HAW HAMBURG

BACHELORARBEIT

Systematische Analyse von unbeabsichtigtem Bias in der Textklassifikation

vorgelegt am 8. September 2023

Kirsten Grahl

Erstprüferin: Prof. Dr. Larissa Putzar
Zweitprüferin: Prof. Dr.-Ing. Sabine Schumann

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
22081 Hamburg

Zusammenfassung

Bei der Klassifikation von Texten kann ein unbeabsichtigter Bias auftreten, welcher die Genauigkeit der Ergebnisse dieser Klassifikation für bestimmte Personengruppen negativ beeinflusst. In dieser Arbeit werden Kommentare mithilfe verschiedener Verfahren für die Klassifikation von Texten auf ihre Toxizität bewertet und in „toxisch“ oder „nicht toxisch“ eingeordnet. Das Ziel dieser Arbeit ist es, die Ergebnisse der verschiedenen Verfahren für die Klassifikation auf einen Bias hinsichtlich des Geschlechts, der sexuellen Orientierung, der Religion, der Herkunft oder Ethnie und der Behinderung zu untersuchen und dabei die Verfahren in Bezug auf den Bias miteinander zu vergleichen. Dazu werden die Verfahren Naive Bayes, Entscheidungsbaum, Random Forest, logistische Regression und ein Convolutional Neural Network (CNN) implementiert. Für die einzelnen Verfahren wird jeweils ein Modell trainiert. Hierbei wird ein Datensatz verwendet, der auf das Vorkommen der Personengruppen im Text des Kommentars bewertet wurde.

Abstract

Text classification can exhibit an unintended bias that negatively affects the accuracy of classification results for certain groups of people. This thesis evaluates the toxicity of comments using different text classification methods to classify them as ”toxic” or ”non-toxic”. The goal of this thesis is to examine the results of the different classification methods for the presence of bias in relation to gender, sexual orientation, religion, race or ethnicity and disability and to compare these methods with each other in terms of bias. For this purpose, the methods Naive Bayes, Decision Tree, Random Forest, Logistic Regression and a Convolutional Neural Network (CNN) are implemented and a model is trained for each method. For the training of the models, a data set is used that was rated on the mention of the groups of people in the comment text.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Soziale Medienplattformen haben in den letzten Jahren durch einen Zuwachs an Internetnutzer*innen ebenfalls einen Zuwachs an Nutzer*innen verzeichnen können. Kommentarbereiche auf diesen Plattformen bieten einen Raum zum Äußern und zum Diskutieren von Meinungen. Jedoch haben mit der Zunahme an Nutzer*innen auf diesen Plattformen ebenfalls die Anzahl an Hasskommentaren und andere Missbräuche dieses Raums zugenommen. Diese toxischen Kommentare stören oder verhindern respektvolle Diskussionen und verleitet andere Nutzer*innen dazu, die Diskussion zu verlassen. Deswegen kann deren Entfernung im Interesse der Plattformen liegen. Um diesen Vorgang zu erleichtern, kann der Einsatz eines Modells zur automatischen Erkennung von toxischen Kommentaren sinnvoll sein. Die eingesetzten Modelle können jedoch einen unbeabsichtigten Bias beinhalten, der Kommentare zu bestimmte Personengruppen unfair einordnet und diese Personengruppen dadurch benachteiligt. Diese Arbeit beschäftigt sich mit der Untersuchung eines solchen Bias in den Modellen. Das Ziel ist es, Modelle mit verschiedenen Verfahren zu trainieren, die Ergebnisse auf einen Bias zu untersuchen und miteinander zu vergleichen.

1.1 Motivation

Im Dezember 2022 gab es weltweit 5,544 Milliarden Internetnutzer*innen, was etwa 69 % der Weltbevölkerung entsprach. Zehn Jahre zuvor lag diese Zahl noch bei 2,497 Milliarden und damit 35,7 % der Weltbevölkerung (Internet World Stats, 2023). Mit der Zunahme an Internetnutzer*innen hat auch die Zahl der Nutzer*innen von sozialen Medien in den letzten Jahren zugenommen (Ortiz-Ospina, 2019). Im April 2023 gab es 4,8 Milliarden aktive Nutzeridentitäten auf den sozialen Medienplattformen, was fast 60 % der Weltbevölkerung entsprach (Kemp, 2023). Unter dieser Zahl können sich jedoch auch Dopplungen und „falsche“ Accounts befinden, weshalb sie möglicherweise nicht die tatsächliche Zahl der Nutzer*innen repräsentiert. Soziale Medienplattformen wie Facebook, YouTube, Twitter und Reddit bieten durch ihre Kommentarbereiche einen wichtigen Raum für den Austausch von Meinungen und stellen einen Platz für Diskussionen zu aktuellen Themen. Auch andere Websites bieten ihren Nutzer*innen einen Kommentarbereiche zum Diskutieren an, wie beispielsweise einige Nachrichtenportale und Blogs (Risch & Krestel, 2020).

