96 Kovarianzmatrix der Annotationen werden anschließend die Eigenwerte und deren zugehörige Eigenvektoren berechnet (vgl. Kapitel 2.4.2). Die ersten beiden Hauptkomponenten der Datenbasis ergeben sich aus den Eigenvektoren mit dem größten und zweitgrößten Eigenwert [66]. Um die hochdimensionalen Annotationen der Studienteilnehmer in den zweidimensionalen Raum zu transformieren, werden die x- und y-Koordinaten eines jeden Studienteilnehmers durch Skalarmultiplikation der ersten und zweiten Hauptkomponente mit den Rängen der Annotationen des Studienteilnehmers berechnet [66]. Die berechneten x- und y-Koordinaten eines jeden Studienteilnehmers können abschließend durch Punkte in einem Streudiagramm visualisiert werden, welches den größtmöglichen Anteil der Varianz aus der hochdimensionalen Datenbasis in nur zwei Dimensionen abbildet [66]. Um die vorangegangenen Ergebnisse des hierarchischen Clusterings im neu berechneten zweidimensionalen Raum zu betrachten, werden die einzelnen Streupunkte der Studienteilnehmer entsprechend der ihnen zugeordneten Gruppen koloriert. Entscheidend ist dabei, dass die Hauptkomponentenanalyse ohne das Vorwissen darüber, welcher Teilnehmer welcher Gruppe zugeordnet wurde, vorgenommen wird, da es sich bei der Hauptkomponentenanalyse um ein unüberwachtes Dimensionsreduktionsverfahren handelt [35]. Somit kann ein Eindruck darüber gewonnen werden, wie gut die Teilnehmergruppen durch die hierarchische Clusteranalyse abgegrenzt worden sind (z. B. ob diese sich überlappen).

Anwendung der linearen Diskriminanzanalyse

Die lineare Diskriminanzanalyse (engl. *Linear Discriminant Analysis*) nach Fisher [32] ist ein weiteres Dimensionsreduktionsverfahren. Im Gegensatz zur Hauptkomponentenanalyse handelt es sich bei der linearen Diskriminanzanalyse jedoch um ein überwachtes Dimensionsreduktionsverfahren (vgl. Kapitel 2.4.3). Im Rahmen dieser Arbeit soll durch die lineare Diskriminanzanalyse ein niedrigerdimensionaler Raum gefunden werden, in welchem die Studienteilnehmer nach einer Transformation aus der hochdimensionalen Datenbasis entsprechend ihrer zugeordneten Teilnehmergruppen bestmöglich separiert werden. Sind die Gruppen im resultierenden dimensionsreduzierten Raum deutlich sichtbar voneinander separiert, so ist diese Separation folglich allein anhand der 96 Annotationen eines jeden Teilnehmers möglich [35]. Die Lineare Diskriminanzanalyse setzt jedoch voraus, dass die Eingabedaten einer multivariaten Normalverteilung [69] folgen [35, 123]. Dass die Ränge der Annotationen der Studienteilnehmer in der Datenbasis dieser Arbeit keiner multivariaten Normalverteilung folgen, wird durch die Ergebnisse des Henze-Zirkler-Testes [52], welche in Tabelle 4.5 dargestellt sind, bestätigt.

Tabelle 4.5: Ergebnisse des Henze-Zirkler-Testes [52] für die Annotationen der Studienteilnehmer in der Datenbasis.

HZ	***p***	**Interpretation**
376	< 0.0001	Keine multivariate Normalverteilung

Die Nullhypothese des Henze-Zirkler-Testes, dass die Datenbasis einer multivariaten Normalverteilung folgt, muss demnach abgelehnt werden. Weiterhin setzt die Anwendung der linearen Diskriminanzanalyse die Homoskedastizität [129] (eine konstante Varianz aller Variablen) der Eingabedaten voraus [123]. Da die Annotationen der Studienteilnehmer in der Datenbasis dieser Arbeit keiner multivariaten Normalverteilung folgen, wurde zu Überprüfung der Homoskedastizität der Levene-Test [27] anstelle des Bartlett-Testes [1] ausgeführt. Die Ergebnisse des Levene-Testes in der Ausführung von Brown und Forsythe [8] sind in Tabelle 4.6 dargestellt.

Tabelle 4.6: Ergebnisse des Levene-Testes [27] in der Ausführung von Brown und Forsythe [8] für die Annotationen der Studienteilnehmer.

L	***p***	**Interpretation**
6.6985	< 0.0001	Keine Homoskedastizität

Wie Tabelle 4.6 zu entnehmen ist, muss auch die Nullhypothese des Levene-Testes, dass alle Aussagen der Datenbasis in Bezug auf die Ränge ihrer Annotationen eine konstante Varianz aufweisen, abgelehnt werden.

> Die Datenbasis dieser Arbeit kann beide Vorbedingungen der linearen Diskriminanzanalyse nicht erfüllen. Viele Publikationen weisen jedoch darauf hin, dass die lineare Diskriminanzanalyse robust gegen die Verletzung ihrer Vorbedingungen ist (vgl. Duda et al. [26], Hastie et al. [50], Huberty [60], Klecka [73] und Li et al. [83]).

Da die Vorbedingungen der linearen Diskriminanzanalyse nicht erfüllt sind, wird die logistische Regressionsanalyse [130], welche in diesem Fall als gängige Alternative gilt [14] zusätzlich auf die Datenbasis dieser Arbeit angewendet. Die Ergebnisse der linearen Diskriminanzanalyse werden nur für die Dimensionsreduktion der Datenbasis und für die Identifikation polarisierender Aussagen in Bezug auf die unterschiedliche Wahrnehmung der verschiedenen Teilnehmergruppen verwendet. Letzteres ist durch die Berechnung der linearen Diskriminanten möglich, die in Kapitel 2.4.3 beschrieben wurde. Die linearen Diskriminanten gewichten die ordinalen Ränge der Sentiment-Polaritäten *negativ*, *neutral* und *positiv*, für alle 96 Aussagen $V_i \in V$ unterschiedlich stark. Infolgedessen haben stärker gewichtete Aussagen einen größeren Einfluss auf die berechneten Koordinaten bzw. die Position eines jeden Teilnehmers im transformierten niedrigerdimensionalen Raum, welcher die Separation der Teilnehmergruppen maximiert [35]. Daraus kann geschlossen werden, dass hoch gewichtete Aussagen besonders unterschiedliche Wahrnehmungen der Studienteilnehmer in verschiedenen Gruppen hervorrufen. Nachdem die geeigneten Gewichte der Annotation durch die lineare Diskriminanzanalyse ermittelt wurde, kann diese zusätzlich zur Anwendung als Dimensionsreduktionsverfahren auch als Klassifizierungsverfahren angewendet werden [118]. Dabei erreicht die lineare Diskriminanzanalyse oftmals sogar genauere Ergebnisse der Klassifizierung als vergleichbare Verfahren, sofern die Vorbedingungen der linearen Diskriminanzanalyse erfüllt sind [118]. Da dies für die Datenbasis dieser Arbeit aber nicht der Fall ist, ist jedoch anzunehmen, dass der Klassifizierungsfehler nicht minimal ist [80]. Daher bietet es sich an, stattdessen eine logistische Regressionsanalyse durchzuführen, mit welcher sich dieselben Forschungsfragen, ohne die Vorbedingungen der multivariaten Normalität und Homoskedastizität, beantworten lassen [14].

4.3.4 Logistische Regressionsanalyse

Die logistische Regressionsanalyse ist ein weitverbreitetes Klassifizierungsverfahren, welches in der Lage ist, anhand einer Menge von Prädiktorvariablen (engl. *Explanatory Variables*) die Ausprägung einer dichotomen abhängigen Variable vorherzusagen [23]. Im Rahmen dieser Arbeit soll die

logistische Regressionsanalyse genutzt werden, um festzustellen, ob eine Teilmenge $X \subset V$ der Aussagen in der Datenbasis ausreichend ist, um die Studienteilnehmer korrekt ihren Teilnehmergruppen zuzuordnen. Somit kann festgestellt werden, wie viele Annotationen eines Studienteilnehmers nötig sind, um dessen Wahrnehmung der Stimmung zu erfassen.

Merkmalsselektion durch die lineare Diskriminanzanalyse

Neben der Anwendung als Klassifizierung- und Dimensionsreduktionsverfahren kann die lineare Diskriminanzanalyse auch zur Merkmalsselektion verwendet werden [144]. Im Rahmen dieser Arbeit kann sich diese Eigenschaft zunutze gemacht werden, um die Anzahl der wichtigsten Aussagen in der Datenbasis iterativ zu reduzieren. Dafür wird zunächst die initiale Berechnung der linearen Diskriminanzanalyse genutzt, um die Aussage zu identifizieren, die den niedrigsten absoluten Koeffizienten $|K|$ besitzt. Da diese Aussage den niedrigsten Einfluss auf die Separation der Teilnehmergruppen nimmt, kann die lineare Diskriminanzanalyse erneut ohne diese Aussage, d. h. anhand der verbleibenden 95 Aussagen berechnet werden. Dieses Verfahren wird dann iterativ wiederholt, bis nur noch eine einzelne Aussage übrig ist. Durch diese Methodik ergibt sich für jedes n, $1 \leq n \leq 95$ eine Teilmenge $X \subset V, |X| = n$ der Aussagen $V = \{V_1, V_2, \ldots, V_{96}\}$, welche durch die lineare Diskriminanzanalyse als die n wichtigsten Aussagen identifiziert wurden. Somit kann für jedes n, beginnend bei $n = 1$ ein logistisches Regressionsmodell berechnet werden, für welches anschließend geprüft werden kann, ob alle Studienteilnehmer ihrer korrekten Teilnehmergruppen zugeordnet werden können. Gelingt dies nicht, so kann n inkrementiert werden, wodurch die nächst wichtigste Aussage der Datenbasis mit für die Vorhersage der Teilnehmergruppen berücksichtigt wird. Dies kann so lange wiederholt werden, bis die Gruppen aller 94 Studienteilnehmer korrekt durch das logistische Regressionsmodell vorhergesagt werden können. Somit ergibt sich die Mindestanzahl an Aussagen der Datenbasis, welche nötig sind, um die verschiedenen Teilnehmergruppen voneinander zu separieren.

Logistische Regression nach Firth

Um die Teilnehmergruppe eines Studienteilnehmers anhand dessen Annotationen einer Teilmenge der Aussagen der Datenbasis zu berechnen, wird die logistische Regression nach Firth [31] verwendet, welche von Heinze und Schemper [51] erstmals detailliert beschrieben wurde. Diese Variante der logistischen Regressionsanalyse ist besonders für Datensätze mit wenigen Datenpunkten geeignet [51]. Zudem wird durch diese Variante das Problem der vollständigen Separation [3], welches häufig bei logistischen Regressionsanalysen kleinerer Datensätze mit kollinearen [25] Prädiktorvariablen zu Konvergenzproblemen [3] führt, eliminiert [51]. Da in der Datenbasis dieser Arbeit nur 94 Datenpunkte vorliegen und von einer Multikollinearität [25] der 96 Aussagen ausgegangen werden kann, ist

die logistische Regression nach Firth aufgrund ihrer Robustheit gegenüber den zuvor beschriebenen Problemen der klassischen logistischen Regression vorzuziehen [51]. Anhand des somit berechneten Regressionsmodells können abschließend die Einflüsse der Aussagen des Regressionsmodells auf die Vorhersage der Teilnehmergruppen abgeschätzt werden. Diese Einflüsse können durch den Wald-Test [161] zudem auf statistische Signifikanz geprüft werden. Auf diese Weise kann ein signifikanter Zusammenhang zwischen den Aussagen des Regressionsmodells und der Vorhersage der Teilnehmergruppen nachgewiesen werden.

4.3.5 Statistischer Vergleich der Teilnehmergruppen

Um die Datenanalyse dieser Arbeit abzuschließen, werden die verschiedenen Gruppen von Studienteilnehmern, welche laut den Ergebnissen der hierarchischen Clusteranalyse eine unterschiedliche Wahrnehmung der Stimmung aufweisen, auf Unterschiede in den restlichen Merkmalen der Datenbasis untersucht. Alle durch das Erhebungsdesign der ursprünglichen Umfrage [111] erfassten Merkmale wurden in Kapitel 4.1.1 beschrieben, und sind in Tabelle A.1 im Anhang dieser Arbeit aufgelistet. Um festzustellen, ob die Unterschiede zwischen den Merkmalswerten der Studienteilnehmer verschiedener Teilnehmergruppen statistisch signifikant sind, müssen je nach Merkmal unterschiedliche statistische Testverfahren angewendet werden. Im Folgenden wird beschrieben, wie die Freitext-Antworten, sowie die nominalen, ordinalen und metrischen Merkmale der Datenbasis auf Unterschiede zwischen den Teilnehmergruppen untersucht werden.

Vergleich der Freitext-Antworten

Die Freitext-Antworten, zu der Frage, nach welchem Kriterium die Teilnehmer den Aussagen eine Sentiment-Polarität zugeordnet haben (vgl. Tabelle A.1e), müssen manuell analysiert werden. Dafür werden die Antworten der Studienteilnehmer nach den jeweiligen Teilnehmergruppen aufgeteilt. Zunächst können die Antworten der Studienteilnehmer so auf Gemeinsamkeiten innerhalb der Gruppen untersucht werden. Abschließend können die Antworten zwischen den verschiedenen Teilnehmergruppen miteinander verglichen werden.

Vergleich der nominalen Merkmale mit Einfachauswahl

Für die Untersuchung aller Merkmale, bei deren Antwort die Studienteilnehmer zwischen zwei oder mehreren nominalen Kategorien eine Einfachauswahl treffen konnten, wird der χ^2-Test [116] angewendet. Dies betrifft das Geschlecht sowie die Angabe, ob Englisch oder eine andere Sprache die Erstsprache des jeweiligen Studienteilnehmers ist (vgl. Tabelle A.1a). Dafür wird zunächst die Kontingenztabelle [24] zwischen den Teilnehmergruppen und den kategorischen Antwortmöglichkeiten der jeweiligen Frage gebildet. Anhand der Kontingenztabelle wird anschließend die χ^2-Statistik, sowie die zugehörige Wahrscheinlichkeit $P(>\chi^2)$ berechnet [116].

Vergleich der nominalen Merkmale mit Mehrfachauswahl

Für einige Fragen mit kategorischen Antwortmöglichkeiten konnte mehr als eine zutreffende Antwortmöglichkeit von den Studienteilnehmern ausgewählt werden [111]. Dazu gehören die Auswahl des beruflichen Status (vgl. Tabelle A.1b) und der vordefinierten Annotationskriterien (vgl. Tabelle A.1e) durch die Studienteilnehmer. Für diese Merkmale wird die χ^2-Statistik verfälscht, da der χ^2-Test eine Einfachauswahl der Antwortmöglichkeiten annimmt, d. h. dass die Summe der Einträge in der Kontingenztabelle m der Gesamtzahl an Studienteilnehmern n entspricht, also $m = n$ gilt [116]. In diesen Fällen muss daher nach der Durchführung des χ^2-Testes die Rao-Scott-Korrektur [120, 121] auf die resultierende χ^2-Statistik angewendet werden. Sei $m > n$ und sei k die Anzahl der mehrfach auswählbaren Kategorien eines Merkmals. Dann wird aus der χ^2-Statistik die angepasste χ^2_C-Statistik nach der Methodik von Decady und Thomas [19] folgendermaßen berechnet.

$$\chi^2_C = \frac{\chi^2}{\tilde{\delta}}\,,\ \tilde{\delta} = 1 - \frac{m}{n \cdot k} \tag{4.6}$$

Die Anzahl der Freiheitsgrade (engl. *Degrees of Freedom*) [59], ändert sich ebenfalls, wenn $m > n$ für die Kontingenztabelle eines Merkmals gilt. Die korrigierte Anzahl der Freiheitsgrade ergibt sich für die betreffenden Merkmale durch $df_C = (g-1) \cdot k$, wobei g der Anzahl an Teilnehmergruppen entspricht, die durch die hierarchische Clusteranalyse ermittelt wurden. Anhand von χ^2_C und df_C kann abschließend wiederum eine Wahrscheinlichkeit $P(>\chi^2_C)$ für das angepasste Ergebnis von χ^2_C berechnet werden.

Vergleich der ordinalen Merkmale

Merkmalswerte, die von den Studienteilnehmern auf einer Likert-Skala angegeben wurden, gelten als ordinalskalierte Daten. In der Datenbasis dieser Arbeit wurden die Merkmale der Programmiererfahrung und Familiarität mit der Arbeit in Entwicklungsteams (vgl. Tabelle A.1c), die englischsprachige Kommunikationshäufigkeit (vgl. Tabelle A.1a) und die Annotationen der 96 Aussagen selbst (vgl. Tabelle A.1d) jeweils durch eine Likert-Skala erfasst [111]. Um zu überprüfen, ob es statistisch signifikante Unterschiede dieser Merkmale zwischen den Teilnehmergruppen gibt, eignet sich der Mann-Whitney-U-Test [92]. Dieser überprüft für zwei Teilnehmergruppen mit mindestens ordinalskalierten Merkmalswerten die Nullhypothese, dass es gleich wahrscheinlich ist, dass ein zufällig ausgewählter Wert der ersten Teilnehmergruppe größer oder kleiner ist als ein zufällig ausgewählter Wert aus der zweiten Teilnehmergruppe [92]. Fällt der Mann-Whitney-U-Test signifikant aus, so ist es bei der zufälligen Auswahl eines Merkmalswertes aus zwei Teilnehmergruppen also immer wahrscheinlicher, einen größeren Wert für einer der beiden Teilnehmergruppen zu erhalten.

Vergleich der metrischen Merkmale

Zu den metrischen Merkmalen der Datenbasis gehören das Alter (vgl. Tabelle A.1a), sowie die Jahre an Berufserfahrung und die Jahre an Berufserfahrung in Entwicklungsteams der Studienteilnehmer (vgl. Tabelle A.1c). Um die Merkmalswerte dieser Merkmale auf Unterschiede zwischen den Teilnehmergruppen zu überprüfen, muss zunächst überprüft werden, welche Vorbedingungen von den Merkmalswerten der Teilnehmergruppen erfüllt werden. Die Entscheidungstabelle 4.7 gibt eine Übersicht über die Auswahl der statistischen Testverfahren für die metrischen Merkmale in Abhängigkeit von den Vorbedingungen der Normalität und Homoskedastizität.

Tabelle 4.7: Statistische Testverfahren für metrische Merkmale in Abhängigkeit von den erfüllten Vorbedingungen (✓) des jeweiligen Merkmals.

Vorbedingungen		**Statistischer Test**
Normalverteilt	**Homoskedastizität**	
✓	✓	Zwei-Stichproben-t-Test
✓	-	Welch-Test
-	-	Mann-Whitney-U-Test

Für die Auswahl eines statistischen Testverfahrens muss zunächst die Vorbedingung der Normalität geprüft werden. Dafür wird der Shapiro-Wilk-Test [139] auf die metrischen Merkmalswerte jeder Teilnehmergruppe angewendet. Fällt der Shapiro-Wilk-Test signifikant aus, so sind die Merkmalswerte nicht normalverteilt [139]. Ist dies für mindestens eine Teilnehmergruppe der Fall, so müssen die Unterschiede des jeweiligen Merkmals mit dem Mann-Whitney-U-Test untersucht werden (vgl. ordinale Merkmale). Fällt der Shapiro-Wilk-Test hingegen für keine der Teilnehmergruppen signifikant aus, so muss weiterhin die Vorbedingung der Homoskedastizität (d. h. konstante Varianz der Merkmalswerte zwischen den Teilnehmergruppen) geprüft werden [129]. Da in diesem Fall schon bekannt ist, dass die Merkmalswerte normalverteilt sind, kann dafür der Bartlett-Test [1] angewendet werden. Fällt der Bartlett-Test signifikant aus, so weist mindestens eine Teilnehmergruppe abweichende Varianzen (Heteroskedastizität) der Merkmalswerte auf [1]. In diesem Fall müssen die Unterschiede des jeweiligen Merkmals zwischen den Gruppen mit dem Welch-Test [164] untersucht werden. Fällt der Bartlett-Test hingegen nicht signifikant aus, so müssen die Unterschiede des jeweiligen Merkmals zwischen den Teilnehmergruppen mit dem Zwei-Stichproben-t-Test [40] untersucht werden. Sowohl der Welch-Test als auch der Zwei-Stichproben-t-Test geben an, ob der Unterschied der Erwartungswerte zwischen den Teilnehmergruppen signifikant ist.

Hypothesentests für die Wahrnehmung der Aussagen

Der Mann-Whitney-U-Test einer Aussage V_i, $i \in \{V_1, V_2, \ldots, V_{96}\}$ sagt aus, ob es statistisch signifikante Unterschiede der Annotationen von V_i zwischen den Teilnehmergruppen gibt. Diese Ergebnisse dienen dazu, die 96 untergeordneten Nullhypothesen $\mathrm{H2}(V_1)_0$ bis $\mathrm{H2}(V_{96})_0$ sowie die übergeordnete Nullhypothese $\mathrm{H2}_0$ zu überprüfen, die in Tabelle 4.8 definiert sind.

Tabelle 4.8: Definition der übergeordneten Nullhypothese $\mathrm{H2}_0$ und der 96 untergeordneten Nullhypothesen $\mathrm{H2}(V_i)_0$.

	Definition der Nullhypothesen
$\mathrm{H2}_0$	Es gibt keine Unterschiede in den Annotationen zwischen den verschiedenen Teilnehmergruppen.
$\mathrm{H2}(V_i)_0$	Es gibt keine Unterschiede in den Annotationen von V_i, $i \in \{V_1, V_2, \ldots, V_{96}\}$ zwischen den verschiedenen Teilnehmergruppen.

Da, wie in Tabelle 4.8 definiert, 96 untergeordnete Nullhypothesen $\mathrm{H2}(V_i)_0$ mit dem Mann-Whitney-U-Test überprüft werden, muss das Signifikanzniveau α mit der Bonferroni-Korrektur [127] angepasst werden. Dadurch ergibt sich ein neues Signifikanzniveau von $\alpha = 0.05/96 \approx 0.000521$. Sobald eine der untergeordneten Nullhypothesen $\mathrm{H2}(V_1)_0$ bis $\mathrm{H2}(V_{96})_0$ das Signifikanzniveau α unterschreitet, kann also auch die übergeordnete Nullhypothese $\mathrm{H2}_0$ abgelehnt werden. Auch wenn im Rahmen der linearen Diskriminanzanalyse bereits Aussagen identifiziert werden konnten, die von den Teilnehmergruppen als unterschiedlich wahrgenommen wurden, kann so zusätzlich quantifiziert werden, für wie viele Aussagen dieser Unterschied auch statistisch signifikant ist.

Hypothesentests für die Wahrnehmung der Sentiment-Polaritäten

Um die Wahrnehmung der Teilnehmergruppen besser charakterisieren zu können, werden für jeden Studienteilnehmer die durchschnittlichen Anteile der Sentiment-Polaritäten *negativ*, *neutral* und *positiv*, von dessen vergebenen Annotationen berechnet (vgl. Kapitel 4.2.2). Diese drei Anteile werden als metrische Merkmale behandelt und entsprechend der erfüllten Vorbedingungen mit einem der statistischen Testverfahren aus Entscheidungstabelle 4.7 auf Unterschiede zwischen den Teilnehmergruppen untersucht. Anhand der Ergebnisse des resultierenden statistischen Testes können die drei untergeordneten Nullhypothesen $\mathrm{H3}(\mathit{negativ})_0$, $\mathrm{H3}(\mathit{neutral})_0$, und $\mathrm{H3}(\mathit{positiv})_0$, sowie die übergeordnete Nullhypothese $\mathrm{H3}_0$ überprüft werden. Diese sind in Tabelle 4.9 definiert.

Tabelle 4.9: Definition der übergeordneten Nullhypothese $H3_0$ und der drei untergeordneten Nullhypothesen $H3(P)_0$.

	Definition der Nullhypothesen
$H3_0$	Es gibt keine Unterschiede in den Anteilen der Sentiment-Polaritäten zwischen den verschiedenen Teilnehmergruppen.
$H3(P)_0$	Es gibt keine Unterschiede in den Anteilen der Sentiment-Polarität P zwischen den verschiedenen Teilnehmergruppen, wobei $P \in \{negativ, neutral, positiv\}$ gilt.

Da für die übergeordnete Nullhypothese $H3_0$ drei untergeordnete Nullhypothesen getestet werden (vgl. Tabelle 4.9), wird das Signifikanzniveau α der untergeordneten Nullhypothesen $H3(negativ)_0$, $H3(neutral)_0$, und $H3(positiv)_0$ mit der Bonferroni-Korrektur angepasst. Dadurch ergibt sich ein neues Signifikanzniveau $\alpha = 0.05/3 = 0.016\overline{6}$. Sobald das Signifikanzniveau α für die untergeordnete Nullhypothese $H3(P)_0$ einer der drei Sentiment-Polaritäten $P \in \{negativ, neutral, positiv\}$ unterschritten wird, kann folglich auch die übergeordnete Nullhypothese $H3_0$ abgelehnt werden.

Kapitel 5

Ergebnisse

In diesem Kapitel werden die Ergebnisse dieser Arbeit, welche durch die in Kapitel 4.3 beschriebenen Forschungsmethoden erarbeitet wurden, vorgestellt. Am Ende von jedem Abschnitt dieses Kapitels werden die wichtigsten Beobachtungen noch einmal zusammengefasst.

5.1 Ergebnisse der Korrelationsanalyse

Wie in Kapitel 4.3.1 beschrieben, wurde die Korrelation der Annotationen zwischen jedem Paar von Studienteilnehmer untersucht. Die Korrelation gibt einen Aufschluss darüber, wie stark die Wahrnehmung der Sentiment-Polaritäten von zwei Studienteilnehmern zusammenhängt. Die Korrelation von zwei Studienteilnehmern S_i und S_j wird durch Spearman's ϱ [145] als $\varrho(S_i, S_j)$ beschrieben (vgl. Kapitel 4.3.1).

5.1.1 Korrelationsmatrix der Studienteilnehmer

Die Korrelationsmatrix $\text{Corr}(S)$ enthält 4371 Einträge für jedes Paar aus den 94 Studienteilnehmern der Datenbasis (vgl. Gleichung 4.4). Abbildung 5.1 visualisiert die Korrelationsmatrix $\text{Corr}(S)$. Dabei ist jeder Korrelationswert $\varrho(S_i, S_j)$ durch ein eingefärbtes Quadrat visualisiert worden. Die Farbe der Korrelationswerte wurde durch den divergierenden Farbverlauf [100] zugeordnet, welcher in der Legende von Abbildung 5.1 dargestellt ist. Dabei repräsentieren blaue Farbtöne eine positive Korrelation zwischen den Annotationen zweier Studienteilnehmer, während rote Farbtöne mit einer negativen Korrelation der Annotationen assoziiert sind. Die Intensität der Farbe repräsentiert dabei jeweils die Stärke der absoluten Korrelation $|\varrho(S_i, S_j)|$. Somit sind dunkler eingefärbte Quadrate mit einem stärkeren Zusammenhang in der Wahrnehmung der betreffenden Studienteilnehmer verknüpft. Schwächer eingefärbte Quadrate deuten hingegen auf einen schwachen oder gar keinen Zusammenhang der Wahrnehmungen für die betreffenden Studienteilnehmer hin.

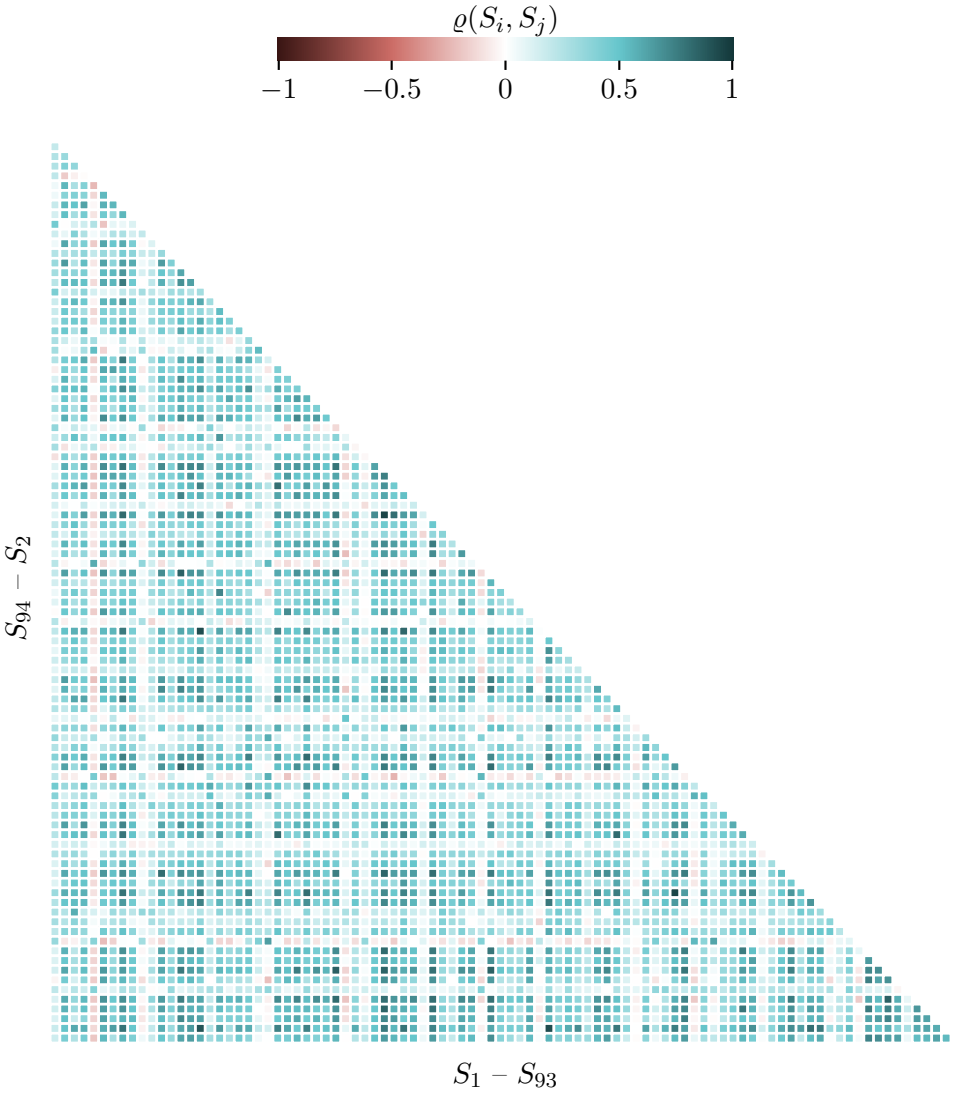

Abbildung 5.1: Korrelationsmatrix Corr(S) der Annotationen zwischen den Studienteilnehmern (vgl. Gleichung 4.4).

Abbildung 5.1 kann entnommen werden, dass es viele, teilweise auch starke, positive Korrelationen (bläuliche Quadrate) zwischen den Annotationen der Teilnehmer gibt, während es nur wenige negative Korrelationen (rötliche Quadrate) gibt. Von den insgesamt 4371 Paaren, von je zwei verschiedenen Studienteilnehmern aus der Datenbasis, weisen die Annotationen von 3968 Paaren (90.78 %) eine positive Korrelation ($\varrho > 0$) miteinander auf. Die Annotationen von 402 Paaren (9.20 %) von Studienteilnehmern weisen hingegen eine negative Korrelation ($\varrho < 0$) miteinander auf. Die Annotationen eines einzelnen Paares (0.02 %) von Studienteilnehmern weisen weder eine positive noch eine negative Korrelation ($\varrho = 0$) miteinander auf.

5.1.2 Hypothesenprüfung von H1$(S_i)_0$ und H1$(S_i, S_j)_0$

Um zu überprüfen, ob die Wahrnehmung der Studienteilnehmer statistisch signifikant zusammenhängt, wurden die 94 übergeordneten Nullhypothesen H1$(S_i)_0$ sowie die 4371 untergeordneten Nullhypothesen H1$(S_i, S_j)_0$ aufgestellt. Tabelle 5.1 zeigt einen für die Überprüfung der untergeordneten Nullhypothesen H1$(S_i, S_j)_0$ relevanten Ausschnitt der Korrelationswerte aus Corr(S), sowie der zugehörigen Wahrscheinlichkeitswerte $P(>|\varrho|)$. Tabelle 5.1 ist nach aufsteigenden Wahrscheinlichkeitswerten $P(>|\varrho|)$ sortiert worden, anschließend wurde jede Spalte von 1 bis 4371 nummeriert.

Tabelle 5.1: Korrelationswerte $\varrho(S_i, S_j)$ von je zwei Studienteilnehmern S_i und S_j, sowie die zugehörigen Wahrscheinlichkeitswerte $P(>|\varrho|)$.

Nr.	S_i	S_j	$\varrho(S_i, S_j)$	$P(>\|\varrho\|)$	**Interpretation**
1	S_{65}	S_{79}	0.9503	< 0.000001	**Signifikant**
⋮			⋮	⋮	⋮
2162	S_{71}	S_{17}	0.3468	0.000536	**Signifikant**
2163	S_{53}	S_{55}	0.3467	0.000540	Nicht signifikant
⋮			⋮	⋮	⋮
4371	S_{73}	S_{84}	0	1	Nicht signifikant

Von den 4371 Paaren, von je zwei verschiedenen Studienteilnehmern, korrelieren die Annotationen von 2162 Paaren (49.46 %) statistisch signifikant miteinander unter dem, nach der Bonferroni-Korrektur [127] auf $\alpha = 0.05/93 \approx 0.000538$ angepassten, Signifikanzniveau. Somit folgt, dass die zugehörige Nullhypothese H1$(S_i, S_j)_0$ in diesen 2162 Fällen abgelehnt werden kann. Von den statistisch signifikanten Korrelation sind alle 2162 Fälle positive Korrelationen ($\varrho > 0.3467$). Da für jeden der 94 Studienteilnehmer S_i der Datenbasis mindestens eine der 93 untergeordneten Nullhypothesen H1$(S_i, S_j)_0$ abgelehnt werden kann, können auch alle 94 übergeordneten Nullhypothesen H1$(S_1)_0$ bis H1$(S_{94})_0$ abgelehnt werden.

Beobachtung 5.1: Nahezu die Hälfte der Fälle (49.46 %) weist eine statistisch signifikante und positive Korrelation der Annotationen zwischen zwei Studienteilnehmern auf. Die restlichen Fälle weisen keine statistisch signifikanten Korrelationen auf. In diesen Fällen korrelieren die Annotationen der Teilnehmer entweder schwach positiv, schwach negativ, oder gar nicht miteinander. Die Annotationen von jedem der 94 Studienteilnehmer in der Datenbasis korrelieren mindestens mit den Annotationen eines weiteren Studienteilnehmers signifikant.

5.2 Ergebnisse der hierarchischen Clusteranalyse

Im Folgenden werden die Ergebnisse der hierarchischen Clusteranalyse, deren methodische Umsetzung in Kapitel 4.3.2 beschriebenen wurde, vorgestellt.

5.2.1 Resultierende Anzahl an Teilnehmerclustern

Für jede mögliche Anzahl an Teilnehmerclustern, d. h. jede Partitionierung der Studienteilnehmer aus den Ergebnissen der hierarchischen Clusteranalyse, wurde der Silhouettenkoeffizient [124] berechnet. Dieser gilt als Gütemaß für Clustering-Ergebnisse, und kann genutzt werden, um die optimale Anzahl an Teilnehmerclustern mit zusammenhängender Wahrnehmung in der Datenbasis dieser Arbeit zu ermitteln. Abbildung 5.2 zeigt den Silhouettenkoeffizienten in Abhängigkeit von der Anzahl an Teilnehmerclustern für zwei bis 94 Cluster. Die Anzahl an Teilnehmerclustern (auf der x-Achse) wurde dabei logarithmisch skaliert, da sich der Silhouettenkoeffizient zu Beginn des Graphen stärker verändert als im weiteren Verlauf. Wie Abbildung 5.2 zeigt, ist der Silhouettenkoeffizient für zwei Cluster maximal, und sinkt anschließend augenfällig ab. Der Silhouettenkoeffizient beträgt 0.3504, für die finalen zwei Cluster des hierarchischen Clusterings, im hochdimensionalen Raum. Für die maximale Anzahl von 94 Clustern ist der Silhouettenkoeffizient hingegen minimal. Unter anderem gibt es für zehn Cluster ein lokales Maximum des Silhouettenkoeffizienten, dieses ist jedoch im Vergleich zum initialen Maximum für zwei Teilnehmercluster niedrig.

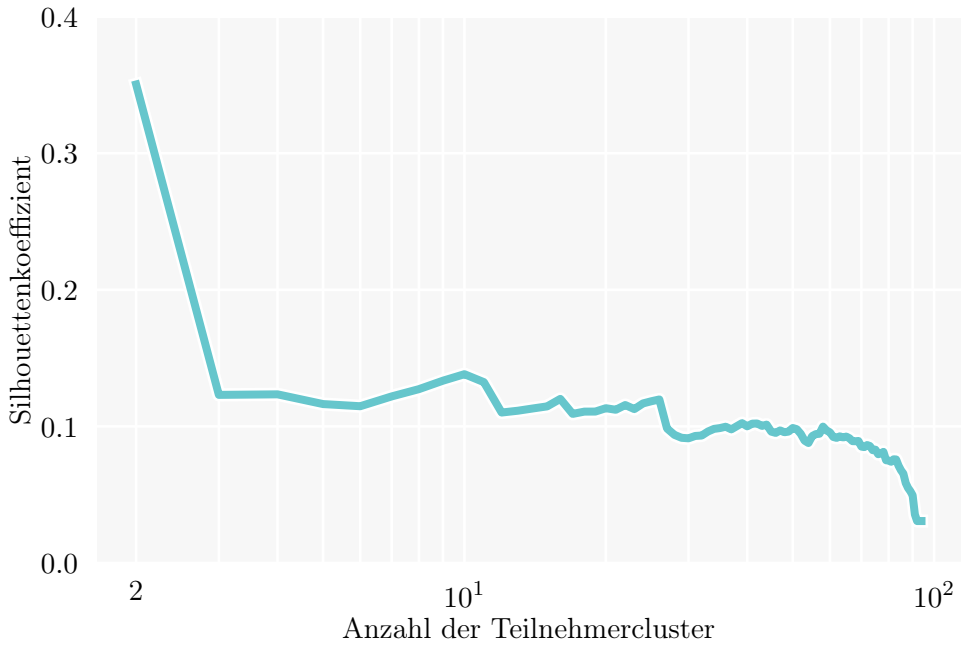

Abbildung 5.2: Der Silhouettenkoeffizient in Abhängigkeit von der Anzahl an Teilnehmerclustern (logarithmisch skaliert).

Damit lassen sich die Studienteilnehmer also objektiv am besten in zwei Teilnehmercluster aufteilen. Nach der Interpretation von Kaufman und Rousseeuw [71] entspricht der zugehörige Silhouettenkoeffizient von 0.3504 für diese beiden Cluster dabei einer schwachen Struktur (vgl. Tabelle 4.3). Der Silhouettenkoeffizient selbst wird jedoch dadurch abgeschwächt, dass die Datenbasis in einem hochdimensionalen Raum liegt [4]. Daher wird der Silhouettenkoeffizient für die dimensionsreduzierte Datenbasis im nachfolgenden Kapitel 5.3 erneut berechnet. Sicher ist jedoch, dass die zwei resultierenden Teilnehmercluster das optimale Verhältnis von maximaler Korrelation der Annotationen innerhalb der Cluster und minimaler Korrelation zwischen den Clustern aufweisen, welches von keiner höheren Anzahl an Teilnehmerclustern erreicht werden kann. Im weiteren Verlauf dieser Arbeit wird daher auf diese zwei Teilnehmercluster Bezug genommen.

5.2.2 Visualisierung der Teilnehmercluster

Die umliegende Abbildung 5.3 visualisiert die Ergebnisse der hierarchischen Clusteranalyse durch eine *Clustermap* [163], wie in Kapitel 4.3.2 beschrieben. Der innere Teil von Abbildung 5.3 ist durch die vollständige geordnete Korrelationsmatrix der Studienteilnehmer gegeben, wobei jedes Quadrat die Korrelation von einem Paar von Studienteilnehmern S_i und S_j identisch zu Abbildung 5.1 visualisiert. Oben links ist die Legende von Spearman's ϱ abgebildet, die Farben für die Korrelationswerte sind ebenfalls identisch zu denen in Abbildung 5.1. Am linken und oberen Rand ist jeweils das zugehörige Dendrogramm [57] des hierarchischen Clusterings abgebildet, wobei die Permutation der Studienteilnehmer in den Blättern identisch zu den S_i und S_j der geordneten Korrelationsmatrix ist (vgl. Kapitel 4.3.2).

Geordnete Korrelationsmatrix der Studienteilnehmer

Im Gegensatz zu Abbildung 5.1, sind in Abbildung 5.3 alle möglichen Kombinationen von S_i und S_j inkludiert. So befinden sich auf der Hauptdiagonalen der geordneten Korrelationsmatrix die Korrelationen eines jeden Teilnehmers mit sich selbst, welche allesamt einen maximalen Wert von $\varrho = 1$ annehmen. Ebenso sind auch die identischen Korrelationen $\varrho(S_i, S_j)$ und $\varrho(S_j, S_i)$ abgebildet, wodurch im Gegensatz zu Abbildung 5.1 keine untere Dreiecksmatrix, sondern eine symmetrische 94×94 Matrix entsteht. Die Reihenfolge der Studienteilnehmer S_i in den Spalten und S_j Zeilen entspricht der in Kapitel 4.3.2 beschriebenen Permutation der Ergebnisse des hierarchischen Clusterings, und damit auch der Reihenfolge der Blätter des links und oben an die geordnete Korrelationsmatrix angrenzenden Dendrogramms. Folglich ist die Reihenfolge der Studienteilnehmer in den Zeilen und Spalten der Korrelationsmatrix so angeordnet, dass sich Studienteilnehmer, deren Annotationen stark positiv miteinander korrelieren, in benachbarten Zeilen und Spalten befinden. Aufgrund dieser Anordnung konnten auch die zwei Teilnehmercluster, welche anhand des Silhouettenkoeffizienten als optimale

Anzahl an Teilnehmerclustern erfasst wurden, dargestellt werden. Wie Abbildung 5.3 zeigt, ergeben die kleineren Quadrate (also die Korrelationen der Annotationen von Studienteilnehmern) zwei größere blau eingefärbte Quadrate entlang der Hauptdiagonalen, welche jeweils die Korrelation der Studienteilnehmer innerhalb ihrer Teilnehmergruppe repräsentieren. Hierbei ist zu sehen, dass eines der beiden Teilnehmercluster bzw. eine Gruppe der Studienteilnehmer sehr viel kleiner ist, als die andere Gruppe. Die kleinere Gruppe der Studienteilnehmer (in Abbildung 5.3 links oben) macht die ersten 16 Indizes von S_i und S_j, und damit auch die ersten 16 Blätter des jeweiligen Dendrogramms aus. Die größere Gruppe der Studienteilnehmer befindet sich in den verbleibenden 78 Zeilen und Spalten der geordneten Korrelationsmatrix, sowie den verbleibenden 78 Blättern des jeweiligen Dendrogramms links und darüber. Zwischen den Gruppen gibt es negative (rot), nur leicht positive (schwach blau) oder gar keine (weiß) Korrelationen.

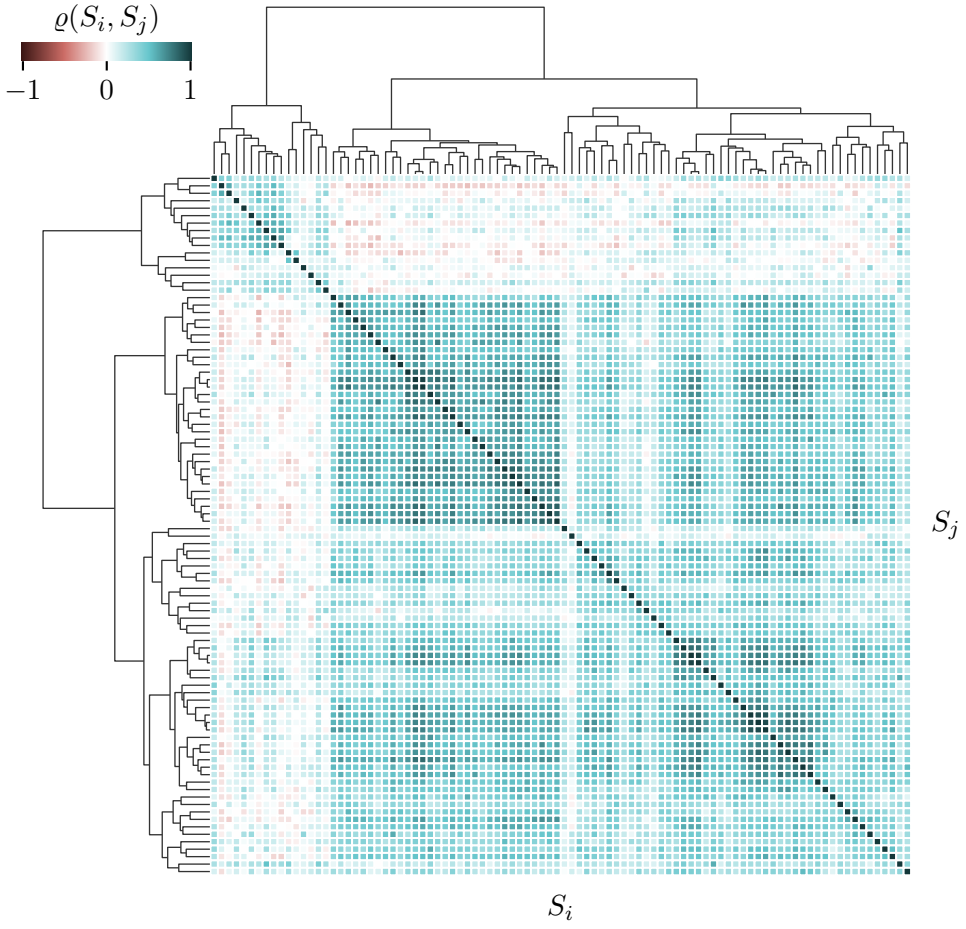

Abbildung 5.3: Seaborn *Clustermap* der Ergebnisse der hierarchischen Clusteranalyse (vgl. Kapitel 4.3.2).

Dendrogramm der Studienteilnehmer

Die Blätter der beiden umliegenden Dendrogramme in Abbildung 5.3 entsprechen den 94 Studienteilnehmern der Datenbasis in der Permutation, welche durch das hierarchische Clustering entstanden ist. Folglich sind die Studienteilnehmer, deren Annotationen stark positiv korrelieren, in den Blättern der Dendrogramme benachbart angeordnet, identisch zu den S_i und S_j der inneren geordneten Korrelationsmatrix. Durch das Dendrogramm lässt sich der hierarchische Aufbau der Teilnehmercluster nachverfolgen. Die Verbindungen von Blättern und Teilbäumen repräsentieren dabei jeweils eine Fusion zweier Cluster, wobei sich die Höhe der Verbindung durch die Distanz der fusionierten Cluster vor der Fusion ergibt. Die Distanz der Cluster wurde dafür durch Ward's minimales Varianzkriterium [162] berechnet. Wie auch in der Korrelationsmatrix im Inneren können die zwei Gruppen der Studienteilnehmer anhand des Dendrogramms erkannt werden. Die Distanz der letzten beiden Cluster ist vor der finalen Fusion des hierarchischen Clusterings sehr hoch im Vergleich zu den vorherigen Distanzen, was für eine gute Abgrenzung der Cluster spricht. Damit spiegelt das Dendrogramm den Verlauf des Silhouettenkoeffizienten aus Abbildung 5.2 wider, wobei ein Teilnehmercluster durch den linken Teilbaum (die ersten 16 Blätter von oben links aus) und das andere Teilnehmercluster durch den rechten Teilbaum (die restlichen 78 Blätter) des Dendrogramms visualisiert wird.

5.2.3 Definition der Teilnehmergruppen

Insgesamt bestätigt Abbildung 5.3, die Beobachtung des Silhouettenkoeffizienten in Abbildung 5.2, dass sich die Studienteilnehmer der Datenbasis am besten in zwei Cluster aufteilen lassen. Abbildung 5.3 verdeutlicht zudem den Größenunterschied dieser beiden Teilnehmercluster. Aus den vorangegangenen Ergebnissen lassen sich die nachfolgenden beiden Gruppen von Studienteilnehmern anhand ihrer Wahrnehmung definieren.

Erste Teilnehmergruppe: Der Großteil von 78 Studienteilnehmern lässt sich zu einer Teilnehmergruppe zusammenfassen. Innerhalb der ersten Teilnehmergruppe korreliert die Wahrnehmung der Studienteilnehmer stark positiv miteinander.