Jeden Tag wächst die Menge der veröffentlichten Kommentare. Allein auf Reddit wurden im Jahr 2022 bis zum 20. November über 430 Millionen Beiträge erstellt und über 2,5 Milliarden Kommentare verfasst (Reddit Inc., 2022). Das ergibt im Durchschnitt über 7,7 Millionen Kommentare am Tag, über 321,5 Tausend in einer Stunde und über 5,3 Tausend Kommentare in einer Minute. Mit dieser Menge an Kommentaren hat ebenfalls die Menge an Hasskommentaren, Trollen, Auseinandersetzungen und anderen Missbräuchen dieses Raums auf den Plattformen zugenommen (Ghosh et al., 2021). Nach einem Bericht vom Pew Research Center im Jahr 2021 wurden 41 % der Amerikaner*innen schon einmal online belästigt. Zudem hat ein wachsender Anteil von mittlerweile 25 % der Amerikaner*innen schwerwiegende Formen der Belästigung erlebt, wie körperliche Drohungen, Stalking, sexuelle Belästigung und anhaltende Belästigung (Vogels, 2021).

Respektlose, unangemessene und unhöfliche Kommentare werden als toxische Kommentare bezeichnet. Sie stören oder verhindern respektvolle Diskussionen und verleiten andere Nutzer*innen dazu, die Diskussion oder die Plattform zu verlassen. Die Diskussionen sollten dementsprechend von der jeweiligen Plattform effektiv und positiv gehalten werden. Nicht nur deshalb kann die Entfernung von toxischen Kommentaren sinnvoll sein. Auch rechtliche Gründe können die Plattformen dazu zwingen, Maßnahmen gegen derartige Kommentare zu ergreifen. Einige Kommentare sind dahingegen legal, können aber durch die Nutzungsbedingungen oder Diskussionsleitlinien der Plattform verboten sein (Risch & Krestel, 2020).

Aufgrund der großen Menge an Kommentaren, kann diese Aufgabe nicht allein von menschlicher Hand durchgeführt werden. Der Einsatz von Systemen zur automatischen Erkennung von toxischen Kommentaren kann diesen Prozess unterstützen. Bei einer automatischen Entfernung von toxischen Kommentaren ist es besonders wichtig eine möglichst gute Erkennung von Toxizität zu erreichen, um nur die wirklich toxischen Kommentare zu entfernen. Es ist ein schmaler Grat zwischen Zensur und Meinungsfreiheit (Ullmann & Tomalin, 2020).

Modelle für eine solche Klassifikation von Kommentaren in „toxisch“ und „nicht toxisch“ können jedoch einen unbeabsichtigten Bias beinhalten. Dieser verschlechtert die Genauigkeit der Modelle für bestimmte Personengruppen, die im Kommentar erwähnt werden. Die Ursache für den Bias kann beispielsweise in den Trainingsdaten, der Modellarchitektur oder der Worteinbettung liegen (Poria et al., 2020). Durch diesen Bias kann es dazu kommen, dass ein nicht toxischer Kommentar wie ”I am a gay man” eine unverhältnismäßig hohe Toxizitätsbewertung erhält. Deswegen sollte ein Ziel bei der Erstellung von solchen Modellen sein, diesen Bias zu erkennen und zu beseitigen (Dixon et al., 2018). Nur auf diese Weise können gleiche Ergebnisse unabhängig von Geschlecht, Religion, Herkunft, sexueller Orientierung, Behinderung oder anderen Merkmalen erzielt werden. Damit wird sichergestellt, dass Kommentare, die beispielsweise „gay“ enthalten, nicht häufiger gelöscht werden. Deswegen ist es wichtig diesen unbeabsichtigten Bias in den Modellen herauszuarbeiten, um anschließend Gegenmaßnahmen ergreifen zu können.

1.2 Ziel der Arbeit

Diese Arbeit beschäftigt sich mit der Untersuchung auf einen solchen unbeabsichtigten Bias in Modellen, die mit verschiedenen Verfahren trainiert wurden. Sie werden auf einen Bias bezüglich des Geschlechts, der sexuellen Orientierung, der Religion, der *Race*[1] oder Ethnie und der Behinderung geprüft. Das Ziel ist es, die Modelle der verschiedenen Verfahren auf einen Bias zu untersuchen und miteinander zu vergleichen. Die Untersuchung auf einen Bias ist wichtig für spätere Arbeiten, die sich anschließend mit der Reduzierung des Bias beschäftigen können. Auf diese Weise können in Zukunft die Modelle für möglichst viele Personengruppen gerechter gestaltet werden. Kommentare zu bestimmten Personengruppen sollten nicht als toxischer bewertet werden als Kommentare zu anderen Personengruppen.