Zweite Teilnehmergruppe: Die Wahrnehmung einer Minderheit von 16 Studienteilnehmern korreliert hingegen nur schwach positiv, negativ, oder gar nicht mit der Wahrnehmung der ersten Teilnehmergruppe. Markant ist dabei jedoch, dass die Wahrnehmung dieser Minderheit von Studienteilnehmern untereinander wieder positiv miteinander korreliert.

Für den weiteren Verlauf dieser Arbeit wird die größere Gruppe der 78 stark positiv korrelierenden Studienteilnehmer als erste Teilnehmergruppe, kurz Gruppe 1, referenziert. Die Minderheit von 16 Studienteilnehmern, wird als zweite Teilnehmergruppe, kurz Gruppe 2, referenziert.

5.2.4 Interne Konsistenz der Teilnehmergruppen

Tabelle 5.2 zeigt die Werte von Cronbach's α [16], welche für beide Teilnehmergruppen berechnet wurden (vgl. Kapitel 4.3.2). Für die Interpretation der internen Konsistenz wurde die Skala von Hair et al. [48] genutzt.

Tabelle 5.2: Interne Konsistenz der beiden Teilnehmergruppen gemessen durch Cronbach's α und interpretiert nach Hair et al. [48].

Teilnehmergruppe	**Cronbach's α**	**Interne Konsistenz**
Gruppe 1	0.8640	Gut
Gruppe 2	0.9142	Exzellent

Wie Tabelle 5.2 zu entnehmen ist, weisen beide Teilnehmergruppen hohe Werte von Cronbach's α von 0.8640 für die erste Teilnehmergruppe und 0.9142 für die zweite Teilnehmergruppe auf. Diese Werte können nach Hair et al. [48] als eine gute interne Konsistenz für die erste Teilnehmergruppe und eine exzellente interne Konsistenz für die zweite Teilnehmergruppe interpretiert werden. Das bedeutet, dass die Aussagen, die von den Studienteilnehmern annotiert wurden, innerhalb beider Teilnehmergruppen dieselbe latente Variable messen. Es ist naheliegend, dass es sich bei dieser latenten Variable um die Wahrnehmung der Studienteilnehmer handelt, wobei jedoch nicht genauer beschrieben werden kann, was diese Wahrnehmung ausmacht.

Beobachtung 5.2: Durch die hierarchische Clusteranalyse konnten zwei Teilnehmergruppen identifiziert werden, welche die Datenbasis optimal nach der Wahrnehmung der Teilnehmer spalten. Die erste Teilnehmergruppe macht den Großteil von 78 Studienteilnehmern aus, deren Wahrnehmung stark positiv miteinander korreliert. Die zweite Gruppe macht die Minderheit von 16 Studienteilnehmern aus, deren Wahrnehmung nur schwach positiv, negativ, oder gar nicht mit der Wahrnehmung der ersten Teilnehmergruppe korreliert. Innerhalb der zweiten Gruppe korreliert die Wahrnehmung der Teilnehmer jedoch wiederum positiv miteinander.

5.3 Ergebnisse der Dimensionsreduktion

Wie in Kapitel 4.3.3 beschrieben wurden sowohl die Hauptkomponentenanalyse als auch die lineare Diskriminanzanalyse als Dimensionsreduktionsverfahren auf die Datenbasis dieser Arbeit angewendet. Ersteres bietet eine weitere Möglichkeit, die Ergebnisse der hierarchischen Clusteranalyse zu visualisieren, während letzteres auch dazu dient, die Wahrnehmung einzelner Aussagen zwischen den Teilnehmergruppen genauer zu betrachten. Im Folgenden werden die Ergebnisse beider Verfahren beschrieben.

5.3.1 Ergebnisse der Hauptkomponentenanalyse

Abbildung 5.4 visualisiert alle 94 Studienteilnehmer der Datenbasis als Punkte in einem Streudiagramm nach der Transformation der Datenbasis in einen zweidimensionalen Raum. Dieser Raum wird durch die ersten beiden Hauptkomponenten aufgestellt, welche aus den Annotationen aller Studienteilnehmer errechnet wurden (vgl. Kapitel 4.3.3). Auf der x-Achse befindet sich die erste Hauptkomponente (PC1) mit einer abgebildeten Varianz von 16.86 %. Auf der y-Achse befindet sich die zweite Hauptkomponente (PC2) mit einer abgebildeten Varianz von 8.42 %. Insgesamt werden 25.28 % der Gesamtvarianz aus dem hochdimensionalen Raum (d. h. aus den ursprünglichen 96 Annotationen pro Studienteilnehmer) abgebildet. Die Positionen der Studienteilnehmer sind jedoch unabhängig von der Kenntnis über die beiden Teilnehmergruppen errechnet worden. Für jeden Studienteilnehmer wurden die x- und y-Koordinaten durch Skalarmultiplikation der ersten und zweiten Hauptkomponente mit den Rängen der von dem jeweiligen Studienteilnehmer gewählten Sentiment-Polaritäten für die 96 Aussagen der Datenbasis errechnet. Anschließend wurde jeder Studienteilnehmer entsprechend seiner x- und y-Koordinaten im Streudiagramm in Abbildung 5.4 aufgetragen. Abschließend wurde der Punkt eines jeden Studienteilnehmers entsprechend seiner Zugehörigkeit zur ersten (blau) oder zweiten (rot) Teilnehmergruppe koloriert.

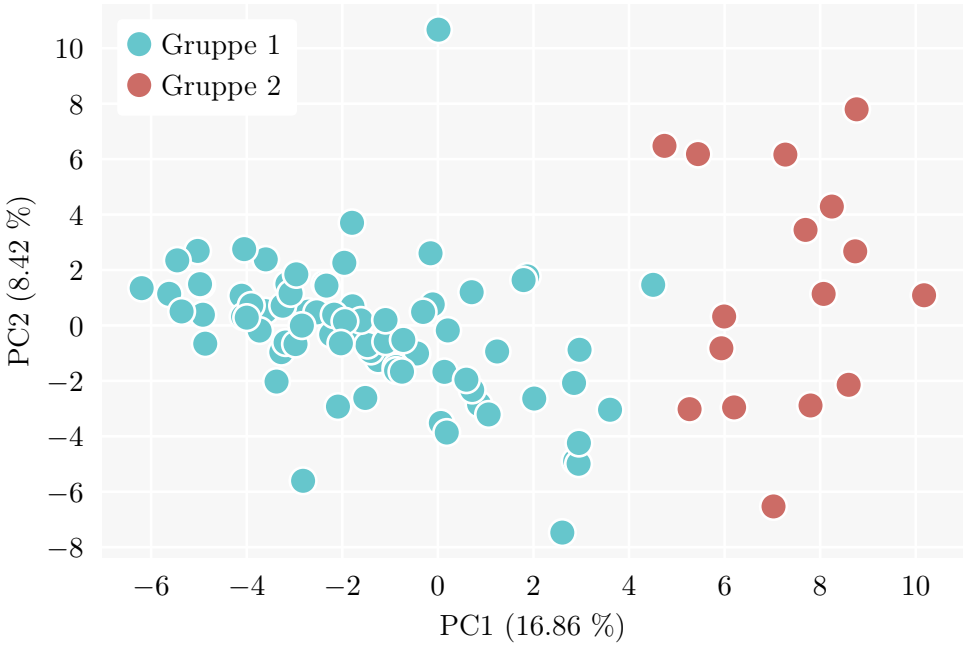

Abbildung 5.4: Durch die Hauptkomponentenanalyse dimensionsreduziertes Streudiagramm aller 94 Studienteilnehmer aus der Datenbasis.

Anhand von Abbildung 5.4 ist eine optische Abgrenzung beider Teilnehmergruppen im durch die Hauptkomponentenanalyse dimensionsreduzierten Raum gut erkennbar. Die Unterschiede scheinen dabei hauptsächlich durch die x-Position, also durch die erste Hauptkomponente, hervorgerufen zu werden. Abgesehen von einigen Ausreißern ist das Cluster von Gruppe 1 sehr viel dichter und definierter im niedrigerdimensionalen Raum erkennbar als das Cluster von Gruppe 2. Dass die beiden erkennbaren Cluster von Gruppe 1 und Gruppe 2 nicht nur eine visuelle Täuschung sind, kann durch die Berechnung des Silhouettenkoeffizienten im zweidimensionalen transformierten Raum bestätigt werden. Für die Koordinaten in Abbildung 5.4 nimmt der Silhouettenkoeffizient beider Cluster von Gruppe 1 und Gruppe 2 einen Wert von 0.5429 an. Nach Kaufman und Rousseeuw [71] (vgl. Tabelle 4.3) kann ein Silhouettenkoeffizient von 0.5429 als eine sinnvolle Struktur interpretiert werden, die so auch tatsächlich in der Datenbasis vorliegt.

5.3.2 Ergebnisse der linearen Diskriminanzanalyse

Da anhand der hierarchischen Clusteranalyse zwei Teilnehmergruppen in der Datenbasis identifiziert wurden, können diese nur durch eine einzelne Diskriminante separiert werden (vgl. Kapitel 2.4.3). Dadurch wird die Datenbasis in einen eindimensionalen Raum, d. h. eine Gerade, auf welcher sich die Studienteilnehmer befinden, transformiert. Abbildung 5.5 zeigt die Häufigkeitsverteilung der Studienteilnehmer auf dieser Geraden als Histogramm.

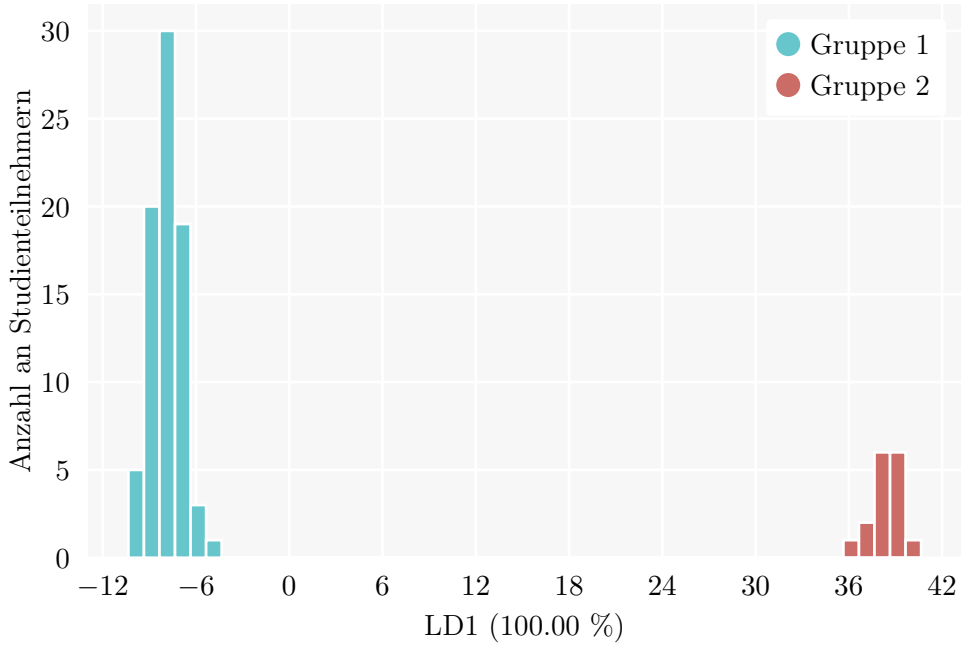

Abbildung 5.5: Histogramm der Studienteilnehmer in dem durch die lineare Diskriminanzanalyse auf eine Dimension reduzierten Raum.

Die Balken des Histogramms sind entsprechend der Zugehörigkeit der jeweiligen Studienteilnehmer zu Gruppe 1 (blau) und Gruppe 2 (rot) koloriert. Die Separation der beiden Teilnehmergruppen kann zu 100 % durch die erste und einzige lineare Diskriminante (LD1) abgebildet werden, wie Abbildung 5.5 aufzeigt. Alle 78 Studienteilnehmer der ersten Teilnehmergruppe befinden sich dabei auf der linken Seite des Histogramms ($5+20+30+19+3+1$). Auf der rechten Seite des Histogramms befinden sich hingegen alle 16 Studienteilnehmer, die Gruppe 2 angehören ($1+2+6+6+1$). Wie zu sehen ist, sind beide Gruppen von Teilnehmern im Histogramm sehr dicht und klar voneinander abgegrenzt repräsentiert. Durch die lineare Diskriminanzanalyse war es also möglich, die Zuordnung der Studienteilnehmer in ihre jeweilige Gruppe anhand einer linearen Kombination ihrer Annotation (oder deren Rängen) vorzunehmen und die beiden Teilnehmergruppen dabei deutlich voneinander zu separieren.

Gewichtung der Aussagen

Die Koeffizienten der 96 Aussagen, welche durch die lineare Diskriminanzanalyse bestimmt wurden, sind in Abbildung 5.6 dargestellt. Die y-Achse zeigt dabei den Betrag $|K|$ eines jeden Koeffizienten, also das absolute Gewicht einer Aussage. Die Beträge der Koeffizienten wurden dabei so normiert, dass der kleinste Wert eins beträgt. Auf der x-Achse sind alle Aussagen V_i nach ihrem aufsteigenden absoluten Koeffizienten $|K|$ angeordnet aufgetragen.

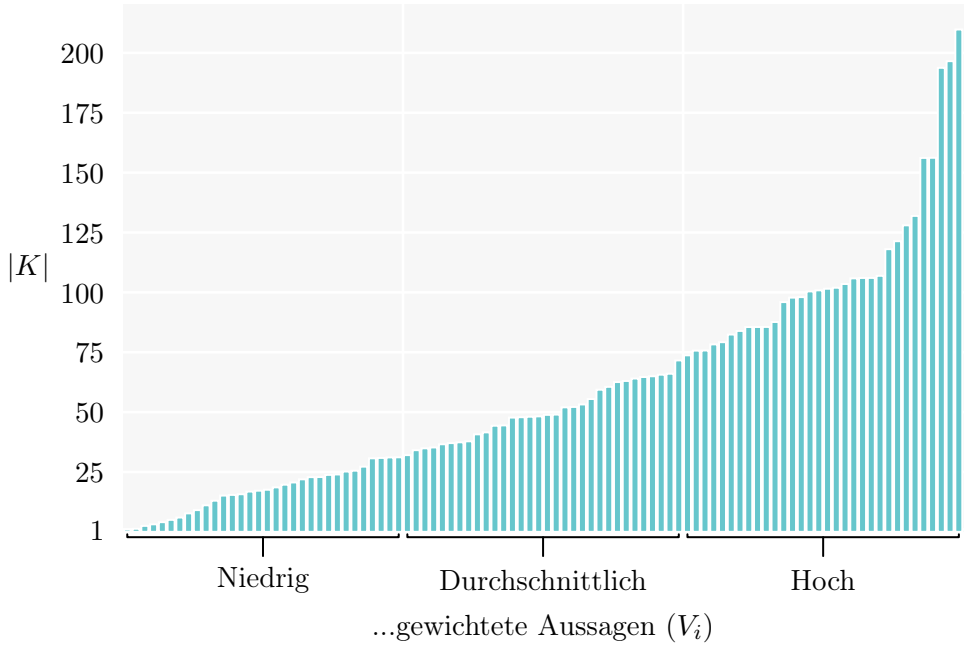

Abbildung 5.6: Verteilung der absoluten Koeffizienten $|K|$ der Aussagen V_1 bis V_{96}, welche durch die lineare Diskriminanzanalyse berechnet wurden.

Je höher der Betrag des Koeffizienten $|K|$ einer Aussage ist, desto höher wurde der Einfluss dieser Aussage auf die Separation der beiden Teilnehmergruppen durch die lineare Diskriminanzanalyse gewichtet. Da der minimale Koeffizient auf einen Wert von eins normiert wurde, geben die Werte von $|K|$ in Abbildung 5.6 also an, um welches Vielfache eine Aussage höher gewichtet wurde, als die am niedrigsten gewichtete Aussage. Für die ersten 78 Aussagen in Abbildung 5.6 scheinen die Gewichte linear bis auf das 100-fache des kleinsten Gewichtes anzusteigen. Danach wachsen die Gewichte der verbleibenden Aussagen jedoch exponentiell an. Dreizehn Aussagen wurden 100- bis 150-mal höher gewichtet, als die am niedrigsten gewichtete Aussage. Weiterhin wurden vier verschiedene Aussagen 150- bis 200-mal höher gewichtet, als die am niedrigsten gewichtete Aussage. Letztlich übersteigt nur eine einzelne Aussage das 200-fache Gewicht, der am niedrigsten gewichteten Aussage.

Hoch gewichtete Aussagen

Tabelle 5.3 zeigt fünf der überdurchschnittlich hoch gewichteten Aussagen aus Abbildung 5.6, sowie deren absolute Koeffizienten $|K|$. Zusätzlich ist für jede Aussage die prozentuale Verteilung der Sentiment-Polaritäten für beide Teilnehmergruppen ($G = 1$ und $G = 2$) dargestellt.

Tabelle 5.3: Fünf Aussagen, welche überdurchschnittlich hoch für die Separation der beiden Teilnehmergruppen gewichtet wurden.

$\|K\|$	Aussage	G	Anteile in %		
			Negativ	Neutral	Positiv
209.8	*lol :)*	1	11.54	23.08	65.38
		2	68.75	25.00	6.25
193.8	*Lol.*	1	25.64	41.03	33.33
		2	87.50	12.50	0.00
156.2	*@timmywil Sounds good!*	1	2.56	0.00	97.44
		2	18.75	50.00	31.25
131.9	*Most awesome! :+1:*	1	0.00	2.56	97.44
		2	6.25	62.50	31.25
106.9	*Hope this helps.*	1	2.56	16.67	80.77
		2	0.00	43.75	56.25

Die fünf Aussagen in Tabelle 5.3 sind allesamt unter den zehn am stärksten gewichteten Aussagen der Datenbasis. Diese fünf Aussagen wurden ausgewählt, da sie am wenigsten Sonderzeichen und technischen Jargon enthalten, und damit am verständlichsten präsentiert werden können. Die erste Aussage („*lol :)*“) ist die einzige Aussage mit einem Koeffizienten, dessen Betrag größer als 200 ist. Damit ist dies die, durch die lineare Diskriminanzanalyse, am stärksten gewichtete Aussage von allen Aussagen der Datenbasis. Die Verteilung der Sentiment-Polaritäten zeigt, dass Teilnehmer aus Gruppe 1 diese Aussage zu 65.38 % als *positiv* wahrnehmen, während die Teilnehmer aus Gruppe 2 dieselbe Aussage zu 68.75 % als *negativ* wahrnehmen. Die zweite Aussage („*Lol.*“) ist die am dritthöchsten gewichtete Aussage und inhaltlich nahezu identisch mit der ersten Aussage in Tabelle 5.3. Diese Aussage wird von 25.64 % der Teilnehmer aus Gruppe 1 als *negativ* wahrgenommen, im Gegensatz zu 87.5 % der Teilnehmer aus Gruppe 2. Die Aussagen „*@timmywil Sounds good!*“ und „*Most awesome! :+1:*“ werden von Gruppe 1 fast ausschließlich als *positiv* wahrgenommen, während weniger als ein Drittel der Teilnehmer aus Gruppe 2 diese Aussagen als *positiv* wahrnimmt. Die letzte Aussage („*Hope this helps.*“) wird von Teilnehmern der ersten Gruppe zu 80.77 % als *positiv* wahrgenommen, während im Vergleich nur 56.25 % der Teilnehmer aus Gruppe 2 diese Aussage als *positiv* wahrnehmen. Dafür wird diese Aussage von Teilnehmern aus Gruppe 2 jedoch häufiger *neutral* wahrgenommen (43.75 % im Vergleich zu 16.67 %). Insgesamt lassen sich für alle der hoch gewichteten Aussagen deutliche Abweichungen in den Verteilungen der drei Sentiments-Polaritäten zwischen den beiden Teilnehmergruppen beobachten. Diese Aussagen sind also, wie durch die lineare Diskriminanzanalyse ermittelt, tatsächlich stark polarisierend. In fast allen Fällen ist die Sentiments-Polarität mit den meisten Annotationen für beide Gruppen unterschiedlich, es gibt jedoch auch Ausnahmen, wie die letzte Aussage in Tabelle 5.3, welche beide Gruppen mehrheitlich als *positiv* wahrnehmen.

Niedrig gewichtete Aussagen

Im Gegensatz zu den Aussagen in Tabelle 5.3, enthält Tabelle 5.4 (auf der umliegenden Seite) Aussagen, welche von der linearen Diskriminanzanalyse überdurchschnittlich niedrig gewichtet wurden. Das heißt, dass diese Aussagen keinen starken Einfluss auf die lineare Kombination der Aussagen haben, welche die beiden Teilnehmergruppen voneinander separiert. Der Betrag des Koeffizienten $|K|$ der Aussagen in Tabelle 5.4 ist entsprechend deutlich niedriger als in Tabelle 5.3. Die fünf Aussagen in Tabelle 5.4 sind allesamt unter den 16 am niedrigsten gewichteten Aussagen der Datenbasis. Die Aussagen in Tabelle 5.4 wurden ähnlich wie in Tabelle 5.3 nach ihrer Länge und Lesbarkeit ausgewählt, da sie wenige Sonderzeichen und technischen Jargon enthalten, und damit am geeignetsten für die Präsentation sind.

Tabelle 5.4: Fünf Aussagen, welche überdurchschnittlich niedrig für die Separation der Teilnehmergruppen gewichtet wurden (vgl. Tabelle 5.3).

$\|K\|$	Aussage	G	Anteile in %		
			Negativ	Neutral	Positiv
17.2	*...because this is how it was before :)*	1	34.62	50.00	15.38
		2	25.00	62.50	12.50
9.1	*OMG stupid me*	1	30.77	42.31	26.92
		2	37.50	50.00	12.50
3.1	*Yay for improving consistency, +1*	1	5.13	8.97	85.90
		2	18.75	43.75	37.50
2.5	*else your GUI will be Hanged.*	1	39.74	55.13	5.13
		2	25.00	43.75	31.25
1	*Is this good to go then?*	1	2.56	74.36	23.08
		2	18.75	68.75	12.50

Die erste Aussage in Tabelle 5.4 („*...because this is how it was before :)*“) wird von beiden Teilnehmergruppen mehrheitlich als *neutral* wahrgenommen (50 % und 62.5 %). Die zweite Aussage („*OMG stupid me*“) wird ebenfalls von beiden Teilnehmergruppen am häufigsten als *neutral* wahrgenommen, mit einem Anteil von jeweils 42.31 % und 50 %. Auffällig ist, dass die Sentiments-Polaritäten im Vergleich zu den vorherigen Beispielen in Tabelle 5.3 sehr gleichmäßig verteilt sind, d. h. dass sich auch die Studienteilnehmer innerhalb beider Teilnehmergruppen für diese Aussagen eher uneinig sind. Die Aussage „*Yay for improving consistency, +1*“ bildet eine Ausnahme in Tabelle 5.4. Diese wird von Teilnehmern aus Gruppe 1 zu 85.9 % als *positiv* wahrgenommen, während nur 37.5 % der Teilnehmer aus Gruppe 2 diese Wahrnehmung teilen. Stattdessen nehmen 43.75 % der Teilnehmer aus Gruppe 2 diese Aussage als *neutral* wahr, im Vergleich zu nur 8.97 % aus Gruppe 1. Die vorletzte Aussage („*else your GUI will be Hanged.*“) wird von beiden Teilnehmergruppen wieder mehrheitlich als *neutral* wahrgenommen. Deutlich mehr Teilnehmer aus Gruppe 2 nehmen diese Aussagen jedoch als *positiv* wahr, mit 31.25 % zu 5.13 %. Die letzte Aussage in Tabelle 5.4 („*Is this good to go then?*“) hat einen Koeffizient mit einem Betrag von eins, und ist damit die am niedrigsten gewichtete Aussagen von allen. Diese Aussage wird von 74.36 % der Teilnehmer aus Gruppe 1 und 68.75 % der Teilnehmer aus

Gruppe 2 als *neutral* wahrgenommen. Teilnehmer der ersten Gruppe nehmen diese Aussage zudem öfter *positiv* wahr, während Teilnehmer der zweiten Gruppe diese Aussage häufiger als *negativ* wahrnehmen. Zusammenfassend fällt auf, dass Aussagen, denen durch die lineare Diskriminanzanalyse ein niedriger Koeffizient zugeordnet wurde, oft mehrheitlich von beiden Teilnehmergruppen als *neutral* wahrgenommen werden. Auch wenn es in Tabelle 5.4 ein Gegenbeispiel dafür gibt, scheinen die Annotationen dieser Aussagen also tatsächlich keine wertvollen Informationen für die Separation der beiden Teilnehmergruppen zu enthalten.