1.3 Struktur der Arbeit

Zunächst werden in Kapitel 2 die erforderlichen Grundlagen für die Textklassifikation erläutert. Dieses Kapitel gibt eine Einführung in die künstliche Intelligenz, das maschinelle Lernen, das Deep Learning, das Natural Language Processing (NLP) und die Textklassifikation. Zudem werden die Verfahren vorgestellt, mit denen die Modelle trainiert werden. Anschließend wird in Kapitel 3 der verwendete Datensatz vorgestellt, welcher einen Trainings- und einen Testdatensatz für die Umsetzung der Textklassifikation mit den verschiedenen Verfahren beinhaltet. In Kapitel 4 werden Methoden zur Modellbewertung vorgestellt, mit denen die Leistungen der Modelle bewertet und anschließend verglichen werden. Im Anschluss erfolgt in Kapitel 5 die Implementierung der Textklassifikation. Die Resultate werden in Kapitel 6 vorgestellt. Die Arbeit schließt mit einem Fazit in Kapitel 7 ab.

[1] In der vorliegenden Arbeit wird der im englischsprachigen Raum gebräuchliche Begriff *Race* anstelle des umstrittenen Begriffs „Rasse“ verwendet. Der Begriff *Race* bezieht sich auf die Beschreibung einer Gruppe von Menschen, die physische Merkmalen gemeinsam haben, welche bei Menschen mit gemeinsamer Abstammung als üblich angesehen werden. Diese Gruppe teilt sich möglicherweise einen gemeinsamen kulturellen, geografischen, sprachlichen oder religiösen Hintergrund (Merriam-Webster, 2023).

2 Grundlagen

Für die Untersuchung von unbeabsichtigtem Bias in der Textklassifikation werden zunächst Modelle mit verschiedenen Verfahren trainiert, mit denen anschließend Kommentare hinsichtlich ihrer Toxizität klassifiziert werden können. Die Ergebnisse dieser Klassifikation können danach auf unbeabsichtigten Bias untersucht werden. Dieses Kapitel befasst sich mit den Grundlagen, die zur Erstellung der Modelle zur Erkennung von Toxizität erforderlich sind. Zunächst wird eine Einführung in die künstliche Intelligenz und ein Teilgebiet der künstlichen Intelligenz, das maschinelle Lernen, gegeben. Danach wird eine Unterkategorie des maschinellen Lernens vorgestellt: das Deep Learning. Anschließend wird das Natural Language Processing (NLP) vorgestellt sowie das Bag-of-Words-Modell und Word Embeddings. Daran anschließend wird ein Überblick über die Textklassifikation gegeben. Zum Schluss werden die in dieser Arbeit verwendeten Verfahren für die Klassifikation vorgestellt: Naive Bayes, Entscheidungsbaum, Random Forest, logistische Regression und Convolutional Neural Network (CNN).

2.1 Künstliche Intelligenz

Bei der *künstlichen Intelligenz* (KI, engl. Artificial Intelligence) handelt „es sich um ein intelligentes System oder Wesen (...), das künstlich erschaffen wurde“ (Mueller & Massaron, 2019/2020, S. 30). Den Begriff der Intelligenz zu definieren ist nicht einfach. Jedoch lassen sich bestimmte Funktionen und Fähigkeiten definieren, die Intelligenz voraussetzen. Dies sind die Aufnahme neuer Informationen, das Ziehen von Schlussfolgerungen aus diesen, das Verstehen der Bedeutung der ausgewerteten Informationen, die Überprüfung der Gültigkeit der Informationen anhand belegbarer Quellen, die Vorhersage von Zusammenhängen auf der Grundlage der überprüften Informationen und die Beurteilung von Situationen mit Hilfe der gefundenen Zusammenhänge. Demnach geht es bei der Intelligenz um bestimmte Denkprozesse. Ein Computersystem kann versuchen, diese Denkprozesse zu imitieren. Durch automatische mathematische Prozesse kann ein Computer Daten manipulieren und auswerten, aber nicht wirklich verstehen oder begreifen. Insofern sind KI-Systeme nicht zu echter Intelligenz fähig, sondern versuchen diese möglichst gut nachzuahmen (Mueller & Massaron, 2019/2020).