Beobachtung 5.3: Die Cluster beider Teilnehmergruppen sind im durch die Hauptkomponentenanalyse unüberwacht dimensionsreduzierten Raum deutlich erkennbar, überschneiden sich nicht, und bilden eine sinnvolle Struktur. Durch die überwachte Dimensionsreduktion mit der lineare Diskriminanzanalyse wurde beobachtet, dass verschiedene Aussagen der Datenbasis einen unterschiedlich hohen Einfluss auf die Separation der Studienteilnehmer nehmen. Dabei unterscheidet sich die Verteilung der Sentiment-Polaritäten, für Aussagen mit hohem Einfluss deutlich zwischen den Teilnehmergruppen, während die Verteilung für Aussagen mit niedrigem Einfluss beinahe identisch ist.

5.4 Ergebnisse der logistischen Regressionsanalyse

Die logistische Regressionsanalyse wurde im Rahmen dieser Arbeit auf die Datenbasis angewendet, um festzustellen, wie viele der 96 Aussagen nötig sind, um ein logistisches Regressionsmodell zu berechnen, welches in der Lage ist, die korrekte Teilnehmergruppe für alle 94 Studienteilnehmer vorherzusagen (vgl. Kapitel 4.3.4).

5.4.1 Anzahl der Prädiktorvariablen

Für die Anwendung der logistischen Regressionsanalyse wurde, wie in Kapitel 4.3.4 beschrieben, zunächst eine Merkmalsselektion der Aussagen $V = \{V_1, V_2, \ldots, V_{96}\}$ durch die lineare Diskriminanzanalyse vorgenommen. Dabei wurden für jede Anzahl n, $1 \leq n \leq 95$ die n wichtigsten Aussagen der Datenbasis für die Separation der Teilnehmergruppen selektiert. Anschließend wurde durch die logistische Regressionsanalyse nach Firth [31, 51] für jedes n anhand der n wichtigsten Aussagen ein logistisches Regressionsmodell zur Vorhersage der Teilnehmergruppen berechnet und dessen Vorhersagegenauigkeit evaluiert. Tabelle 5.5 zeigt die Vorhersagegenauigkeiten des jeweiligen logistischen Regressionsmodells in Abhängigkeit von der Anzahl $n = |X|$, der vom Regressionsmodell für die Vorhersage der Teilnehmergruppen genutzten Prädiktorvariablen $X \in V$. Neben den absoluten und prozentualen Vorhersagegenauigkeiten ist für jedes $|X|$ in Tabelle 5.5 angegeben, ob die beiden Teilnehmergruppen vollständig separiert werden können.

Tabelle 5.5: Vorhersagegenauigkeit des logistischen Regressionsmodells in Abhängigkeit von der Anzahl $|X|$ der Prädiktorvariablen $X \in V$.

$\|X\|$	X	Genauigkeit abs.	Genauigkeit in %	Separation der Teilnehmergruppen
1	$\{V_{80}\}$	85	92.43	Unvollständig
2	$\{V_{78}, V_{80}\}$	88	93.92	⋮
3	$\{V_{21}, V_{78}, V_{80}\}$	90	95.74	
4	$\{V_{21}, V_{49}, V_{78}, V_{80}\}$	93	98.94	Unvollständig
5	$\{V_{21}, V_{49}, V_{57}, V_{78}, V_{80}\}$	94	100.00	**Vollständig**
⋮	⋮	⋮	⋮	⋮
96	$\{V_1, V_2, \ldots, V_{96}\}$	94	100.00	**Vollständig**

Wie Tabelle 5.5 zu entnehmen ist, konnten anhand der (lt. der linearen Diskriminanzanalysen) wichtigsten Aussage V_{80}, für 85 Studienteilnehmer die korrekte Teilnehmergruppe vorhergesagt werden, was einer Vorhersagegenauigkeit von 92.43 % entspricht. Durch Hinzunahme der Aussage V_{78} konnten im nächsten Schritt für 88 Teilnehmer die korrekte Gruppe vorhergesagt werden. Somit wurde durch die zwei Prädiktorvariablen $\{V_{78}, V_{80}\}$ eine Vorhersagegenauigkeit von 93.92 % erreicht. Im nächsten Schritt wurde den Prädiktorvariablen die Aussage V_{21} hinzugefügt, was zu einer Steigerung der Vorhersagegenauigkeit auf 95.74 % führte, da für 90 Studienteilnehmer die korrekte Teilnehmergruppe vorhergesagt werden konnte. Das Hinzufügen der Aussage V_{49} hatte zufolge, dass nur noch ein einzelner Studienteilnehmer nicht seiner korrekten Teilnehmergruppe zugeordnet werden konnte, was einer Vorhersagegenauigkeit von 98.94 % entspricht. Eine korrekte Vorhersage aller Studienteilnehmer, und damit eine vollständige Separation beider Teilnehmergruppen gelang durch das Hinzufügen der Aussage V_{57}. Das Ensemble der Prädiktorvariablen ist damit durch $\{V_{21}, V_{49}, V_{57}, V_{78}, V_{80}\}$ gegeben. Alle Aussagen V_1 bis V_{96} der Datenbasis befinden sich in Anhang A.2 dieser Arbeit.

5.4.2 Vorhersage der Teilnehmergruppen

Tabelle 5.6 stellt die Parameter des logistischen Regressionsmodells, d. h. die fünf Prädiktorvariablen, sowie deren Regressionskoeffizienten β_i und Standardfehler σ_i dar. Zudem sind die zugehörigen z-Werte und deren Wahrscheinlichkeit $P(>|z|)$ nach dem Wald-Test [161] angegeben. Gilt $P(>|z|) < 0.05$, so hat die jeweilige Prädiktorvariable einen signifikanten Einfluss auf die Vorhersage der Teilnehmergruppe für die Studienteilnehmer der Datenbasis durch das logistische Regressionsmodell.

Tabelle 5.6: Parameter des logistischen Regressionsmodells zur Vorhersage der Teilnehmergruppen für die Studienteilnehmer der Datenbasis.

V_i	β_i	σ_i	z	$P(>\|z\|)$	Interpretation
V_{21}	4.4901	1.8068	2.4851	0.0130	**Signifikant**
V_{49}	3.7397	1.8545	2.0166	0.0437	**Signifikant**
V_{57}	−1.3372	1.3942	−0.9591	0.3375	Nicht signifikant
V_{78}	−2.2945	0.9235	−2.4845	0.0130	**Signifikant**
V_{80}	−3.5870	1.4477	−2.4778	0.0132	**Signifikant**

Tabelle 5.6 zeigt, dass alle Prädiktorvariablen außer V_{57} einen signifikanten Einfluss auf die Vorhersage der Teilnehmergruppe durch das Regressionsmodell haben. Dies spiegelt die Reihenfolge der Wichtigkeit der Variablen wider, welche durch die lineare Diskriminanzanalyse bestimmt wurde, da V_{57} den Prädiktorvariablen als letztes hinzugefügt wurde. Die Variable V_{49} wurde den Prädiktorvariablen als vorletztes hinzugefügt und liegt im Gegensatz zu V_{21}, V_{78} und V_{80} nur knapp unter dem Signifikanzniveau von $\alpha = 0.05$. Für β_0 (nicht in der Tabelle 5.6 abgebildet) ergab sich ein Wert von -1.932. Der Vektor der Regressionskoeffizienten $\beta = \{\beta_0, \beta_{21}, \beta_{49}, \beta_{57}, \beta_{78}, \beta_{80}\}$ ergibt zusammen mit den Prädiktorvariablen $X = \{V_{21}, V_{49}, V_{57}, V_{78}, V_{80}\}$ den linearen Prädiktor ω als das folgende Produkt (vgl. Kapitel 2.5).

$$\omega = \beta_0 + \beta_{21}V_{21} + \beta_{49}V_{49} + \beta_{57}V_{57} + \beta_{78}V_{78} + \beta_{80}V_{80} \tag{5.1}$$

Für jeden Studienteilnehmer wird der lineare Prädiktor ω unter der Belegung von $\{V_{21}, V_{49}, V_{57}, V_{78}, V_{80}\}$ mit den Rängen der Sentiment-Polaritäten *negativ*, *neutral* und *positiv*, welche der Teilnehmer für diese fünf Aussagen wahrgenommen hat, berechnet. Die Wahrscheinlichkeit, dass sich ein Studienteilnehmer in der zweiten Teilnehmergruppe befindet, wird anschließend folgendermaßen berechnet (vgl. Kapitel 2.5).

$$P(\text{Gruppe } 2 \mid X) = \frac{e^{\omega}}{1 + e^{\omega}} \tag{5.2}$$

Abbildung 5.7 stellt die Wahrscheinlichkeit $P(\text{Gruppe } 2 \mid X)$ in Abhängigkeit vom linearen Prädiktor ω dar. Der jeweilige *Rug Plot* [114] am unteren und oberen Rand von Abbildung 5.7 visualisiert tatsächlichen Werte, von ω für die 94 Studienteilnehmer der Datenbasis. Die Werte von ω für die Studienteilnehmer der ersten Gruppe wurden unter, und die Werte der zweiten Teilnehmergruppen über, der logistischen Funktion platziert. Zudem wurde der jeweilige *Rug Plot* sowie der Verlauf des Graphen selbst entsprechend der beiden Teilnehmergruppen koloriert.

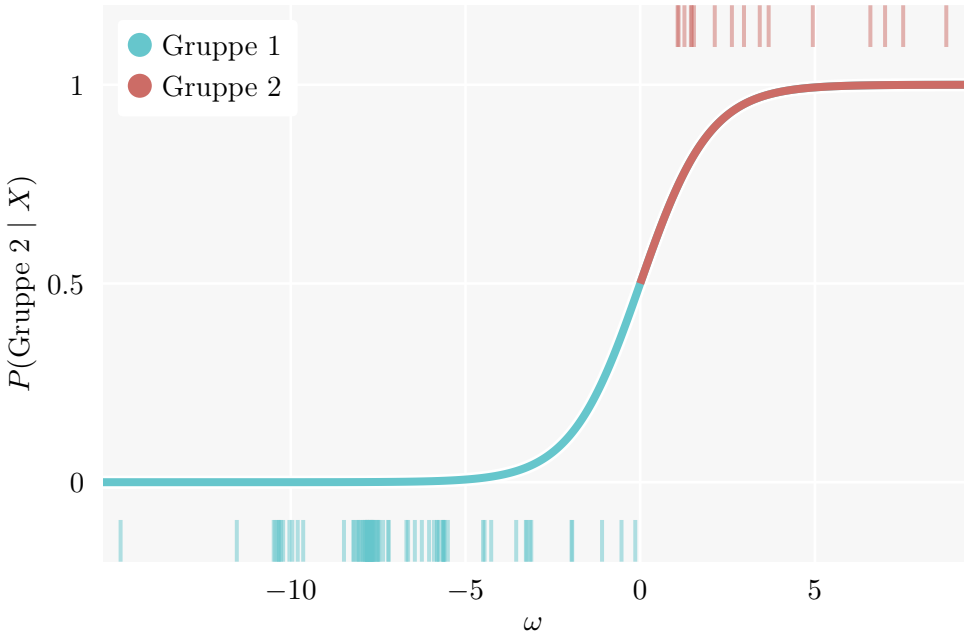

Abbildung 5.7: Die Wahrscheinlichkeit $P(\text{Gruppe } 2 \mid X)$ in Abhängigkeit vom linearen Prädiktor ω zur Vorhersage der Teilnehmergruppen.

Wie Abbildung 5.7 darstellt, erreicht der lineare Prädiktor für einen Studienteilnehmer der ersten Gruppe einen Wert von nahezu null. Damit liegt die Wahrscheinlichkeit, dass dieser Teilnehmer der zweiten Gruppe zuzuordnen ist, nur knapp unter 50 % und der Teilnehmer wird seiner tatsächlichen Teilnehmergruppe (Gruppe 1) zugeordnet. Es ist also naheliegend, dass das logistische Regressionsmodell für vier Prädiktorvariablen nicht mehr in der Lage war, diesen Teilnehmer seiner korrekten Gruppe zuzuordnen, weshalb für $|X| = 4$ nur für 93 Studienteilnehmer die korrekte Teilnehmergruppe vorhergesagt werden konnte (vgl. Tabelle 5.5). Dass mit nur fünf Aussagen, also nur knapp über 5 % der ursprünglich Aussagen der Datenbasis eine korrekte Vorhersage der Teilnehmergruppe für alle Studienteilnehmer entsprechend ihrer Wahrnehmung möglich ist, ist dennoch bemerkenswert. Dieses Ergebnis zeigt, dass die Annotationen weniger sorgfältig ausgewählter Aussagen genauso dazu in der Lage sind, die Teilnehmer entsprechend ihrer Wahrnehmung zu separieren, wie die gesamten Annotationen aller 96 Aussagen aus der Datenbasis.

Beobachtung 5.4: Für alle 94 Studienteilnehmer kann anhand der Annotationen von nur fünf (5.21 %) der insgesamt 96 Aussagen aus der Datenbasis die korrekte Teilnehmergruppe, entsprechend der Wahrnehmung eines jeden Teilnehmers, durch ein logistisches Regressionsmodell vorhergesagt werden.

5.5 Statistischer Vergleich der Teilnehmergruppen

Für den statistischen Vergleich der verschiedenen Merkmale zwischen den Teilnehmergruppen werden zunächst die Ergebnisse der Testverfahren für Vorbedingungen der metrischen Merkmale präsentiert, welche in Kapitel 4.3.5 beschrieben wurden. Anschließend werden die Ergebnisse des statistischen Vergleiches für das Geschlecht und Alter, das sprachliche Verständnis, die berufliche Erfahrung, die Annotationskriterien und die Wahrnehmung der Teilnehmergruppen präsentiert.

5.5.1 Ergebnisse des Shapiro-Wilk-Testes

Um einen geeigneten statistischen Test zur Untersuchung der Unterschiede zwischen den Teilnehmergruppen für die metrischen Merkmale der Datenbasis zu ermitteln, wurde zunächst der Shapiro-Wilk-Test [139] für die jeweiligen Merkmalswerte beider Teilnehmergruppen durchgeführt. Tabelle 5.7 zeigt die Ergebnisse des Shapiro-Wilk-Testes, für alle metrischen Merkmale. Signifikante Ergebnisse des Shapiro-Wilk-Testes sind dabei hervorgehoben und geben an, dass die jeweiligen Merkmalswerte nicht normalverteilt sind.

Tabelle 5.7: Ergebnisse des Shapiro-Wilk-Testes für die metrischen Merkmalswerte beider Teilnehmergruppen ($G = 1$ und $G = 2$).

Merkmal	G	W	p	**Statistischer Test**
Alter der Teilnehmer	1	0.8883	$<$ **0.0001**	Mann-Whitney-U-Test
	2	0.9452	0.4526	
Berufsjahre als Entwickler	1	0.6789	$<$ **0.0001**	Mann-Whitney-U-Test
	2	0.7293	**0.0005**	
Berufsjahre in Entwicklungsteams	1	0.5804	$<$ **0.0001**	Mann-Whitney-U-Test
	2	0.7256	**0.0005**	
Anteil negativer Annotationen	1	0.9587	**0.0127**	Mann-Whitney-U-Test
	2	0.9637	0.7300	
Anteil neutraler Annotationen	1	0.9679	**0.0464**	Mann-Whitney-U-Test
	2	0.9565	0.5992	
Anteil positiver Annotationen	1	0.8933	$<$ **0.0001**	Mann-Whitney-U-Test
	2	0.9688	0.8193	

Wie Tabelle 5.7 darstellt, unterschreitet das Ergebnis des Shapiro-Wilk-Testes eines jeden metrischen Merkmals für mindestens die erste der beiden Teilnehmergruppen das Signifikanzniveau $\alpha = 0.05$. Folglich sind die Merkmalswerte aller metrischen Merkmale in Tabelle 5.7 für die erste Teilnehmergruppe nicht normalverteilt. Damit kann unabhängig von weiteren Vorbedingungen nur der Mann-Whitney-U-Test [92] für die Untersuchung der Unterschiede zwischen den Teilnehmergruppen angewendet werden. Da der Mann-Whitney-U-Test nur ein ordinales Skalenniveau der Merkmalswerte voraussetzt, ist es unerheblich, ob diese normalverteilt sind.

5.5.2 Vergleich der Geschlechter- und Altersverteilung

Die Verteilung der männlichen und weiblichen Studienteilnehmer zwischen den beiden Teilnehmergruppen ist in der Kontingenztabelle in Tabelle 5.8 zusammen mit den zugehörigen Ergebnissen des χ^2-Testes dargestellt. Dabei gibt *df* die Anzahl der Freiheitsgrade an, und $P(>\chi^2)$ die Wahrscheinlichkeit des Ergebnisses in Abhängigkeit von χ^2 und *df*.

Tabelle 5.8: Kontingenztabelle der Geschlechterverteilung zwischen den beiden Teilnehmergruppen $G1$ und $G2$ (auf der linken Seite), sowie die zugehörigen Ergebnisse des χ^2-Testes (auf der rechten Seite).

Geschlecht	**G1**	**G2**	$\boldsymbol{\chi^2}$	***df***	$\boldsymbol{P(>\chi^2)}$	**Interpretation**
Männlich	64	13	0.0	1	1.0	Nicht signifikant
Weiblich	14	3				

Die χ^2_C-Statistik nimmt für die Verteilung der männlichen und weiblichen Studienteilnehmer auf die Teilnehmergruppen einen Wert von null an. Die Anzahl der weiblichen und männlichen Studienteilnehmer innerhalb der beiden Teilnehmergruppen entspricht also genau den erwarteten Häufigkeiten. Das durchschnittliche Alter ø und dessen Standardabweichung σ innerhalb beider Teilnehmergruppen ($G = 1$ und $G = 2$), sowie die zugehörigen Ergebnisse des Mann-Whitney-U-Testes sind in Tabelle 5.9 dargestellt.

Tabelle 5.9: Ergebnisse des Mann-Whitney-U-Testes für die Altersangaben der Studienteilnehmer zwischen den Teilnehmergruppen.

Merkmal	$\boldsymbol{G}$	**ø**	$\boldsymbol{\sigma}$	$\boldsymbol{U}$	$\boldsymbol{p}$	**Interpretation**
Alter der	1	27.5769	6.9010	722	0.1527	Nicht signifikant
Teilnehmer	2	24.4000	3.5010			

Wie bei der Verteilung der Geschlechter gibt es auch bei der Altersverteilung keinen statistisch signifikanten Unterschied zwischen den Teilnehmergruppen. Auffallend ist jedoch, dass Studienteilnehmer der ersten Teilnehmergruppe im Durchschnitt mehr als drei Jahre älter sind, als Studienteilnehmer der zweiten Teilnehmergruppe. Die Wahrscheinlichkeit p des Ergebnisses des Mann-Whitney-U-Testes beträgt dabei 15.27 %.

5.5.3 Vergleich des englischsprachigen Verständnisses

Da die zu annotierenden Aussagen den Studienteilnehmer auf Englisch präsentiert worden sind, ist es relevant zu überprüfen, ob es Unterschiede in den Sprachkenntnissen der beiden Teilnehmergruppen gibt. Tabelle 5.10 stellt die Kontingenztabelle über die Verteilung der englischen Muttersprachler und die Muttersprachler anderer Sprachen zwischen den beiden Teilnehmergruppen, sowie die zugehörigen Ergebnisse des χ^2-Testes, dar.

Tabelle 5.10: Kontingenztabelle der Anteile an englischen Muttersprachlern und anderen Muttersprachlern, zwischen den beiden Teilnehmergruppen $G1$ und $G2$ (links), sowie die zugehörigen Ergebnisse des χ^2-Testes (rechts).

Erstsprache	**G1**	**G2**
Englisch	1	1
Andere	76	15

$\boldsymbol{\chi^2}$	***df***	$\boldsymbol{P(>\chi^2)}$	**Interpretation**
0.0872	1	0.7678	Nicht signifikant

Wie die Ergebnisse des χ^2-Testes in Tabelle 5.10 aufzeigen, entspricht diese Verteilung einer Wahrscheinlichkeit von 76.78 %. Damit gibt es keinen statistisch signifikanten Unterschied in der Verteilung der Muttersprachler auf die Teilnehmergruppen. Dies wird durch den Fakt widergespiegelt, dass es in beiden Teilnehmergruppen jeweils nur einen Studienteilnehmer gibt, dessen Muttersprache Englisch ist. Eine weitere Unterteilung der anderen Muttersprachen der Studienteilnehmer ist in der ursprünglichen Umfrage [111] nicht erfolgt. Jedoch wurden die Studienteilnehmer zusätzlich dazu befragt, wie häufig sie auf Englisch kommunizieren. Dies gibt mitunter weitere Aufschlüsse über die englischen Sprachkenntnisse der Studienteilnehmer. Da die Antwortmöglichkeiten zur Häufigkeit der englischsprachigen Kommunikation einer Ordinalskala von *Nie* bis *Täglich* entsprechen (vgl. Tabelle A.1a), wurden die Antworten der Teilnehmer mit dem Mann-Whitney-U-Test untersucht. Dabei wurden die ordinalen Antwortmöglichkeiten der Kommunikationshäufigkeit wurden durch die folgenden Ränge kodiert.

- **1:** Nie
- **2:** Gelegentlich
- **3:** Einmal pro Woche
- **4:** Mehrmals pro Woche
- **5:** Täglich

Obwohl die Ränge der Kommunikationshäufigkeiten ordinalskalierte Merkmalswerte sind, wurde der durchschnittliche Rang ø sowie die Standardabweichung σ beider Teilnehmergruppen berechnet. Selbiges gilt auch für alle weiteren ordinalskalierten Merkmale der Datenbasis. Da der Median bei geringen Unterschieden zwischen den Merkmalswerten identisch ist, kann somit mehr Aufschluss über die Unterschiede zwischen den Teilnehmergruppen gewonnen werden [147]. Diese Durchschnitte sollten dennoch mit Vorsicht betrachtet werden [147]. Die durchschnittlichen Ränge ø der englischsprachigen Kommunikationshäufigkeit beider Teilnehmergruppen, deren Standardabweichungen σ, sowie die zugehörigen Ergebnisse des Mann-Whitney-U-Testes sind in Tabelle 5.11 dargestellt.

Tabelle 5.11: Ergebnisse des Mann-Whitney-U-Testes für die Kommunikationshäufigkeit auf englischer Sprache zwischen den Teilnehmergruppen.

Merkmal	G	ø	σ	U	p	**Interpretation**
Kommunikations-häufigkeit (Engl.)	1	3.4342	1.2684	591	0.8199	Nicht signifikant
	2	3.3333	1.3452			

Tabelle 5.11 zeigt, dass die Studienteilnehmer beider Teilnehmergruppen angeben, durchschnittlich zwischen *Einmal pro Woche* und *Mehrmals pro Woche* auf Englisch zu kommunizieren. Die Wahrscheinlichkeit des Ergebnisses liegt laut dem Mann-Whitney-U-Test bei 81.99 %. Daraus folgt, dass es keinen statistisch signifikanten Unterschied in den englischsprachigen Kommunikationshäufigkeiten zwischen den beiden Teilnehmergruppen gibt.

5.5.4 Vergleich der Berufs- und Programmiererfahrung

Da die berufliche Erfahrung im Bereich der Informatik möglicherweise einen Einfluss auf das Verständnis der zu annotierenden Aussagen hat, werden auch die Unterschiede in der Berufs- und Programmiererfahrung zwischen den Teilnehmergruppen miteinander verglichen.

Vergleich des beruflichen Status

Im Rahmen der Umfrage wurden die Studienteilnehmer zu ihrem beruflichen Status befragt, wobei eine Mehrfachauswahl aus vordefinierten Antwortmöglichkeiten getroffen werden konnte (vgl. Tabelle A.1b). Da keiner der 94 Studienteilnehmer angab, im Ruhestand zu sein, wurde diese Kategorie nicht für die Berechnung der χ^2-Statistik berücksichtigt. Weiterhin wurden die Freitext-Antworten zum beruflichen Status mangels ausreichender Daten nicht analysiert, da nur drei Studienteilnehmer diese Möglichkeit nutzten. Die Ergebnisse der korrigierten χ^2-Statistik χ^2_C sowie die zugehörige Kontingenztabelle zwischen dem beruflichen Status der Teilnehmer und den beiden Teilnehmergruppen $G1$ und $G2$ sind in Tabelle 5.12 dargestellt.

Tabelle 5.12: Kontingenztabelle des beruflichen Status und den beiden Teilnehmergruppen $G1$ und $G2$ (links), sowie die zugehörigen Ergebnisse des χ^2-Testes nach der Rao-Scott-Korrektur χ^2_C (rechts).

Tätigkeit	**G1**	**G2**
Student	53	13
Akademia	15	0
Wirtschaft	23	4
Arbeitslos	1	1

$\boldsymbol{\chi^2_C}$	$\boldsymbol{df_C}$	$\boldsymbol{P(>\chi^2_C)}$	**Interpretation**
7.3098	4	0.1204	Nicht signifikant

Wie Tabelle 5.12 zu entnehmen ist, nimmt χ^2_C einen Wert von ungefähr 7.31 an. Bei einer korrigierten Anzahl der Freiheitsgrade von $df_C = 4$ entspricht dieses Ergebnis einer Wahrscheinlichkeit von 12.04 %. Obwohl das Ergebnis damit nicht statistisch signifikant ist, deutet es auf dennoch auf eine unausgewogene Verteilung des beruflichen Status der Teilnehmer zwischen den Teilnehmergruppen hin. Die Kontingenztabelle auf der linken Seite von Tabelle 5.12 zeigt beispielsweise, dass 15 Studienteilnehmer der ersten Teilnehmergruppe in der Akademia angestellt sind, während dies auf keinen Studienteilnehmer der zweiten Teilnehmergruppe zutrifft. Weiterhin sind 23 Teilnehmer der ersten Teilnehmergruppe in der freien Wirtschaft angestellt, was jedoch nur auf vier Studienteilnehmer der zweiten Teilnehmergruppe zutrifft. Insgesamt scheint es also so, dass für einen deutlich höheren Anteil der ersten Teilnehmergruppe ein Arbeitsverhältnis besteht. Für beide Teilnehmergruppen gibt der Großteil der Studienteilnehmer jedoch (zusätzlich) an, Student zu sein.

Vergleich der Angaben zur Berufs- und Programmiererfahrung

Bezüglich ihrer beruflichen Erfahrung wurden den Studienteilnehmer weitere Fragen gestellt. Zum einen sollten die Studienteilnehmer ihre Programmierkenntnisse und ihre Vertrautheit mit der kollaborativen Softwareentwicklung in Entwicklungsteams selbst einschätzen. Für beide Fragen wurde den Teilnehmern dabei eine fünfstufige Likert-Skala zur Verfügung gestellt (vgl. Tabelle A.1c). Für die Einschätzung der Programmierkenntnisse wurde die erste Stufe der Likert-Skala mit *grundlegend* und die letzte Stufe mit *fortgeschritten* beschriftet. Für die Angabe der Familiarität mit der Arbeit in Entwicklungsteams wurde die erste Stufe der Likert-Skala mit *wenig familiär* und die letzte Stufe mit *sehr familiär* beschriftet. Weiterhin wurden die Studienteilnehmer zur Angabe ihrer Berufsjahre als professioneller Entwickler und als professioneller Entwickler in Entwicklungsteams befragt. Falls die Studienteilnehmer noch keine Erfahrung in einem professionellen Umfeld hatten (z. B. bei Studierenden) konnten dabei auch null Jahre

angegeben werden. Tabelle 5.13 zeigt die durchschnittlichen Merkmalswerte ø, deren Standardabweichungen σ, und die zugehörigen Ergebnisse des Mann-Whitney-U-Testes für die Antworten aller vier zuvor beschriebenen Fragen zwischen beiden Teilnehmergruppen.

Tabelle 5.13: Ergebnisse des Mann-Whitney-U-Testes für die Merkmale der Berufserfahrung zwischen den Teilnehmergruppen ($G = 1$ und $G = 2$).

Merkmal	G	ø	σ	U	p	**Interpretation**
Programmier-kenntnisse	1	3.1923	0.8383	581	0.6391	Nicht signifikant
	2	3.2500	0.6831			
Vertrautheit mit Entwicklungsteams	1	2.7436	1.2632	542	0.6508	Nicht signifikant
	2	2.8667	1.1255			
Berufsjahre als Entwickler	1	3.2564	4.9738	712	0.1713	Nicht signifikant
	2	1.4000	2.0976			
Berufsjahre in Entwicklungsteams	1	2.1688	4.0145	678	0.2617	Nicht signifikant
	2	0.8667	1.3020			

Wie Tabelle 5.13 zu entnehmen ist, schätzen die Studienteilnehmer beider Teilnehmergruppen ihre Programmierkenntnisse leicht überdurchschnittlich, mit 3.19 und 3.25 der möglichen Stufen auf der fünfstufigen Likert-Skala, ein. Zudem schätzen beide Teilnehmergruppen ihre Vertrautheit mit der Arbeit in Entwicklungsteams leicht unterdurchschnittlich ein, mit durchschnittlich 2.74 und 2.87 von fünf möglichen Stufen der Likert-Skala. Die Studienteilnehmer aus Gruppe 2 schätzen sowohl ihre Programmierkenntnisse als auch ihre Vertrautheit mit Entwicklungsteams durchschnittlich höher ein als die Teilnehmer aus Gruppe 1. In beiden Fällen ist der Unterschied zwischen den Teilnehmergruppen nicht signifikant, mit einer Wahrscheinlichkeit von 63.91 % und 65.08 %. Auffällig ist, dass Studienteilnehmer aus Gruppe 1 im Durchschnitt über 22 Monate mehr an Berufsjahren als professioneller Entwickler an Erfahrung haben, als Studienteilnehmer aus Gruppe 2. Zudem haben Studienteilnehmer aus Gruppe 1 im Durchschnitt mehr als 15 Monate mehr an Berufserfahrung in Entwicklungsteams, als Studienteilnehmer aus Gruppe 2. Diese Unterschiede entsprechen nach den Ergebnissen des Mann-Whitney-U-Testes einer Wahrscheinlichkeit von 17.13 % und 26.17 %. Auch wenn die Unterschiede in den Jahren an Berufserfahrung als Entwickler sowohl allein als auch in Entwicklungsteams damit nicht statistisch signifikant sind, weisen sie dennoch eine Auffälligkeit auf. Interessant scheint dabei, dass

die Studienteilnehmer aus Gruppe 2 sowohl ihre Programmierkenntnisse als auch ihre Vertrautheit mit der Arbeit in Entwicklungsteams durchschnittlich selbst höher einschätzen als die Teilnehmer aus Gruppe 1, obwohl letztere tatsächlich mehr praktische Erfahrung haben.

5.5.5 Vergleich der Annotationskriterien

Im Anschluss an die Annotation der präsentierten Aussagen konnten die Studienteilnehmer aus zwei vordefinierten Antwortmöglichkeiten eine Mehrfachauswahl treffen, je nachdem, welche Kriterien sie für die Annotation der Aussagen genutzt haben (vgl. Tabelle A.1d). Zusätzlich konnte eine eigene Freitext-Antwort formuliert werden. Im Folgenden werden die Antworten zwischen der Teilnehmergruppen für die vordefinierten Annotationskriterien und Freitext-Antworten der Teilnehmer verglichen.

Vergleich der vordefinierten Annotationskriterien

Die Kontingenztabelle zwischen den beiden Teilnehmergruppen und den zwei vordefinierten Annotationskriterien, sowie die zugehörigen Ergebnisse des χ^2-Testes nach der Rao-Scott-Korrektur sind in Tabelle 5.14 dargestellt.

Tabelle 5.14: Kontingenztabelle für die vordefinierten Annotationskriterien und die beiden Teilnehmergruppen $G1$ und $G2$ (links), sowie die zugehörigen Ergebnisse des χ^2-Test nach der Rao-Scott-Korrektur χ^2_C (rechts).

Kriterium	**G1**	**G2**	$\boldsymbol{\chi^2_C}$	$\boldsymbol{df_C}$	$\boldsymbol{P(>\chi^2_C)}$	**Interpretation**
Inhalt	35	8	0.0223	2	0.9889	Nicht signifikant
Tonalität	48	9				

Wie Tabelle 5.14 zeigt, sind die Antworten gleichmäßig zwischen den beiden Teilnehmergruppen verteilt, wodurch die korrigierte χ^2-Statistik χ^2_C einen Wert nahe null annimmt. Die zugehörige Wahrscheinlichkeit dieses Ergebnisses entspricht dabei 98.89 % und es gibt somit keinen signifikanten Unterschied in den Angaben der vordefinierten Annotationskriterien zwischen den beiden Teilnehmergruppen.

Freitext-Antworten der ersten Teilnehmergruppe

Von den 78 Studienteilnehmern aus Gruppe 1 haben 13 Teilnehmer eine Freitext-Antwort zur Wahl ihrer Annotationskriterien verfasst. Die Antworten dieser 13 Teilnehmer sind im englischsprachigen Original in Tabelle 5.15 aufgelistet. Zusätzlich sind die Antworten der ersten Teilnehmergruppe in Tabelle 5.15 farblich so hinterlegt, dass ähnlichen Motiven und Inhalten der Antworten gleiche Farben zugeordnet wurden. Diese Zuordnung wurde manuell durchgeführt, und ist daher möglicherweise subjektiv.

Tabelle 5.15: Freitext-Antworten zu den Annotationskriterien von den Studienteilnehmern S_i der ersten Teilnehmergruppe.

S_i	Annotationskriterien (Freitext-Antworten)
S_{17}	Constructiveness (“this solution sucks” < “this solution sucks because ... - let’s do ... instead”)
S_{25}	Based on what i can guess from the context. This fails if i get presented with a single “lol”. “lol :)” on the other hand is positive (assuming not being sarcastic). If a sentence could be positive or negative i choose neutral, knowing that it has both dimensions instead of neutrality.
S_{28}	Politeness, success vs. error
S_{29}	Emotions/ smiley/slangs (LOL etc) used
S_{34}	:-)
S_{36}	Emoticons and if it was mostly descriptive (neutral) or implied having found a solution to a problem (positive)
S_{38}	:-) or LOLs
S_{56}	I based my decision on my first emotional response to the sentences. Most of them didn’t lead to much of a response because I didn’t know enough about their context. Some that sounded to come from people with a good understanding of English who used wrong grammer despite of this lead to a negative response.
S_{57}	Emoticons
S_{58}	Smileys
S_{60}	Partly based on my own expertiese and how I would feel /react if I read some of the questions. Some specialities were out of my space of knowledge: for those I responed Neutral too
S_{77}	Sometimes the message may be neutral but the context (i. e. talking abt erros) may be negative for development
S_{92}	It can be complicated without context. Basically, I tried to guess the tone of the message, reflecting how the person typing the comment or reading it might feel , as opposed to just communicating a technical fact. Using this definition, the two examples above (“I don’t like this phone”) and “Not again, Mike” in fact have the same effect.

Wie Tabelle 5.15 zu entnehmen ist, gibt es drei wiederkehrende Motive in den Freitext-Antworten der Studienteilnehmer aus Gruppe 1. So erläutern drei Teilnehmer, dass sie die Annotation anhand von inhaltlichen Erfolgen, wie Problemlösungen, oder Berichten von Fehlern in der Software

durchgeführt haben (vgl.). Der Teilnehmer S_{77} führt weiterhin aus, dass Berichte über Fehler in der Software möglicherweise schlecht für die Entwicklung des betreffenden Projektes sein könnten (und damit folglich auch die Sentiment-Polarität der Aussage selbst *negativ* wahrgenommen wird). Drei weitere Studienteilnehmer der ersten Teilnehmergruppe geben an, dass sie die Vergabe der Annotationen anhand ihrer initialen emotionalen Resonanz vorgenommen haben (vgl.). Mit sieben von 13 Teilnehmern, gibt die Mehrheit der Studienteilnehmer aus Gruppe 1 in Tabelle 5.15 an, dass sie Smileys bzw. Emoticons und Jargon (engl. *Slang*) als Annotationskriterien genutzt haben (vgl.). Der Studienteilnehmer S_{25} fügt beispielsweise hinzu, dass „*lol :)*“ (welches im englischen Jargon für „*Laughing Out Loud*“ steht, also in etwa soviel wie „*Lautes Lachen*“ bedeutet) von ihm als *positiv* wahrgenommen wurde, „*lol*“ ohne einen lächelnden Smiley jedoch nicht.

Freitext-Antworten der zweiten Teilnehmergruppe

Von den 16 Studienteilnehmer der zweiten Teilnehmergruppe haben nur zwei Teilnehmer eine Freitext-Antwort abgegeben, um ihre Annotationskriterien zu erläutern. Die Antworten der beiden Studienteilnehmer sind in Tabelle 5.16 dargestellt. Dabei sind ähnliche Inhalte und Motive innerhalb der Teilnehmergruppe wieder durch gleiche Farben hervorgehoben.

Tabelle 5.16: Freitext-Antworten zu den Annotationskriterien von den Studienteilnehmern S_i der zweiten Teilnehmergruppe.

S_i	Annotationskriterien (Freitext-Antwort)
S_1	Being the right amount of verbose. Not too much, but also asking incomplete questions. Also comments which are helpful in explaining design choices are appreciated and are also lightyears better than unncessary/unprofessional ones.
S_{33}	How informative the post was, if the post was written clear and understandable, also if the post had meaning. “lol” e.g. is ok in a casual environment but i would not want to have to check my notifications for a useless “lol”

Wie Tabelle 5.16 darstellt, überschneiden sich die Antworten der beiden Teilnehmer aus der zweiten Teilnehmergruppe ebenfalls in drei Motiven. Beide Teilnehmer S_1 und S_{33} gaben an, dass sie die Aussagen nach Verständlichkeit (vgl.) und danach, wie hilfreich, informativ oder bedeutsam ihnen die Aussagen erschienen (vgl.) annotierten. Anschließend gehen beide Teilnehmer darauf ein, dass sie unnötige und unprofessionelle Aussagen als *negativ* wahrnehmen (vgl.). Der Studienteilnehmer S_{33} elaboriert

weiterhin, dass eine Antwort, die nur aus „*lol*“ besteht, zwar in einer zwanglosen Umgebung in Ordnung ist, er jedoch darüber verärgert wäre, seine Benachrichtigungen in einem professionellen Umfeld für eine solche Nachricht zu überprüfen. Diese unterschiedlichen Ansichten zwischen den Teilnehmergruppen decken sich mit den Beobachtungen aus Tabelle 5.3. Dort wiesen die Aussagen „*lol :)*“ und „*Lol.*“ enorme Unterschiede in der Verteilung der Sentiment-Polaritäten zwischen den Teilnehmergruppen auf.

5.5.6 Vergleich der Wahrnehmung

Um die Unterschiede in der Wahrnehmung zwischen den Studienteilnehmer auf statistische Signifikanz zu überprüfen, wurden in Kapitel 4.3.5 die 96 untergeordneten Nullhypothese $\text{H2}(V_1)_0$ bis $\text{H2}(V_{96})_0$, sowie die übergeordnete Nullhypothese H2_0 aufgestellt. Um die Unterschiede in der Wahrnehmung zu charakterisieren, wurden zudem die drei untergeordneten Nullhypothesen $\text{H3}(\textit{negativ})_0$, $\text{H3}(\textit{neutral})_0$, und $\text{H3}(\textit{positiv})_0$, sowie die übergeordnete Nullhypothese H3_0 aufgestellt. Im Folgenden werden die Ergebnisse der Hypothesenprüfung für diese Nullhypothesen präsentiert.

Hypothesenprüfung von H2_0 und $\text{H2}(V_i)_0$

Die Nullhypothese H2_0 besagt, dass es keinen Unterschied zwischen den Annotationen der Teilnehmergruppen gibt. Um dies zu überprüfen, wurden die 96 untergeordneten Nullhypothesen $\text{H2}(V_1)_0$ bis $\text{H2}(V_{96})_0$ getestet, indem jede Aussage der Datenbasis mit dem Mann-Whitney-U-Test auf Unterschiede in den Annotationen der Teilnehmergruppen geprüft wurde. Das Signifikanzniveau für die untergeordneten Nullhypothesen $\text{H2}(V_i)_0$ wurde dabei durch die Bonferroni-Korrektur [127] auf $\alpha = 0.05/96 \approx 0.000521$ angepasst. Tabelle 5.17 zeigt einen relevanten Ausschnitt der Ergebnisse des Mann-Whitney-U-Testes für die 96 Aussagen der Datenbasis dieser Arbeit.

Tabelle 5.17: Ergebnisse des Mann-Whitney-U-Testes für die Annotationen der Aussagen V_1 bis V_{96} geordnet nach aufsteigender Wahrscheinlichkeit p.

Nr.	$\boldsymbol{V_i}$	$\boldsymbol{U}$	$\boldsymbol{p}$	**Interpretation**
1	V_{78}	1162.0	< 0.000001	**Signifikant**
⋮			⋮	⋮
37	V_{66}	348.0	0.000428	**Signifikant**
38	V_{52}	333.0	0.000599	Nicht signifikant
⋮			⋮	⋮
96	V_{86}	619.5	0.965731	Nicht signifikant

Wie Tabelle 5.17 zu entnehmen ist, unterschreitet die Wahrscheinlichkeit p der Ergebnisse des Mann-Whitney-U-Testes für die Aussagen der Datenbasis das Signifikanzniveau $\alpha = 0.000521$ in 37 von 96 Fällen (38.54 %). Für diese 37 Aussagen gibt es also einen statistisch signifikanten Unterschied in der Wahrnehmung zwischen den beiden Teilnehmergruppen. Damit kann neben den 37 untergeordneten Nullhypothesen $\text{H2}(V_i)_0$ dieser 37 Aussagen V_i also auch die übergeordnete Nullhypothese H2_0 abgelehnt werden.

Hypothesenprüfung von H3_0 und $\text{H3}(P)_0$

Um besser charakterisieren zu können, wie sich die Wahrnehmung der beiden Teilnehmergruppen im Allgemeinen unterscheidet, wurde die übergeordnete Nullhypothese H3_0 aufgestellt, welche besagt, dass es keinen Unterschied in den durchschnittlichen relativen Anteilen der Sentiment-Polaritäten *negativ*, *neutral* und *positiv* zwischen Studienteilnehmern der beiden Teilnehmergruppen gibt (vgl. Kapitel 4.3.5). Zur Prüfung von H3_0 wurden für jeden Studienteilnehmer die relativen Anteile der *negativen*, *neutralen* und *positiven* Annotationen berechnet und zwischen den Teilnehmergruppen mit dem Mann-Whitney-U-Test verglichen. Der Mann-Whitney-U-Test wurde dabei dreimal, für die Prüfung der drei untergeordneten Nullhypothesen $\text{H3}(\textit{negativ})_0$, $\text{H3}(\textit{neutral})_0$ und $\text{H3}(\textit{positiv})_0$ durchgeführt. Das Signifikanzniveau für die drei untergeordneten Nullhypothesen $\text{H3}(P)_0$, $P \in \{\textit{negativ}, \textit{neutral}, \textit{positiv}\}$ wurde durch die Bonferroni-Korrektur auf $\alpha = 0.05/3 = 0.016\overline{6}$ angepasst. Die durchschnittlichen Anteile der Sentiment-Polaritäten ø, deren Standardabweichungen σ, und die zugehörigen Ergebnisse des Mann-Whitney-U-Testes zwischen den Teilnehmergruppen sind in Tabelle 5.18 dargestellt.

Tabelle 5.18: Ergebnisse des Mann-Whitney-U-Testes für die durchschnittlichen Anteile der Sentiment-Polaritäten zwischen den Studienteilnehmern der beiden Teilnehmergruppen ($G = 1$ und $G = 2$).

Anteil in %	G	ø	σ	U	p	**Interpretation**
Negativ	1	21.7682	9.7927	583	0.6832	Nicht signifikant
	2	22.0703	11.1494			
Neutral	1	52.2169	13.1476	917	0.0032	**Signifikant**
	2	41.6016	11.5085			
Positiv	1	26.0150	9.6352	327	0.0028	**Signifikant**
	2	36.3281	13.2193			

Wie Tabelle 5.18 zeigt, nehmen die Studienteilnehmer aus Gruppe 1 die Aussagen der Datenbasis durchschnittlich zu 21.77 % als *negativ* wahr, während die Teilnehmer aus Gruppe 2 durchschnittlich 22.07 % der Aussagen als *negativ* wahrnehmen. Zwischen beiden Teilnehmergruppen liegt also nur eine Differenz von 0.3 Prozent-Punkten. Auch wenn die Teilnehmer der zweiten Teilnehmergruppe damit im Durchschnitt häufiger Aussagen als *negativ* wahrnehmen, ist dieser Unterschied nicht signifikant. Das Ergebnis des Mann-Whitney-U-Test entspricht einer Wahrscheinlichkeit von 68.32 %. Anders verhält es sich für die Anteile der *neutralen* Annotationen zwischen den Teilnehmergruppen. Die Studienteilnehmer der ersten Teilnehmergruppe nehmen durchschnittlich 52.22 % der 96 Aussagen als *neutral* wahr, während durchschnittlich nur 41.60 % der Aussagen von der zweiten Teilnehmergruppe als *neutral* wahrgenommen werden. Diese Differenz entspricht also ungefähr 10.62 Prozent-Punkten zwischen den Teilnehmergruppen für den Anteil der *neutral* wahrgenommenen Aussagen. Das Ergebnis des Mann-Whitney-U-Test ist dabei mit einer Wahrscheinlichkeit von nur 0.32 % unter dem Signifikanzniveau $\alpha = 0.016\overline{6}$ statistisch signifikant. Letztlich werden von Studienteilnehmern aus Gruppe 1 durchschnittlich 26.02 % der Aussagen in der Datenbasis als *positiv* wahrgenommen, im Vergleich zu 36.33 % der Aussagen für die zweite Teilnehmergruppe. Das entspricht einer Differenz von 10.31 Prozent-Punkten zwischen den beiden Teilnehmergruppen. Das Ergebnis des Mann-Whitney-U-Testes ist mit einer Wahrscheinlichkeit von nur 0.28 % statistisch signifikant. Da die beiden untergeordneten Nullhypothesen H3$(\textit{neutral})_0$ und H3$(\textit{positiv})_0$ folglich abgelehnt werden können, kann auch die übergeordnete Nullhypothese H3_0 abgelehnt werden.

Beobachtung 5.5: Im statistischen Vergleich der Teilnehmergruppen konnten keine signifikanten Unterschiede in den demografischen Merkmalen, dem englischsprachigen Verständnis, der Berufserfahrung und Programmierkenntnisse, oder den vordefinierten Annotationskriterien zwischen den Teilnehmergruppen ausgemacht werden. Die Freitext-Antworten zu den Annotationskriterien unterscheiden sich jedoch zwischen den Teilnehmergruppen, wobei Studienteilnehmer aus Gruppe 2 angeben, kurze saloppe Aussagen in einem beruflichen Umfeld als *negativ* wahrzunehmen. Teilnehmer aus Gruppe 1 geben hingehen teilweise an, dieselben kurzen saloppen Aussagen als *positiv* wahrzunehmen. Insgesamt unterscheidet sich die Wahrnehmung zwischen den Teilnehmergruppen für 37 der 96 Aussagen (38.54 %) in der Datenbasis signifikant. Zudem weisen die durchschnittlichen Anteile der Sentiment-Polaritäten aller 96 Aussagen zwischen den Teilnehmergruppen signifikante Unterschiede auf: Studienteilnehmer aus Gruppe 2 nehmen durchschnittlich über 10 % weniger Aussagen als *neutral*, dafür aber über 10 % mehr Aussagen als *positiv* wahr, im Vergleich zu den Teilnehmern aus Gruppe 1.

Kapitel 6

Diskussion

In diesem Kapitel werden die Ergebnisse aus Kapitel 5 dieser Arbeit interpretiert. Dabei wird auf die Relevanz der Ergebnisse für die zukünftige Forschung im Bereich der Stimmungsanalyse im Software Engineering eingegangen und über die möglichen Ursachen für die unterschiedlichen Wahrnehmungen der Studienteilnehmer diskutiert. Abschließend wird dieses Kapitel mit einer Diskussion über die Einschränkungen, welchen die Ergebnisse dieser Arbeit unterliegen, beendet.

6.1 Interpretation der Ergebnisse

Die Ergebnisse dieser Arbeit liefern wichtige Erkenntnisse für die Anwendung der Stimmungsanalyse in Softwareprojekten. Durch die hierarchische Clusteranalyse konnten zwei Teilnehmergruppen in der Datenbasis identifiziert werden. Die Wahrnehmungen der Sentiment-Polaritäten der Studienteilnehmer korrelieren innerhalb beider Teilnehmergruppen stark positiv miteinander. Zwischen den Teilnehmergruppen unterscheidet sich die Wahrnehmung jedoch signifikant für viele Aussagen der Datenbasis. Zu diesen Aussagen gehören insbesondere kurze saloppe Aussagen, wie „*lol :)*“, die vom Großteil der zweiten Teilnehmergruppe als *negativ* und vom Großteil der ersten Teilnehmergruppe als *positiv* wahrgenommen werden. Diese Wahrnehmung wurde durch die qualitativen Freitext-Antworten der Teilnehmer zu ihren Annotationskriterien bestätigt (vgl. Kapitel 5.5). Zudem wurden statistisch signifikante Unterschiede in der Wahrnehmung zwischen den Teilnehmergruppen für die Sentiment-Polaritäten *neutral* und *positiv* über alle annotierten Aussagen hinweg beobachtet. Die stark unterschiedlichen Wahrnehmungen von Aussagen können zu Missverständnissen in der Kommunikation zwischen den Entwicklern innerhalb von Softwareprojekten führen. Diese Ergebnisse liefern wichtige Anhaltspunkte, welche für die zukünftige Forschung im Bereich der Stimmungsanalyse in Softwareprojekten von Relevanz sind.

6.1.1 Implikationen für zukünftige Forschung

Wie die Ergebnisse dieser Arbeit aufzeigen, gibt es Gruppen von Entwicklern, die Emotionen in Aussagen, aus der Domäne der Softwareentwicklung, signifikant unterschiedlich wahrnehmen. Um die eingangs beschriebene (vgl. Kapitel 1.1) industrielle Anwendung der Stimmungsanalyse in Softwareprojekten zu ermöglichen, müssen die Stimmungsanalysetools daher kalibriert werden, um die Wahrnehmung der einzelnen Entwickler oder Entwicklungsteams widerzuspiegeln. Da die Studienteilnehmer bereits anhand ihrer Annotationen von nur fünf verschiedenen Aussagen korrekt in ihre jeweiligen Teilnehmergruppen zugeordnet werden konnten, sollte auch eine solche Kalibrierung nicht mit einem zu hohem Aufwand verbunden sein. Jedoch müssen dafür auch die Grundgedanken hinter den Datensätzen [84, 107] und Stimmungsanalysetools [64] überdacht werden. Datensätze mit einer einzelnen Sentiment-Polarität pro Aussage sind nicht ausreichend, wenn unterschiedliche Wahrnehmungen berücksichtigt werden sollen. Die bisherigen Ansätze sind daher nicht für eine industrielle Anwendung geeignet, da sie allesamt die unterschiedlichen Wahrnehmungen von verschiedenen Entwicklern vernachlässigen. So würde ein Stimmungsanalysetool für die Aussagen „*lol :)*“ aufgrund der positiven Assoziation der Bedeutung (*Laughing Out Loud*), sowie des lächelnden Emoticons, eine *positive* Sentiment-Polarität ausgeben (vgl. Sentimentlexikon von SentiStrength-SE [64]). Somit wäre die Ausgabe eines Stimmungsanalysetools nicht repräsentativ für die Wahrnehmung eines Entwicklers der zweiten Teilnehmergruppe. In der Forschung muss daher ein Umdenken stattfinden: Es gibt keine Einheitslösung, welche die Wahrnehmung der Stimmung von jedem einzelnen Entwickler widerspiegeln kann. Da einzelne Aussagen von verschiedenen Entwicklern sowohl *negativ* als auch *positiv* wahrgenommen werden können, müssen Stimmungsanalysetools in der Lage sein, auch beide Wahrnehmungen widerzuspiegeln.

> Es ist notwendig, die unterschiedlichen Wahrnehmungen von Entwicklern zu berücksichtigen, wenn das Ziel eine industrielle Anwendung der Stimmungsanalyse in Softwareprojekten ist.

Jedoch konnten keine statistisch signifikanten Unterschiede in den demografischen Merkmalen zwischen den Teilnehmergruppen festgestellt werden. Ohne die Kenntnis von mindestens einer kleinen Menge von Annotationen eines Entwicklers ist es nicht möglich, diesen einer der beiden Teilnehmergruppen mit unterschiedlicher Wahrnehmung zuzuordnen. Für die Kalibrierung eines Stimmungsanalysetools sind daher manuelle Annotationen der Entwickler des betrachteten Softwareprojektes notwendig. Eine Identifikation relevanter Merkmale für die Wahrnehmung der Stimmung wäre jedoch ein notwendiger nächster Schritt, um eine Kalibrierung von Stimmungsanalysetools auf Entwicklungsteams durchführen zu können, ohne dass die betreffenden Teammitglieder selbst Aussagen annotieren müssen.

6.1.2 Ursachen der unterschiedlichen Wahrnehmungen

Über die Gründe der unterschiedlichen Wahrnehmungen zwischen den Teilnehmergruppen kann nur spekuliert werden, da die betrachteten demografischen Merkmale keine statistisch signifikanten Unterschiede aufwiesen. Eine mögliche Ursache könnte in unterschiedlichen Ausprägungen von Persönlichkeitsmerkmalen zwischen den Studienteilnehmern liegen. Diese werden in der Psychologie durch das Fünf-Faktoren-Modell (engl. *Big Five*) modelliert und umfassen *Offenheit*, *Gewissenhaftigkeit*, *Extraversion*, *Verträglichkeit* und *Neurotizismus* [105]. Ausgeprägter *Neurotizismus* geht dabei beispielsweise mit erhöhter Reizbarkeit, Unzufriedenheit, Stressexposition, und einer allgemeinen negativeren Affektlage einher [105]. Es ist zumindest eine naheliegende Vermutung, dass sich diese Aspekte auch auf die Wahrnehmung der Stimmung von Entwicklern in Softwareprojekten auswirken. Die Anwendung von psychometrischen Analysen ist zudem bereits Gegenstand der Forschung im Software Engineering. Feldt et al. [30] untersuchten so die Zusammenhänge zwischen den fünf Persönlichkeitsmerkmalen und der Haltung von Entwicklern zu verschiedenen Prozessen, Aktivitäten und Tools des Software Engineerings. Die Ausprägungen der Persönlichkeitsmerkmale der Entwickler wurden über den standardisierten IPIP-Test [39] gemessen [30]. Dabei wurde insbesondere ein Zusammenhang zwischen der *Gewissenhaftigkeit* und dem Verlangen nach einer Änderung des genutzten Softwareprozesses der befragten Entwickler nachgewiesen [30]. Feldt et al. [30] empfehlen daher Forschenden im Software Engineering mehr psychometrische Analysen von Studienteilnehmern einzubinden. So kann ausgeschlossen werden, dass anstelle von tatsächlichen Präferenzen unbewusst nur die unterschiedlichen Ausprägungen der Persönlichkeitsmerkmale der Entwickler gemessen werden [30]. Im Rahmen einer Folgestudie zu den Ergebnissen dieser Arbeit wäre es daher sinnvoll, das Erhebungsdesign von Obaidi et al. [111] um einen standardisierten Test zur Erhebung der Persönlichkeitsmerkmale (z. B. NEO-PI-R [15] oder IPIP [39]), zu erweitern. Anschließend könnte durch eine Korrelationsanalyse gemessen werden, ob sich Zusammenhänge zwischen den Annotationen und Persönlichkeitsmerkmalen der Studienteilnehmer identifizieren lassen.

6.2 Einschränkungen der Ergebnisse

Die Ergebnisse dieser Arbeit sind auf die ausgewählte Population von 94 Studienteilnehmern beschränkt, und nicht die Grundgesamtheit aller Entwickler generalisierbar. Weiterhin unterliegen die Ergebnisse allen Einschränkungen, welche auch im Abschnitt „*Threats to Validity*“ der Studie von Herrmann et al. [54] aufgeführt sind. Nachfolgend werden die wichtigsten dieser Einschränkungen noch einmal zusammengefasst. Zusätzlich werden die spezifischen Einschränkungen, der im Rahmen dieser Arbeit angewendeten Forschungsmethoden aus Kapitel 4 ergänzt.

6.2.1 Demografie der Studienteilnehmer

Da die ursprüngliche Umfrage [111] von der Leibniz Universität Hannover aus veröffentlicht wurde, besteht die betrachtete Population fast ausschließlich aus Teilnehmern, deren Erstsprache nicht Englisch ist [54]. Diese Einschränkung wird jedoch dadurch abgeschwächt, dass die Studienteilnehmer beider Teilnehmergruppen angaben, durchschnittlich einmal pro Woche bis mehrmals pro Woche auf Englisch zu kommunizieren. Auch gab es zwischen den Angaben beider Gruppen keine statistisch signifikanten Unterschiede zwischen den Häufigkeiten der englischsprachigen Kommunikation. Folglich kann davon ausgegangen werden, dass die sprachlichen Kenntnisse die Ergebnisse der hierarchischen Clusteranalyse sowie dem allgemeinen Verständnis der Studienteilnehmer bezüglich der ihnen präsentierten Aussagen nicht beträchtlich beeinflussten [54]. Zusätzlich gab der Großteil der Studienteilnehmer beider Teilnehmergruppen an, noch zu studieren. Damit handelt es sich bei der betrachteten Population also zum Großteil nicht um professionelle Entwickler. Dennoch kann davon ausgegangen werden, dass Studenten der Informatik repräsentative Personen für die Rolle eines Entwicklers sind [54]. Außerdem werden selbst „*Goldstandard*"-Datensätze für die Stimmungsanalyse im Software Engineering [109] mit der Unterstützung von Informatik-Studierenden annotiert.

6.2.2 Fehlender Kontext der präsentierten Aussagen

Eine weitere Einschränkung ergibt sich dadurch, dass den Studienteilnehmern nur einzelne zufällig selektierte Aussagen präsentiert wurden, ohne einen zugehörigen Kontext oder gar einen vollständigen Konversationsverlauf [54]. Dies hatte zur Folge, dass einige Studienteilnehmer angaben, dass sie versuchten, den Kontext zu erraten, um sich für die Annotation einer Sentiment-Polarität zu entscheiden [54]. Dies kann die erhobenen Annotationen beeinflussen, da Aussagen abhängig vom Kontext unterschiedlich wahrgenommen werden können [54]. Allerdings könnte dies auch dazu geführt haben, dass die initiale emotionale Reaktion der Teilnehmer auf die Aussagen erfasst wurde [54]. Dies würde sich mit der Beobachtung von Murgia et al. [101] decken, deren Ergebnisse ergaben, dass sich Teilnehmer unsicherer bei der Annotation einer Sentiment-Polarität sind, wenn ihnen zusätzlich der zugehörige Kontext der Aussagen präsentiert wird (vgl. Kapitel 3). Eine denkbare Erklärung dafür wäre beispielsweise, dass die initiale emotionale Reaktion eher die subjektive Wahrnehmung eines Teilnehmers widerspiegelt, während die Teilnehmer stattdessen bei der Präsentation von mehreren zugehörigen Informationen durch den Kontext länger nachdenken und versuchen, zu einem rationalen Schluss zu gelangen. In diesem Fall wäre eine Erhebung der Annotationen ohne den zugehörigen Kontext der Aussagen also von Vorteil.

6.2.3 Fehlende Annotation der Datenbasis

Eine Einschränkung der Ergebnisse dieser Arbeit ist, dass ein durchschnittlicher Anteil von 25.92 % der Annotation der Studienteilnehmer im Rohdatensatz [111] fehlten. Das Fehlen dieser Annotationen kann zu einer Verzerrung der Ergebnisse führen [70]. Um dieser Verzerrung entgegenzuwirken, wurden Empfehlungen der Literatur zu den Mechanismen fehlender Daten und den empfohlenen Verfahrensweisen befolgt [70]. Da nicht sichergestellt werden konnte, welcher Mechanismus für das Fehlen der Annotationen verantwortlich ist (*Missing Completely at Random* oder *Missing at Random* [125]), wäre eine Analyse ausschließlich anhand der vollständigen Beobachtungen (*Complete Case Analysis*) mit einer Verzerrung der Ergebnisse einhergegangen [70]. Daher wurden die fehlenden Annotationen der Datenbasis mit dem Verfahren MICE-RF [138, 157] imputiert. Dieses Imputationsverfahren hat den Vorteil, dass die natürliche Variabilität der fehlenden Daten in Bezug auf die unvollständigen Beobachtungen erhalten bleibt, und dass die Unsicherheit aufgrund der fehlenden Daten berücksichtigt wird, was zu einer gültigen statistischen Inferenz führt [70]. Auch wenn die Verzerrung der Datenbasis durch die fehlenden Annotationen somit abgeschwächt werden konnte (vgl. Kapitel 4.2.2), wäre es vorzuziehen, wenn von vorneherein nur vollständige Beobachtungen vorgelegen hätten. Für eine mögliche Folgestudie gilt daher sicherzustellen, dass alle Beobachtungen vollständig sind.

6.2.4 Vorbedingungen der linearen Diskriminanzanalyse

Im Rahmen dieser Arbeit wurden die lineare Diskriminanzanalyse auf die Datenbasis angewendet, obwohl die Vorbedingungen der multivariaten Normalität und Homoskedastizität nicht durch die Datenbasis erfüllt werden konnten. Einige Fachartikel weisen jedoch darauf hin, dass die lineare Diskriminanzanalyse robust gegen die Verletzung ihrer Vorbedingungen ist und dennoch verwendbare Ergebnisse erzielt [60]. Mitunter erreicht die lineare Diskriminanzanalyse sogar unter der Verwendung von dichotomen Variablen eine gute Separation der Klassen [73]. Li et al. [83] und Duda et al. [26] erzielten mit der linearen Diskriminanzanalyse gute Ergebnisse der Klassifikation in der Gesichts- und Objekt-Erkennung, obwohl die Voraussetzungen der multivariaten Normalität und Homoskedastizität verletzt wurden. Hastie et al. [50] verweisen darauf, dass die Optimierungsfunktion der linearen Diskriminanzanalyse für eine Klassifikation von lediglich zwei Klassen keine Normalverteilung der Variablen annimmt und somit dennoch anwendbar ist. Dies würde auf die Anwendung der linearen Diskriminanzanalyse im Rahmen dieser Arbeit ebenfalls zutreffen, da nur die beiden Teilnehmergruppen (also zwei Klassen) separiert wurden. Weiterhin hat sich gezeigt, dass die fünf Aussagen, welche mithilfe der linearen Diskriminanzanalyse selektiert wurden, tatsächlich in der Lage waren, die Studienteilnehmer mittels logistischer

Regressionsanalyse korrekt ihren jeweiligen Teilnehmergruppen zuzuordnen. Daraus kann geschlossen, dass die lineare Diskriminanzanalyse auch in dieser Arbeit robust gegen die Verletzung ihrer Vorbedingungen war, und sinnvolle Aussagen selektiert wurden. Es ist jedoch unbekannt, ob dies bereits die bestmögliche Lösung oder nur eine gute Lösung darstellt. Möglicherweise ist also eine Separation der Teilnehmergruppen auch mit weniger als fünf der 96 Aussagen möglich, welche aber aufgrund der verletzten Vorbedingungen der lineare Diskriminanzanalyse nicht ermittelt werden konnte. Hier gibt es eine offene Möglichkeit auf den Ergebnissen aufzubauen und andere Merkmalsselektionsverfahren zu nutzen, um zu überprüfen, ob eine weitere Reduktion der Anzahl an Prädiktorvariablen, oder eine bessere Separation der Teilnehmer mit fünf anderen Prädiktorvariablen, möglich ist.

6.2.5 Glaubwürdigkeit der Ergebnisse

Die Ergebnisse dieser Arbeit bauen hauptsächlich auf den Ergebnissen der Clusteranalyse auf. Die Wahl des Distanzmaßes, dem Fusionierungsalgorithmus, sowie der hierarchischen Clusteranalyse selbst wurden nach bester Kenntnis im Sinne der Zielsetzung dieser Arbeit vorgenommen. Die Anzahl der Cluster wurde objektiv durch den Silhouettenkoeffizienten bestimmt. Durch die Bonferroni-Korrektur [127] wurde das Signifikanzniveau für die Korrelationen zwischen den Studienteilnehmern, sowie die Unterschiede in den Annotationen der einzelnen Aussagen und der durchschnittlich vergebenen Sentiment-Polaritäten zwischen den Teilnehmergruppen angepasst. Somit konnte fälschlicherweise statistisch signifikanten Ergebnissen vorgebeugt werden. Dennoch konnten viele statistisch signifikante Korrelation zwischen den Annotationen der Teilnehmer innerhalb derselben Teilnehmergruppen beobachtet werden. Ebenso wurden viele statistisch signifikante Unterschiede zwischen den Annotationen beider Teilnehmergruppen für einzelne Aussagen, sowie allgemeine Unterschiede zwischen den Anteilen der Sentiment-Polaritäten *neutral* und *positiv* beobachtet. Daher kann mit hoher Sicherheit davon ausgegangen werden, dass diese Teilnehmergruppen so tatsächlich in der Datenbasis vorliegen und unterschiedliche Wahrnehmungen aufweisen. Es ist jedoch nicht bekannt, ob sich diese Teilnehmergruppen bei der Betrachtung einer deutlich größeren Population $n \gg 94$ in weitere Untergruppen aufteilen würden. Es gibt keinerlei anerkannte Möglichkeiten, die Signifikanz der Ergebnisse von Clusterverfahren zu validieren [72]. Stattdessen ist es gängige Praxis, die Ergebnisse anhand von Dimensionsreduktionsverfahren wie der Hauptkomponentenanalyse zu visualisieren [72], so wie es auch im Rahmen dieser Arbeit durchgeführt worden ist.

Kapitel 7

Zusammenfassung und Ausblick

Um diese Arbeit abzuschließen, werden die wichtigsten Resultate in diesem Kapitel zusammengefasst. Außerdem wird ein Ausblick auf die zukünftige Forschung im Bereich der Stimmungsanalyse in Softwareprojekten gegeben, welche durch diese Arbeit angeregt werden soll.

7.1 Zusammenfassung

Motiviert durch die Fortentwicklung der Stimmungsanalyse für eine industrielle Anwendung in Softwareprojekten wurden im Rahmen dieser Arbeit Unterschiede in der Wahrnehmungen der Stimmung, bezüglich Aussagen aus der Domäne der kollaborativen Softwareentwicklung, von potenziellen Mitgliedern eines Entwicklungsteams untersucht. Dafür wurden die Annotationen der Sentiment-Polaritäten *negativ*, *neutral* und *positiv* zu jeweils 96 Aussagen der Plattformen *GitHub* und *Stack Overflow* von 94 verschiedenen Informatikern betrachtet. Anhand einer Korrelationsanalyse zwischen den Annotationen der Studienteilnehmer konnten hoch korrelative Strukturen innerhalb der Datenbasis aufgedeckt werden. Mit der folgenden hierarchischen Clusteranalyse auf Basis der Annotationen wurden die Studienteilnehmer in zwei Gruppen von 78 und 16 Studienteilnehmern geclustert. Die Wahrnehmungen der Studienteilnehmer dieser beiden Teilnehmergruppen korrelieren innerhalb ihrer eigenen Gruppe stark positiv miteinander. Zwischen den Teilnehmergruppen gibt es stattdessen entweder schwach positive, negative oder gar keine Korrelationen. Dies liegt daran, dass es statistisch signifikante Unterschiede in der Wahrnehmung der beiden Teilnehmergruppen gibt: Die zweite Teilnehmergruppe mit 16 Teilnehmern nimmt durchschnittlich 10 % mehr Aussagen als *positiv*, und dafür 10 % weniger Aussagen als *negativ* wahr. Indes gibt es wiederum einzelne Aussagen, welche von der zweiten Teilnehmergruppe als *negativ*

und von der ersten Teilnehmergruppe als *positiv* wahrgenommen werden. Dazu gehören z. B. kurze saloppe Aussagen, welche die Teilnehmer der zweiten Gruppe als *negativ* wahrnehmen, während die Teilnehmer der ersten Gruppe diese *positiv* wahrnehmen. In den demografischen Merkmalen der Studienteilnehmer gibt es jedoch keine statistisch signifikanten Unterschiede zwischen den Teilnehmergruppen, sodass weitere Untersuchungen erfolgen müssen, um aufzuklären, was ursächlich für diese Unterschiede in der Wahrnehmung ist. Offenbar muss aber ein Umdenken in der Forschung zur Anwendung der Stimmungsanalyse im Software Engineering stattfinden: Die deutlichen Unterschiede in der Wahrnehmung der Stimmung von einzelnen Entwicklern sollten nicht weiterhin vernachlässigt werden.

7.2 Ausblick

Die Forschung im Bereich der Stimmungsanalyse in Softwareprojekten muss einen Perspektivwechsel auf die Ebenen der einzelnen Entwickler vollziehen, d. h. dass Stimmungsanalysetools in der Lage sein müssen, ihre Ausgabe an die subjektive Wahrnehmung unterschiedlicher Entwickler anzupassen. Dafür ist eine Kalibrierung der Stimmungsanalysetools nötig. Zunächst müssen die relevanten Merkmale für eine solche Kalibrierung bestimmt werden, damit die Ergebnisse dieser Arbeit vollständig genutzt werden können, um die Stimmungsanalyse in industriellen Softwareprojekten erfolgreich anwenden zu können. Eine mögliche Erklärung der unterschiedlichen Wahrnehmung von Entwicklern sind unterschiedliche ausgeprägte Persönlichkeitsmerkmale. Diese Persönlichkeitsmerkmale können durch das Fünf-Faktoren-Modell modelliert [105], und durch standardisierte Testverfahren gemessen werden [15]. Eine sinnvolle Ergänzung einer Folgestudie wäre also neben der Wahrnehmung der Emotionen auch die Persönlichkeitsmerkmale der Studienteilnehmer zu erheben. Kann der Zusammenhang zwischen den Persönlichkeitsmerkmalen eines Entwicklers und dessen Wahrnehmung der Stimmung von Aussagen, aus der Domäne der Softwareentwicklung, nachgewiesen werden, so kann dieses Wissen genutzt werden, um eine Kalibrierung von Stimmungsanalysetools auf die Wahrnehmung einzelner Entwickler vorzunehmen. Eine Möglichkeit dazu wäre die automatisierte Erkennung von Persönlichkeitsmerkmalen durch Software, wobei jedoch zusätzlich eine Anpassung auf die Domäne des Software Engineerings nötig ist [10]. Weiterentwicklungen, die auf solch personalisierte Stimmungsanalysetools hinarbeiten, haben das Potenzial, den Weg für eine industrielle Anwendung der Stimmungsanalyse im Software Engineering zu ebnen.

Anhang A

Ergänzende Informationen

Dieser Anhang enthält ergänzende Informationen zum Hauptteil dieser Arbeit, die dem Verständnis des Lesers dienen sollen.

A.1 Erhebungsdesign

In Kapitel 4 wurde auf das Erhebungsdesign der Umfrage von Obaidi et al. [111] aus der Studie von Herrmann et al. [54] Bezug genommen. Tabelle A.1 enthält, in A.1a bis A.1e unterteilt, eine deutschsprachige Übersetzung des Erhebungsdesigns, welches im Fachartikel von Herrmann et al. [54] und auf *Zenodo* [111] im englischsprachigen Original zu finden ist.

Tabelle A.1: Erhebungsdesign der Umfrage von Obaidi et al. [111] in sinngemäßer deutschsprachiger Übersetzung

(a) Demografie

Fragen	**Antwortmöglichkeiten**
Wie alt sind Sie?	18 – 99
Welches Geschlecht haben Sie?	☐ Männlich / ☐ Weiblich / ☐ Divers
Ist Englisch Ihre Muttersprache?	☐ Ja / ☐ Nein
Wie oft kommunizieren Sie auf Englisch?	☐ Nie ☐ Gelegentlich ☐ Einmal pro Woche ☐ Mehrmals pro Woche ☐ Täglich

(b) Zugehörigkeit zur Informatik

Fragen	**Antwortmöglichkeiten**
Identifizieren Sie sich als Informatiker?	Ja / Nein
Was ist Ihr beruflicher Status? (Mehrfachauswahl)	☐ Student ☐ Angestellt in der Akademia ☐ Angestellt in der Wirtschaft ☐ Im Ruhestand ☐ Arbeitslos ☐ Andere (Freitext-Antwort)

(c) Berufserfahrung

Fragen	**Antwortmöglichkeiten**
Haben Sie Erfahrung mit Programmierung?	Ja / Nein
Wie schätzen Sie Ihre Programmierkenntnisse ein?	1 — 2 — 3 — 4 — 5 1: grundlegend, 5: fortgeschritten
Wie viele Jahre an Erfahrung haben Sie als professioneller Entwickler?	0 – 99
Wie familiär sind Sie damit, in einem Entwicklungsteam zu arbeiten?	1 — 2 — 3 — 4 — 5 1: wenig familiär, 5: sehr familiär
Wie viele Jahre an Erfahrung haben Sie als professioneller Entwickler in einem Entwicklungsteam?	0 – 99

(d) Annotation

Frage	**Antwortmöglichkeiten**
Welche Sentiment-Polaritäten würden Sie den folgenden Aussagen anhand Ihrer Wahrnehmung zuweisen?	*negativ* — *neutral* — *positiv* **×100**

(e) Annotationskriterium

Frage	**Antwortmöglichkeiten**
Nach welchem Kriterium haben Sie den Aussagen die Sentiment-Polaritäten zugeordnet? (Mehrfachauswahl)	☐ Inhalt ☐ Ton ☐ Andere (Freitext-Antwort)

A.2 Aussagen der Datenbasis

Nachfolgend befindet sich eine Auflistung der 96 Aussagen aus der Datenbasis dieser Arbeit, welche in den Ergebnissen in Kapitel 5 als V_1 bis V_{96} referenziert wurden, sowie die Information darüber, welcher der beiden Datensätze [84, 107] die Quelle der jeweiligen Aussagen ist.

V_1: „*Trust URI.*“ - Lin et al. [84]

V_2: „*(Hopefully with a good example.)*“ - Lin et al. [84]

V_3: „*It's really good.*“ - Lin et al. [84]

V_4: „*I don't know what else to do to make things to work.*“ - Lin et al. [84]

V_5: „*The data structure you are saving your data is not very optimal for the days with daylight saving time.*“ - Lin et al. [84]

V_6: „*Use CDI's CODE with CODE or JSF's CODE with CODE.*“ - Lin et al. [84]

V_7: „*quit spamming my notifications please, kthkbye*“ - Novielli et al. [107]

V_8: „*If I still wasn't explicit enough: GIT_HASH I can either submit a pull request or just forget it :)*“ - Novielli et al. [107]

V_9: „*why allocate a new String instance? def apply(name: String): Node = hash(name) def fromHash(hash: String): Node = hash*“ - Novielli et al. [107]

V_{10}: „*I know a lot of server admins will be happy about that.*“ - Novielli et al. [107]

V_{11}: „*else your GUI will be Hanged.*“ - Lin et al. [84]

V_{12}: „*Because I thought I didn't need the quotes, which I no know I do.*“ - Novielli et al. [107]

V_{13}: „*So you can easy reduce count of objects by factor two (one byte [ ] instead of pair String + char [ ]), and array length of UTF-8 symbols usually less than length of UTF-16 chars.*“ - Lin et al. [84]

V_{14}: „*Sure can do. Privacy is a right!*“ - Novielli et al. [107]

V_{15}: „*It does its job, but was built for our use case.*“ - Lin et al. [84]

V_{16}: „*You have problem with classpath.*“ - Lin et al. [84]

V_{17}: „*This is a silly decision that does not represent the majority of the rails community. Official plugin? Absolutely. Core? **No fucking way**.*“ - Novielli et al. [107]

V_{18}: „*Most awesome! :+1:*“ - Novielli et al. [107]

V_{19}: „*Well a solution that works for me.*“ - Lin et al. [84]

V_{20}: „*There is a good example showing how to put a file onto WebDAV server.*“ - Lin et al. [84]

V_{21}: „*Comparison time should be fast, so total run time should be only slightly more than sum of run time for each ordered query.*“ - Lin et al. [84]

V_{22}: „*There is a slight improvement that you can do that uses the CODE's internal cache by using its CODE method.*“ - Lin et al. [84]

V_{23}: „*The maintainers' rationale is covered in the JSF specification.*“ - Lin et al. [84]

V_{24}: „*Still, we need a potentially unlimited number of buffers. What should we do with them if we don't need them any more if we don't let them be garbage collected?*“ - Novielli et al. [107]

V_{25}: „*This is the only one that bothers me. If an old compilers fails to optimize a static 'strlen', that's ok, the code will run a bit slower... But if an old compiler fails to optimize this one, this will just not compile. We need to change this to a numeric* “ - Novielli et al. [107]

V_{26}: „*Now we're getting to the good part.*“ - Lin et al. [84]

V_{27}: „*i was afraid it'd do unimaginable things!*“ - Novielli et al. [107]

V_{28}: „*This impl is weird. Uses system identityHashCode for hashCode AND equals?*“ - Novielli et al. [107]

V_{29}: „*Its a little complicated, there is a full example in this loadNativeLibrary () method.*“ - Lin et al. [84]

V_{30}: „*@timmywil Sounds good!*“ - Novielli et al. [107]

V_{31}: „*lol :)*“ - Novielli et al. [107]

V_{32}: „*There, I fixed the two cleanup-issues :) Sorry about the sloppyness!*“ - Novielli et al. [107]

V_{33}: „*true, but then i would also need to be host, methods and schemes attributes instead of elements. see https://github.com/symfony/symfony/blob/master/src/Symfony/Component/Routing/Tests/Fixtures/validpattern.xml*“ - Novielli et al. [107]

V_{34}: „*That's what obfuscation does. We do our best to unobfuscate but only for class names. This is a mojang thing, not us.*“ - Novielli et al. [107]

V_{35}: „*Let me know if there are any other details that might help!*“ - Lin et al. [84]

V_{36}: „*I still think it's not the best design, but I did quite a bit of testing and can say with a good bit of confidence that it had no real impact on performance.*“ - Lin et al. [84]

V_{37}: „*oh, forget it, my bad; maybe add a comment?*“ - Novielli et al. [107]

V_{38}: „*Lol.*“ - Novielli et al. [107]

V_{39}: „*@ultrasaurus: skip_before_filter :verify_authenticity_token, :only => [:aircrafts_by_manufacturer] ?*“ - Novielli et al. [107]

V_{40}: „*oh nice find, that's been bugging the crap out of me*“ - Novielli et al. [107]

V_{41}: „*Which we currently are saved from using CODE.*“ - Lin et al. [84]

V_{42}: „*https://github.com/jquery/jquery/commit/GIT_HASH thanks for your expert eye.*“ - Novielli et al. [107]

V_{43}: „*Yay for improving consistency, +1*“ - Novielli et al. [107]

V_{44}: „*If I run the code in the GUI, it just hangs.*“ - Lin et al. [84]

V_{45}: „*I'm having trouble figuring out how to do this in drools 6 +.*“ - Lin et al. [84]

V_{46}: „*OMG stupid me*“ - Novielli et al. [107]

V_{47}: „*I think they should be the other way around. It's the 'consistentHash' that should be in the atomic since you don't want to overwrite it with a stale 'consistentHash'. Overwriting the 'consistentHashRoutees' with stale values will just trigger a new updat*“ - Novielli et al. [107]

V_{48}: „*If i access the /deploy_keys on a project and there is a key that is always assigned to multiple Projects, i see the key multiple times, think you must call uniq or do a group deploy_key_id via sql If i add the key, the -@enabled_keys remove all duplic*“ - Novielli et al. [107]

V_{49}: „*I am trying to link my native library to FILE_NAME application but when I try to run it I get a CODE exception complaining about missing symbols (CODE).*“ - Lin et al. [84]

V_{50}: „*Thanks Steven. For a dot release, 3.1 sure changed a lot of fundamental things. :(*“ - Novielli et al. [107]

V_{51}: „*So, I thought I could create FILE_NAME that counts and pass the stream through a API_NAME code is in this answer.*“ - Lin et al. [84]

V_{52}: „*The following is the error I keep getting.*“ - Lin et al. [84]

V_{53}: „*I have looked and found that there are some packages that would automatically enter my keyring on login but that isn't really an option.*“ - Lin et al. [84]

V_{54}: „*So, everything builds fine, but when we try to deploy the application to GFNUMBER we get the FILE_NAME file not found "error.*“ - Lin et al. [84]

V_{55}: „*Very good example of steady pooling readHere.*“ - Lin et al. [84]

V_{56}: „*It dependes where it is exactly located.*“ - Lin et al. [84]

V_{57}: „*I can successfully call my service but when generating the response, it seems to crash.*“ - Lin et al. [84]

V_{58}: „*Finally, an IDE would be helpful for you.*“ - Lin et al. [84]

V_{59}: „*When I run this on Windows, the console tells me.*“ - Lin et al. [84]

V_{60}: „*The following code therefore, might be (as I have not attempted using it) a better one, if you are willing to use Mojarra specific classes.*“ - Lin et al. [84]

V_{61}: „*Here's an example of a JSON structure.*“ - Lin et al. [84]

V_{62}: „*You're twisting my words a little bit, but it's ok :-) Your examples are strange. [‘krsort()‘](http://php.net/manual/en/function.krsort.php) is designed to sort arrays while maintaining keys (i.e. associations, as the name suggests) - hence it should not*“ - Novielli et al. [107]

V_{63}: „*At the moment I am receiving a ClassNotFoundException to API_NAME.*“ - Lin et al. [84]

V_{64}: „*Can you just do “‘javascript var appender = url.length > 0 && url[url.length - 1] == '?' : '&' : '?'; ““* - Novielli et al. [107]

V_{65}: „*lol just refreshed page for me and I'm seeing all the comments on the comments, I think GitHub needs to be more Real time! =)*“ - Novielli et al. [107]

V_{66}: „*If you can change the code it helps to use CODE (with no host parameter or with a connected socket).*“ - Lin et al. [84]

V_{67}: „*I have no idea either, I just trust the spray guys*“ - Novielli et al. [107]

V_{68}: „*Is this good to go then?*“ - Novielli et al. [107]

V_{69}: „*The following warning messages should be ignored, they are 'false-positive 'alert messages.*“ - Lin et al. [84]

V_{70}: „*Using a concurrent data structure lets you get rid of the synchronized block.*“ - Lin et al. [84]

V_{71}: „*Alright, one more try at GIT_HASH*“ - Novielli et al. [107]

V_{72}: „*Is link to wkipedia article enough? :)*“ - Novielli et al. [107]

V_{73}: „*Fixed. But, is this rule documented somewhere? There are other places where short syntax is used https://gist.github.com/4352487. Does this means that cast is done with: “‘php (integer) $var; “‘ as well? Again, there are many short syntax*“ - Novielli et al. [107]

V_{74}: „*md5 not good enough for you?*“ - Novielli et al. [107]

V_{75}: „*On the minus side, I hate curly braces. On the plus side, I hate painting bikesheds. Style guides, while vaguely creepy and authoritarian, definitely minimize the number of painters we need to keep around. (And fwiw, consistency could be maintained w*“ - Novielli et al. [107]

V_{76}: „*I think hints are specific to the ORM. So having them in the generic DoctrineType looks weird to me*“ - Novielli et al. [107]

V_{77}: „*...because this is how it was before :)*“ - Novielli et al. [107]

V_{78}: „*A bi thanks for this :) we all are really happy that now this is fully supported by Core. Now is missing Vehicles support and MaNGOS will rulez =D Congrats for all your work*“ - Novielli et al. [107]

V_{79}: „*Why was this reverted? :(*“ - Novielli et al. [107]

V_{80}: „*And it works like a charm now SMILE_FACE.*“ - Lin et al. [84]

V_{81}: „*How would you prefer it to be implemented?*“ - Novielli et al. [107]

V_{82}: „*We ran into the same sort of problem with Flex and JPA/Hibernate.*“ - Lin et al. [84]

V_{83}: „*I ended up using ‘strtotime()‘ as the various formats that I saw in the wild and in the specs resulted in icky code. However, strtotime() seems to be able to sort everything out.*“ - Novielli et al. [107]

V_{84}: „*which someone converted to a JSON string as follows.*“ - Lin et al. [84]

V_{85}: „*Don’t use.*“ - Lin et al. [84]

V_{86}: „*Sorry, my fail https://github.com/Imprtat/TrinityCore/commit/GIT_HASH* “ - Novielli et al. [107]

V_{87}: „*Oh, sorry... It’s patch. :(In short: activerecord/lib/active_record/connection_adapters/postgresql_adapter.rb:856 add unless @config[:database] == ‘template1’*“ - Novielli et al. [107]

V_{88}: „*This might prove to be useful unbescape.*“ - Lin et al. [84]

V_{89}: „*Ops, that seems to have [failed the build](https://travis-ci.org/rails/rails/builds/6066789) =(*“ - Novielli et al. [107]

V_{90}: „*final class?*“ - Novielli et al. [107]

V_{91}: „*Hope this helps.*“ - Lin et al. [84]

V_{92}: „*I'm investigating the problem and seems like I already have a fix to the first part of the issue. Working on second part, will report soon. EDIT: hm, seems fixing the automatic deinterlacer is somewhat more difficult... :(*“ - Novielli et al. [107]

V_{93}: „*Problem is pretty API_NAME.*“ - Lin et al. [84]

V_{94}: „*And a string contains the name of the enum.*“ - Lin et al. [84]

V_{95}: „*However, having said that, if you have a concrete need for modifying the mock request created by CODE manually and then having that re-used with CODE, you can create the FILE_NAME in your project.*“ - Lin et al. [84]

V_{96}: „*To allow your programs to work without re-compiling, run your app as.*“ - Lin et al. [84]

Literaturverzeichnis

[1] Maurice S. Bartlett and Ralph H. Fowler. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 160(901):268–282, Royal Society, London, 1937. doi:10.1098/rspa.1937.0109.

[2] Bojana D. Basic. Distance measures. In *International Encyclopedia of Statistical Science*, pages 397–398. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-04898-2_626.

[3] Joachim Behnke. Das Logit-Modell. In *Logistische Regressionsanalyse: Eine Einführung*, pages 23–35. Springer Fachmedien, Wiesbaden, 2015. doi:10.1007/978-3-658-05082-5_3.

[4] Richard E. Bellman. *Dynamic Programming*, volume 33 of *Princeton Landmarks in Mathematics and Physics*. Princeton University Press, Princeton, NJ, 2010. doi:10.2307/j.ctv1nxcw0f.

[5] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Kluwer Academic Publishers, Norwell, MA, USA, 1996. doi:10.1023/A:1018054314350.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Kluwer Academic Publishers, Norwell, MA, USA, 2001. doi:10.1023/A:1010933404324.

[7] Frederick P. Brooks Jr. No silver bullet essence and accidents of software engineering. *Computer*, 20(4):10–19, IEEE, New York, NY, USA, 1987. doi:10.1109/MC.1987.1663532.

[8] Morton B. Brown and Alan B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, Taylor & Francis, Oxfordshire, 1974. doi:10.1080/01621459.1974.10482955.

[9] Ted Byrt, Janet Bishop, and John B. Carlin. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429, Elsevier, Amsterdam, 1993. doi:10.1016/0895-4356(93)90018-V.

[10] Fabio Calefato and Filippo Lanubile. Using personality detection tools for software engineering research: How far can we go? *ACM Transactions on Software Engineering and Methodology*, 31(3):1–42, Association for Computing Machinery, New York, NY, USA, 2022. doi:10.1145/3491039.

[11] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment polarity detection for software development. In *Proceedings of the 40th International Conference on Software Engineering (ICSE'18)*, page 128, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3180155.3182519.

[12] Zhenpeng Chen, Yanbin Cao, Huihan Yao, Xuan Lu, Xin Peng, Hong Mei, and Xuanzhe Liu. Emoji-powered sentiment and emotion detection from software developers' communication data. *ACM Transactions on Software Engineering and Methodology*, 30(2):1–48, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3424308.

[13] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, Sage Publications, Inc., Thousand Oaks, CA, USA, 1960. doi:10.1177/001316446002000104.

[14] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*, 3rd edition. Routledge, Oxfordshire, 2002. doi:10.4324/9780203774441.

[15] Paul T. Costa Jr. and Robert R. McCrae. The revised NEO personality inventory (NEO-PI-R). *The SAGE Handbook of Personality Theory and Assessment: Volume 2 - Personality Measurement and Testing*, pages 179–198, Sage Publications, Ltd., London, 2008. doi:10.4135/9781849200479.n9.

[16] Lee J. Cronbach. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16:297–334, Springer International Publishing, Cham, 1951. doi:10.1007/BF02310555.

[17] Adele Cutler, David R. Cutler, and John R. Stevens. Random forests. In *Ensemble Machine Learning: Methods and Applications*, pages 157–175. Springer US, Boston, MA, 2012. doi:10.1007/978-1-4419-9326-7_5.

[18] Alexandra de Raadt, Matthijs J. Warrens, Roel J. Bosker, and Henk A. L. Kiers. A comparison of reliability coefficients for ordinal rating scales. *Journal of Classification*, 38(3):519–543, Springer, Berlin, Heidelberg, 2021. doi:10.1007/s00357-021-09386-5.

[19] Yves J. Decady and Roland Thomas. A simple test of association for contingency tables with multiple column responses. *Biometrics*, 56(3):893–896, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2000. doi:10.1111/j.0006-341X.2000.00893.x.

[20] Giuseppe Destefanis, Marco Ortu, David Bowes, Michele Marchesi, and Roberto Tonelli. On measuring affects of github issues' commenters. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion'18)*, page 14–19, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3194932.3194936.

[21] Enrico di Bella, Alberto Sillitti, and Giancarlo Succi. A multivariate classification of open source developers. *Information Sciences*, 221:72–83, Elsevier, Amsterdam, 2013. doi:10.1016/j.ins.2012.09.031.

[22] Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. Entity-level sentiment analysis of issue comments. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion'18)*, page 7–13, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3194932.3194935.

[23] Márcio A. Diniz and Tiago M. Magalhães. Logistic regression and related methods. In *Principles and Practice of Clinical Trials*, pages 1–23. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-319-52677-5_122-2.

[24] Yadolah Dodge. Contingency table. In *The Concise Encyclopedia of Statistics*, pages 110–111. Springer, New York, NY, USA, 2008. doi:10.1007/978-0-387-32833-1_77.

[25] Yadolah Dodge. Multicollinearity. In *The Concise Encyclopedia of Statistics*, pages 362–363. Springer, New York, NY, USA, 2008. doi:10.1007/978-0-387-32833-1_272.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*, 2nd edition. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2000. doi:10.1007/s00357-007-0015-9.

[27] Nataša Erjavec. Tests for homogeneity of variance. In *International Encyclopedia of Statistical Science*, page 1595. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-04898-2_590.

[28] Brian S. Everitt. *An Introduction to Latent Variable Models*. Monographs on Statistics and Applied Probability (MSAP). Springer Netherlands, Dordrecht, 1984. doi:10.1007/978-94-009-5564-6.

[29] Alvan R. Feinstein and Domenic V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, Elsevier, Amsterdam, 1990. doi:10.1016/0895-4356(90)90158-L.

[30] Robert Feldt, Richard Torkar, Lefteris Angelis, and Maria Samuelsson. Towards individualized software engineering: Empirical studies should collect psychometrics. In *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'08)*, page 49–52, Association for Computing Machinery, New York, NY, USA, 2008. doi:10.1145/1370114.1370127.

[31] David Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38, [Oxford University Press, Biometrika Trust], Oxford, 1993. doi:10.1093/biomet/80.1.27.

[32] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1936. doi:10.1111/j.1469-1809.1936.tb02137.x.

[33] Ronald A. Fisher. Statistical methods for research workers. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 66–70. Springer, New York, NY, USA, 1992. doi:10.1007/978-1-4612-4380-9_6.

[34] Barbara Fredrickson. The role of positive emotions in positive psychology. *The American Psychologist*, 56(3):218–226, American Psychological Association, Washington, DC, USA, 2001. doi:10.1037/0003-066X.56.3.218.

[35] Max Garzon, Ching-Chi Yang, Deepak Venugopal, Nirman Kumar, Kalidas Jana, and Lih-Yuan Deng, editors. *Dimensionality Reduction in Data Science*. Springer International Publishing, Cham, 2022. doi:10.1007/978-3-031-05371-9.

[36] R. Stuart Geiger, Kevin Yu, Yanlai Yang, Mindy Dai, Jie Qiu, Rebekah Tang, and Jenny Huang. Garbage in, garbage out? Do machine learning application papers in social computing report where human-labeled training data comes from? In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT*'20)*, page 325–336, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3351095.3372862.

[37] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, pages 22–30, Springer, Berlin, Heidelberg, 2004. doi:10.1007/978-3-540-24775-3_5.

[38] Xavier Golay, Spyros Kollias, Gautier Stoll, Dieter Meier, Anton Valavanis, and Peter Boesiger. A new correlation-based fuzzy logic clustering algorithm for FMRI. *Magnetic Resonance in Medicine*, 40(2):249–260, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1998. doi:10.1002/mrm.1910400211.

[39] Lewis R. Goldberg, John A. Johnson, Herbert W. Eber, Robert Hogan, Michael C. Ashton, C. Robert Cloninger, and Harrison G. Gough. The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, 40(1):84–96, Elsevier, Amsterdam, 2006. doi:10.1016/j.jrp.2005.08.007.

[40] William S. "Student" Gosset. The probable error of a mean. *Biometrika*, 6(1):1–25, [Oxford University Press, Biometrika Trust], Oxford, 1908. doi:10.2307/2331554.

[41] John C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3/4):325–338, [Oxford University Press, Biometrika Trust], Oxford, 1966. doi:10.2307/2333639.

[42] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. Happy software developers solve problems better: psychological measurements in empirical software engineering. *PeerJ*, 2:e289, PeerJ Inc., San Diego, CA, USA, 2014. doi:10.7717/peerj.289.

[43] Daniel Graziotin, Xiaofeng Wang, and Pekka Abrahamsson. How do you feel, developer? an explanatory theory of the impact of affects on programming performance. *PeerJ Computer Science*, 1:e18, PeerJ Inc., San Diego, CA, USA, 2015. doi:10.7717/peerj-cs.18.

[44] Jürgen Groß. The linear regression model. In *Linear Regression*, pages 33–86. Springer, Berlin, Heidelberg, 2003. doi:10.1007/978-3-642-55864-1_2.

[45] Carl Gutwin, Reagan Penner, and Kevin Schneider. Group awareness in distributed software development. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW'04)*, page 72–81, Association for Computing Machinery, New York, NY, USA, 2004. doi:10.1145/1031607.1031621.

[46] Emitza Guzman, David Azócar, and Yang Li. Sentiment analysis of commit comments in github: An empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR'14)*, page 352–355, Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2597073.2597118.

[47] Kilem L. Gwet. Intrarater reliability. In *Wiley Encyclopedia of Clinical Trials*, pages 1–13. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008. doi:10.1002/9780471462422.eoct631.

[48] Joseph F. Hair, Mary Wolfinbarger, Arthur H. Money, Phillip Samouel, and Michael J. Page. *The Essentials of Business Research Methods*, 3rd edition. Routledge, Oxfordshire, 2015. doi:10.4324/9781315716862.

[49] Maria Halkidi. Hierarchial clustering. In *Encyclopedia of Database Systems*, pages 1–5. Springer, New York, NY, USA, 2016. doi:10.1007/978-1-4899-7993-3_604-2.

[50] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Linear methods for classification. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, pages 101–137. Springer, New York, NY, USA, 2009. doi:10.1007/978-0-387-84858-7_4.

[51] Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. *Statistics in Medicine*, 21(16):2409–2419, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2002. doi:10.1002/sim.1047.

[52] Norbert K. Henze and Bernd Zirkler. A class of invariant consistent tests for multivariate normality. *Communications in Statistics - Theory and Methods*, 19(10):3595–3617, Taylor & Francis, Oxfordshire, 1990. doi:10.1080/03610929008830400.

[53] Marc Herrmann and Jil Klünder. From textual to verbal communication: Towards applying sentiment analysis to a software project meeting. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 371–376, IEEE, New York, NY, USA, 2021. doi:10.1109/REW53955.2021.00065.

[54] Marc Herrmann, Martin Obaidi, Larissa Chazette, and Jil Klünder. On the subjectivity of emotions in software projects: How reliable are pre-labeled data sets for sentiment analysis? *Journal of Systems and Software*, 193:111448, Elsevier, Amsterdam, 2022. doi:10.1016/j.jss.2022.111448.

[55] Adolf Heß. Der Kosinussatz. In *Trigonometrie für Maschinenbauer und Elektrotechniker: Ein Lehr- und Aufgabenbuch für den Unterricht und zum Selbststudium*, pages 73–75. Springer, Berlin, Heidelberg, 1911. doi:10.1007/978-3-662-38242-4_12.

[56] Ralf. D. Hilgers, Nicole Heussen, and Sven Stanzel. Korrelationskoeffizient nach Pearson. In *Lexikon der Medizinischen Laboratoriumsdiagnostik*, 3rd edition, pages 1389–1389. Springer, Berlin, Heidelberg, 2019. doi:10.1007/978-3-662-48986-4_1763.

[57] Alexander Hinneburg. Visualizing clustering results. In *Encyclopedia of Database Systems*, 2nd edition, pages 4556–4566. Springer, New York, NY, USA, 2018. doi:10.1007/978-1-4614-8265-9_617.

[58] Stefan Hougardy and Jens Vygen. Gaussian elimination. In *Algorithmic Mathematics*, pages 133–154. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-39558-6_11.

[59] David C. Howell. Chi-square test: Analysis of contingency tables. In *International Encyclopedia of Statistical Science*, pages 250–252. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-04898-2_174.

[60] Carl J. Huberty. Discriminant analysis. *Review of Educational Research*, 45(4):543–598, [Sage Publications, Inc., American Educational Research Association], Thousand Oaks, CA, USA, 1975. doi:10.2307/1170065.

[61] Nasif Imtiaz, Justin Middleton, Peter Girouard, and Emerson Murphy-Hill. Sentiment and politeness analysis tools on developer discussions are unreliable, but so are people. In *Proceedings of the Third International Workshop on Emotion Awareness in Software Engineering*, page 55–61, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3194932.3194938.

[62] Md Rakibul Islam and Minhaz F. Zibran. Towards understanding and exploiting developers' emotional variations in software engineering. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 185–192, IEEE, New York, NY, USA, 2016. doi:10.1109/SERA.2016.7516145.

[63] Md Rakibul Islam and Minhaz F. Zibran. A comparison of dictionary building methods for sentiment analysis in software engineering text. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'17)*, pages 478–479, IEEE, New York, NY, USA, 2017. doi:10.1109/ESEM.2017.67.

[64] Md Rakibul Islam and Minhaz F. Zibran. Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. *Journal of Systems and Software*, 145:125–146, Elsevier, Amsterdam, 2018. doi:10.1016/j.jss.2018.08.030.

[65] Alan Julian Izenman. Linear discriminant analysis. In *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, pages 237–280. Springer, New York, NY, USA, 2008. doi:10.1007/978-0-387-78189-1_8.

[66] Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics (SSS). Springer, New York, NY, USA, 1986. doi:10.1007/978-1-4757-1904-8.

[67] Capers Jones. Software metrics: good, bad and missing. *Computer*, 27(9):98–100, IEEE, New York, NY, USA, 1994. doi:10.1109/2.312055.

[68] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, Springer International Publishing, Cham, 2017. doi:10.1007/s10664-016-9493-x.

[69] Damir Kalpić and Nikica Hlupić. Multivariate normal distributions. In *International Encyclopedia of Statistical Science*, pages 907–910. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-04898-2_623.

[70] Hyun Kang. The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, 64(5):402–406, Korean Society of Anesthesiologists, Seoul, 2013. doi:10.4097/kjae.2013.64.5.402.

[71] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1990. doi:10.1002/9780470316801.

[72] Jon R. Kettenring. The practice of cluster analysis. *Journal of Classification*, 23:3–30, Springer, Berlin, Heidelberg, 2006. doi:10.1007/s00357-006-0002-6.

[73] William R. Klecka. *Discriminant analysis*. Quantitative Applications in the Social Sciences. Sage Publications, Inc., Thousand Oaks, CA, USA, 1980. doi:10.4135/9781412983938.

[74] Jil Klünder, Julian Horstmann, and Oliver Karras. Identifying the mood of a software development team by analyzing text-based communication in chats with machine learning. In *Human-Centered Software Engineering*, pages 133–151, Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-64266-2_8.

[75] Jil Klünder, Dzejlana Karajic, Paolo Tell, Oliver Karras, Christian Münkel, Jürgen Münch, Stephen G. MacDonell, Regina Hebig, and Marco Kuhrmann. Determining context factors for hybrid development methods with trained models. In *Proceedings of the International Conference on Software and System Processes (ICSSP'20)*, page 61–70, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3379177.3388898.

[76] Jil Klünder, Nils Prenner, Ann-Kathrin Windmann, Marek Stess, Michael Nolting, Fabian Kortum, Lisa Handke, Kurt Schneider, and Simone Kauffeld. Do you just discuss or do you solve? Meeting analysis in a software project at early stages. In *Proceedings of the IEEE/ACM 42 International Conference on Software Engineering Workshops (ICSEW'20)*, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3387940.3391468.

[77] Jil Klünder, Carolin Unger-Windeler, Fabian Kortum, and Kurt Schneider. Team meetings and their relevance for the software development process over time. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 313–320, IEEE, New York, NY, USA, 2017. doi:10.1109/SEAA.2017.57.

[78] Robert E. Kraut and Lynn A. Streeter. Coordination in software development. *Communications of the ACM*, 38(3):69–81, Association for Computing Machinery, New York, NY, USA, 1995. doi:10.1145/203330.203345.

[79] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, Association for Computing Machinery, New York, NY, USA, 2009. doi:10.1145/1497577.1497578.

[80] Wojtek J. Krzanowski. The performance of fisher's linear discriminant function under non-optimal conditions. *Technometrics*, 19(2):191–200, Taylor & Francis, Oxfordshire, 1977. doi:10.1080/00401706.1977.10489527.

[81] Filippo Lanubile. Collaboration in distributed software development. In *Software Engineering: International Summer Schools (ISSSE 2006-2008)*, pages 174–193. Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-95888-8_7.

[82] Cheng Li. Little's test of missing completely at random. *The Stata Journal*, 13(4):795–809, Stata Press, College Station, TX, USA, 2013. doi:10.1177/1536867X1301300407.

[83] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. Using discriminant analysis for multi-class classification: An experimental investigation. *Knowledge and Information Systems*, 10(4):453–472, Springer, Berlin, Heidelberg, 2006. doi:10.1007/s10115-006-0013-y.

[84] Bin Lin, Fiorella Zampetti, Gabriele Bavota, Massimiliano Di Penta, Michele Lanza, and Rocco Oliveto. Sentiment analysis for software engineering: How far can we go? (icse'18). In *Proceedings of the 40th International Conference on Software Engineering*, page 94–104, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3180155.3180195.

[85] Roderick J. A. Little. A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(404):1198–1202, Taylor & Francis, Oxfordshire, 1988. doi:10.1080/01621459.1988.10478722.

[86] Roderick J. A. Little and Donald B. Rubin. Single imputation methods. In *Statistical Analysis with Missing Data*, 2nd edition, chapter 4, pages 59–74. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2002. doi:10.1002/9781119013563.ch4.

[87] Roderick J. A. Little and Donald B Rubin. *Statistical Analysis with Missing Data*, 3rd edition. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2019. doi:10.1002/9781119482260.

[88] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*, 2nd edition, page 664–704. Chapman & Hall/CRC, New York, NY, USA, 2010. doi:10.1201/9781420085938.

[89] Nicholas T. Longford. Single imputation and related methods. In *Missing Data and Small-Area Estimation: Modern Analytical Equipment for the Survey Statistician*, Statistics for Social and Behavioral Sciences (SSBS), pages 37–58. Springer, London, 2005. doi:10.1007/1-84628-195-4_3.

[90] Malcolm Maclure and Walter C. Willett. Misinterpretation and misuse of the kappa statistic. *American Journal of Epidemiology*, 126(2):161–169, [Oxford University Press, Johns Hopkins Bloomberg School of Public Health], Oxford, 1987. doi:10.1093/aje/126.2.161.

[91] Paul Madley-Dowd, Rachael Hughes, Kate Tilling, and Jon Heron. The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of Clinical Epidemiology*, 110:63–73, Elsevier, Amsterdam, 2019. doi:10.1016/j.jclinepi.2019.02.016.

[92] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50 – 60, Institute of Mathematical Statistics, Beachwood, OH, USA, 1947. doi:10.1214/aoms/1177730491.

[93] Francesco Masulli and Stefano Rovetta. Clustering high-dimensional data. In *Revised Selected Papers of the First International Workshop on Clustering High–Dimensional Data (CHDD 2012)*, volume 7627, pages 1–13, Springer, Berlin, Heidelberg, 2015. doi:10.1007/978-3-662-48577-4_1.

[94] Ian R McChesney and Séamus Gallagher. Communication and coordination practices in software engineering projects. *Information and Software Technology*, 46(7):473–489, Elsevier, Amsterdam, 2004. doi:10.1016/j.infsof.2003.10.001.

[95] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, [Elsevier, Ain Shams University], Amsterdam, 2014. doi:10.1016/j.asej.2014.04.011.

[96] Andrew Meneely, Laurie Williams, Will Snipes, and Jason Osborne. Predicting failures with developer networks and social network analysis. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT'08/FSE-16)*, page 13–23, Association for Computing Machinery, New York, NY, USA, 2008. doi:10.1145/1453101.1453106.

[97] André N. Meyer, Thomas Zimmermann, and Thomas Fritz. Characterizing software developers by perceptions of productivity. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'17)*, page 105–110, IEEE, New York, NY, USA, 2017. doi:10.1109/ESEM.2017.17.

[98] Ivan Mistrík, John Grundy, André van der Hoek, and Jim Whitehead. Collaborative software engineering: Challenges and prospects. In *Collaborative Software Engineering*, pages 389–403. Springer, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-10294-3_19.

[99] Pablo Montero and José A. Vilar. Tsclust: An r package for time series clustering. *Journal of Statistical Software*, 62(1):1–43, Foundation for Open Access Statistics, Los Angeles, CA, USA, 2014. doi:10.18637/jss.v062.i01.

[100] Kenneth Moreland. Diverging color maps for scientific visualization. In *Advances in Visual Computing*, pages 92–103, Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-10520-3_9.

[101] Alessandro Murgia, Parastou Tourani, Bram Adams, and Marco Ortu. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR'14)*, page 262–271, Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2597073.2597086.

[102] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery*, 2(1):86–97, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2012. doi:10.1002/widm.53.

[103] Ingunn Myrtveit, Erik. Stensrud, and Ulf H. Olsson. Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods. *IEEE Transactions on Software Engineering*, 27(11):999–1013, IEEE, New York, NY, USA, 2001. doi:10.1109/32.965340.

[104] David Nettleton. Selection of variables and factor derivation. In *Commercial Data Mining*, pages 79–104. Morgan Kaufmann, Boston, MA, USA, 2014. doi:10.1016/B978-0-12-416602-8.00006-6.

[105] Franz J. Neyer and Jens B. Asendorpf. Methodik. In *Psychologie der Persönlichkeit*, 5th edition, pages 81–130. Springer, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-30264-0_3.

[106] Frank Nielsen. Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pages 195–211. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-21903-5_8.

[107] Nicole Novielli, Fabio Calefato, Davide Dongiovanni, Daniela Girardi, and Filippo Lanubile. Can we use se-specific sentiment analysis tools in a cross-platform setting? In *Proceedings of the 17th International Conference on Mining Software Repositories (MSR'20)*, pages 158–168, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3379597.3387446.

[108] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. Towards discovering the role of emotions in stack overflow. In *Proceedings of the 6th International Workshop on Social Software Engineering (SSE'14)*, page 33–36, Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2661685.2661689.

[109] Nicole Novielli, Fabio Calefato, and Filippo Lanubile. A gold standard for emotion annotation in stack overflow. In *Proceedings of the 15th International Conference on Mining Software Repositories (MSR'18)*, page 14–17, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3196398.3196453.

[110] Nicole Novielli, Daniela Girardi, and Filippo Lanubile. A benchmark study on sentiment analysis for software engineering research. In *Proceedings of the 15th International Conference on Mining Software Repositories (MSR'18)*, page 364–375, Association for Computing Machinery, New York, NY, USA, 2018. doi:10.1145/3196398.3196403.

[111] Martin Obaidi, Marc Herrmann, Larissa Chazette, and Jil Klünder. Dataset: SentiSurvey for sentiment analysis in software projects. Zenodo, 2022. doi:10.5281/zenodo.6611729.

[112] Martin Obaidi and Jil Klünder. Development and application of sentiment analysis tools in software engineering: A systematic literature review. In *Evaluation and Assessment in Software Engineering (EASE'21)*, page 80–89, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3463274.3463328.

[113] Marco Ortu, Giuseppe Destefanis, Bram Adams, Alessandro Murgia, Michele Marchesi, and Roberto Tonelli. The jira repository dataset: Understanding social aspects of software development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE'15)*, pages 1–4, Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2810146.2810147.

[114] Ashwin Pajankar. Introduction to data visualization with seaborn. In *Hands-on Matplotlib: Learn Plotting and Visualizations with Python 3*, pages 243–267. Apress Media LLC, Berkeley, CA, USA, 2022. doi:10.1007/978-1-4842-7410-1_17.

[115] Behrooz Parhami. Voting algorithms. *IEEE Transactions on Reliability*, 43(4):617–629, IEEE, New York, NY, USA, 1994. doi:10.1109/24.370218.

[116] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, Taylor & Francis, Oxfordshire, 1900. doi:10.1080/14786440009463897.

[117] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, Microtome Publishing, Brookline, MA, USA, 2011. doi:10.48550/arXiv.1201.0490.

[118] Maja Perme, Mateja Blas, and Sandra Turk. Comparison of logistic regression and linear discriminant analysis: A simulation study. *Metodološki Zvezki - Advances in Methodology and Statistics*, 1:143–161, Faculty of Social Sciences of the University of Ljubljana, Ljubljana, 2004. doi:10.51936/ayrt6204.

[119] John R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Kluwer Academic Publishers, Norwell, MA, USA, 1986. doi:10.1023/A:1022643204877.

[120] Jon N. K. Rao and Alastair J. Scott. The analysis of categorical data from complex sample surveys: Chi-squared tests for goodness of fit and independence in two-way tables. *Journal of the American Statistical Association*, 76(374):221–230, Taylor & Francis, Oxfordshire, 1981. doi:10.1080/01621459.1981.10477633.

[121] Jon N. K. Rao and Alastair J. Scott. On Chi-Squared Tests for Multiway Contingency Tables with Cell Proportions Estimated from Survey Data. *The Annals of Statistics*, 12(1):46 – 60, Institute of Mathematical Statistics, Beachwood, OH, USA, 1984. doi:10.1214/aos/1176346391.

[122] Alexander Robitzsch. Why ordinal variables can (almost) always be treated as continuous variables: Clarifying assumptions of robust continuous and ordinal factor analysis estimation methods. *Frontiers in Education*, 5, Frontiers Media SA, Lausanne, 2020. doi:10.3389/feduc.2020.589965.

[123] Reinhard Roßner. Diskriminanzanalyse. In *Statistische Methoden II: Mehrvariable Methoden und Datenverarbeitung*, Lecture Notes in Economics and Mathematical Systems, pages 14–16. Springer, Berlin, Heidelberg, 1970. doi:10.1007/978-3-642-88253-1_4.

[124] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, Elsevier, Amsterdam, 1987. doi:10.1016/0377-0427(87)90125-7.

[125] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, [Oxford University Press, Biometrika Trust], Oxford, 1976. doi:10.1093/biomet/63.3.581.

[126] Donald B. Rubin. Introduction. In *Multiple Imputation for Nonresponse in Surveys*, chapter 1, pages 1–26. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1987. doi:10.1002/9780470316696.ch1.

[127] George P. Rédei. Bonferroni correction. In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 3rd edition, pages 227–227. Springer Netherlands, Dordrecht, 2008. doi:10.1007/978-1-4020-6754-9_1966.

[128] George P. Rédei. Distance matrix. In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 3rd edition, pages 517–517. Springer Netherlands, Dordrecht, 2008. doi:10.1007/978-1-4020-6754-9_4572.

[129] George P. Rédei. Homoscedasticity. In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 3rd edition, pages 902–902. Springer Netherlands, Dordrecht, 2008. doi:10.1007/978-1-4020-6754-9_7805.

[130] George P. Rédei. Logistic regression. In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 3rd edition, pages 1119–1119. Springer Netherlands, Dordrecht, 2008. doi:10.1007/978-1-4020-6754-9_9550.

[131] George P. Rédei. Product-moment correlation (pearson's product-moment correlation coefficient). In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 3rd edition, pages 1562–1562. Springer Netherlands, Dordrecht, 2008. doi:10.1007/978-1-4020-6754-9_13485.

[132] Bernhard Rüger. *Induktive Statistik. Einführung für Wirtschafts- und Sozialwissenschaftler*, 3rd edition. Oldenbourg Wissenschaftsverlag, Munich, 1995. doi:10.1515/9783486789560.

[133] Lothar Sachs. Einführung in die Statistik. In *Angewandte Statistik: Anwendung statistischer Methoden*, 11th edition, pages 11–15. Springer, Berlin, Heidelberg, 2004. doi:10.1007/978-3-662-05744-5_2.

[134] Claude Sammut and Geoffrey I. Webb. Logistic regression. In *Encyclopedia of Machine Learning*, pages 631–631. Springer US, Boston, MA, 2010. doi:10.1007/978-0-387-30164-8_493.

[135] Laurie A. Schintler. High dimensional data. In *Encyclopedia of Big Data*, pages 546–548. Springer International Publishing, Cham, 2022. doi:10.1007/978-3-319-32010-6_552.

[136] Kurt Schneider, Jil Klünder, Fabian Kortum, Lisa Handke, Julia Straube, and Simone Kauffeld. Positive affect through interactions in meetings: The role of proactive and supportive statements. *Journal of Systems and Software*, 143:59–70, Elsevier, Amsterdam, 2018. doi:10.1016/j.jss.2018.05.001.

[137] Lennart Schroth, Martin Obaidi, Alexander Specht, and Jil Klünder. On the potentials of realtime sentiment analysis on text-based communication in software projects. In *Human-Centered Software Engineering*, pages 90–109, Springer International Publishing, Cham, 2022. doi:10.1007/978-3-031-14785-2_6.

[138] Anoop D. Shah, Jonathan W. Bartlett, James Carpenter, Owen Nicholas, and Harry Hemingway. Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using MICE: A CALIBER Study. *American Journal of Epidemiology*, 179(6):764–774, [Oxford University Press, Johns Hopkins Bloomberg School of Public Health], Oxford, 2014. doi:10.1093/aje/kwt312.

[139] Samuel S. Shapiro and Martin B. Wilk. An analysis of variance test for normality (complete samples)†. *Biometrika*, 52(3-4):591–611, [Oxford University Press, Biometrika Trust], Oxford, 1965. doi:10.1093/biomet/52.3-4.591.

[140] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of Personality and Social Psychology*, 52(6):1061–1086, American Psychological Association, Washington, DC, USA, 1987. doi:10.1037/0022-3514.52.6.1061.

[141] Shashi Shekhar, Hui Xiong, and Xun Zhou. Root-mean-square error. In *Encyclopedia of GIS*, 2nd edition, pages 1794–1794. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-17885-1_101126.

[142] Jingyi Shen, Olga Baysal, and M. Omair Shafiq. Evaluating the performance of machine learning sentiment analysis algorithms in software engineering. In *2019 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 1023–1030, IEEE, New York, NY, USA, 2019. doi:10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00185.

[143] Karen L. Soeken and Patricia A. Prescott. Issues in the use of kappa to estimate reliability. *Medical Care*, 24(8):733–741, Lippincott Williams & Wilkins, Philadelphia, PA, USA, 1986. doi:10.1097/00005650-198608000-00008.

[144] Fengxi Song, Dayong Mei, and Hongfeng Li. Feature selection based on linear discriminant analysis. In *2010 International Conference on Intelligent System Design and Engineering Application*, volume 1, pages 746–749, IEEE, New York, NY, USA, 2010. doi:10.1109/ISDEA.2010.311.

[145] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, University of Illinois Press, Champaign, IL, USA, 1904. doi:10.2307/1422689.

[146] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*, pages 273–309. Springer, Berlin, Heidelberg, 2004. doi:10.1007/978-3-662-08968-2_16.

[147] Gail M. Sullivan and Anthony R. Artino Jr. Analyzing and Interpreting Data From Likert-Type Scales. *Journal of Graduate Medical Education*, 5(4):541–542, Accreditation Council for Graduate Medical Education, Chicago, IL, USA, 2013. doi:10.4300/JGME-5-4-18.

[148] Vladimir Temlyakov. Greedy algorithms. In *Encyclopedia of Applied and Computational Mathematics*, pages 611–614. Springer, Berlin, Heidelberg, 2015. doi:10.1007/978-3-540-70529-1_295.

[149] Gerald Teschl and Susanne Teschl. Eigenwerte und Eigenvektoren. In *Mathematik für Informatiker: Band 1: Diskrete Mathematik und Lineare Algebra*, 4th edition, pages 389–414. Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-37972-7_14.

[150] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2010. doi:10.1002/asi.21416.

[151] John A. Trangenstein. Eigenvalues and eigenvectors. In *Scientific Computing : Eigenvalues and Optimization*, volume 2 of *Texts in Computational Science and Engineering*, pages 1–201. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-69107-7_1.

[152] John W. Tukey. Exploratory data analysis. In *The Concise Encyclopedia of Statistics*, pages 192–194. Springer, New York, NY, 1977. doi:10.1007/978-0-387-32833-1_136.

[153] Tien Rahayu Tulili, Andrea Capiluppi, and Ayushi Rastogi. Burnout in software engineering: A systematic mapping study. *Information and Software Technology*, 155:107116, Elsevier, Amsterdam, 2022. doi:10.1016/j.infsof.2022.107116.

[154] Gias Uddin and Foutse Khomh. Automatic mining of opinions expressed about apis in stack overflow. *IEEE Transactions on Software Engineering*, 47(3):522–559, IEEE, New York, NY, USA, 2021. doi:10.1109/TSE.2019.2900245.

[155] Stef van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall/CRC, New York, NY, USA, 2012. doi:10.1201/b11826.

[156] Stef van Buuren, Jacob P. L. Brand, Catharina G. M. Groothuis-Oudshoorn, and Donald B. Rubin. Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*, 76(12):1049–1064, Taylor & Francis, Oxfordshire, 2006. doi:10.1080/10629360600810434.

[157] Stef van Buuren and Catharina G. M. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, Foundation for Open Access Statistics, Los Angeles, CA, USA, 2011. doi:10.18637/jss.v045.i03.

[158] Vijaya, Shweta Sharma, and Neha Batra. Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 568–573, IEEE, New York, NY, USA, 2019. doi:10.1109/COMITCon.2019.8862232.

[159] Michail Vlachos. Dimensionality reduction. In *Encyclopedia of Machine Learning*, pages 274–279. Springer US, Boston, MA, 2010. doi:10.1007/978-0-387-30164-8_216.

[160] Marlis von der Hude. Korrelation. In *Predictive Analytics und Data Mining : Eine Einführung mit R*, pages 33–40. Springer Fachmedien, Wiesbaden, 2020. doi:10.1007/978-3-658-30153-8_3.

[161] Abraham Wald. Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, 54:426–482, American Mathematical Society, Providence, RI, USA, 1943. doi:10.1090/S0002-9947-1943-0012401-3.

[162] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, Taylor & Francis, Oxfordshire, 1963. doi:10.1080/01621459.1963.10500845.

[163] Michael L. Waskom. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021, Open Source Initiative, Palo Alto, CA, USA, 2021. doi:10.21105/joss.03021.

[164] Bernard L. Welch. The generalisation of student's problems when several different population variances are involved. *Biometrika*, 34(1-2):28–35, [Oxford University Press, Biometrika Trust], Oxford, 1947. doi:10.1093/biomet/34.1-2.28.

[165] Niklaus E. Wirth. A plea for lean software. *Computer*, 28(2):64–68, IEEE, New York, NY, USA, 1995. doi:10.1109/2.348001.

[166] Junfang Wu, Chunyang Ye, and Hui Zhou. Bert for sentiment classification in software engineering. In *2021 International Conference on Service Science (ICSS)*, pages 115–121, IEEE, New York, NY, USA, 2021. doi:10.1109/ICSS53362.2021.00026.

[167] Petros Xanthopoulos, Panos M. Pardalos, and Theodore B. Trafalis. Linear discriminant analysis. In *Robust Data Mining*, pages 27–33. Springer, New York, NY, USA, 2013. doi:10.1007/978-1-4419-9878-1_4.

[168] Umer Zaman, Zulaikha Jabbar, Shahid Nawaz, and Mazhar Abbas. Understanding the soft side of software projects: An empirical study on the interactive effects of social skills and political skills on complexity – performance relationship. *International Journal of Project Management*, 37(3):444–460, Elsevier, Amsterdam, 2019. doi:10.1016/j.ijproman.2019.01.015.

[169] Ting Zhang, Bowen Xu, Ferdian Thung, Stefanus Agus Haryono, David Lo, and Lingxiao Jiang. Sentiment analysis for software engineering: How far can pre-trained transformer models go? In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 70–80, IEEE, New York, NY, USA, 2020. doi:10.1109/ICSME46990.2020.00017.

[170] Xinhua Zhang. Covariance matrix. In *Encyclopedia of Machine Learning and Data Mining*, 2nd edition, pages 290–293. Springer US, Boston, MA, 2017. doi:10.1007/978-1-4899-7687-1_57.

[171] Zhongheng Zhang. Missing data imputation: focusing on single imputation. *Annals of Translational Medicine*, 4(1), AME Publishing Company, Hong Kong, 2016. doi:10.3978/j.issn.2305-5839.2015.12.38.

Abbildungsverzeichnis

Tabellenverzeichnis