2.2 Machine Learning

Das *maschinelle Lernen* (ML, engl. Machine Learning) ist ein Teilgebiet der künstlichen Intelligenz. Das Machine Learning beschäftigt sich mit der Entwicklung von selbstlernenden Algorithmen. Diese gewinnen Erkenntnisse aus Daten, um hierdurch Vorhersagen treffen zu können. Durch das Machine Learning wird das Eingreifen von Menschen für die Erstellung von Modellen anhand der Analyse von großen Datenmengen immer unwichtiger, denn die Verfahren zum Machine Learning stellen eine Alternative zur Erfassung des Wissens dar, welches in den Daten enthalten ist. Diese Art der Wissenserfassung verbessert zudem die Entscheidungsfindung und die Aussagekraft von Vorhersagemodellen.

Beim Machine Learning werden drei Kategorien betrachtet: überwachtes Lernen, unüberwachtes Lernen und Reinforcement Learning. Beim überwachten Lernen kann mithilfe von Trainingsdaten mit bereits zugeteilten Labels ein Modell erstellt werden. Mit diesem Modell ist es anschließend möglich eine Voraussage über Daten ohne ein solches Label zu treffen. Dies ist in Abbildung 2.1 dargestellt. Es wird als „überwachtes“ Lernen bezeichnet, weil die Trainingsdaten bereits mit den erwünschten Ausgabewerten gekennzeichnet sind. Die *Klassifikation* und die *Regression* sind Unterkategorien des überwachten Lernens. Bei der Klassifikation wird die Zugehörigkeit von neuen Daten in Klassen vorhergesagt. Die Klassenbezeichnungen sind eindeutig und ungeordnet. Bei der binären Klassifikation wird zwischen zwei Klassen unterschieden, beispielsweise für die Vorhersage der Toxizität zwischen den Klassen „toxisch“ und „nicht toxisch“. Jedoch können es auch mehr als zwei Klassenbezeichnungen sein. Bei der Regression werden stetige Ereignisse vorhergesagt, mithilfe von unabhängigen und erklärenden Variablen sowie einer stetigen Zielvariable. Um bei der Regression Ereignisse vorhersagen zu können, wird versucht eine Beziehung zwischen diesen Variablen zu finden.

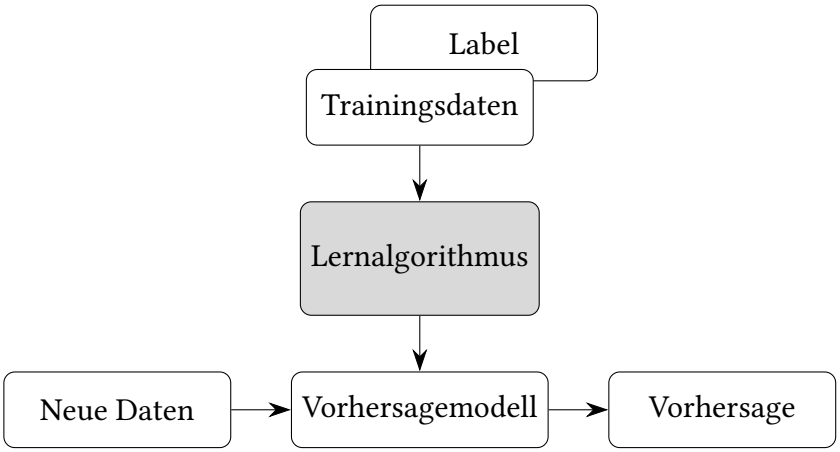

Abbildung 2.1: Vorhersagen treffen mit überwachtem Lernen [eigene Grafik nach (Raschka & Mirjalili, 2019/2021, S. 31)]

Das unüberwachte Lernen verwendet hingegen Daten ohne Label, denn die Verfahren des unüberwachten Lernens können durch die Erkundung der Datenstruktur Informationen aus den Daten entnehmen, ohne dabei Hinweise auf eine Zielvariable zu haben. Beim Reinforcement Learning ist das Ziel die Entwicklung eines Systems (Agenten), welches durch Interaktionen mit seiner Umgebung die Leistung verbessert. Dies wird mit einer Belohnungsfunktion realisiert (Raschka & Mirjalili, 2019/2021).

2.2.1 Herausforderungen

Die Auswahl des Lernalgorithmus und des Datensatzes sind für die Ergebnisse beim Machine Learning von großer Bedeutung. Für gute Ergebnisse wird eine ausreichende Menge an Trainingsdaten benötigt. Zudem sollten die Trainingsdaten repräsentativ für die zu verallgemeinernden neuen Situationen sein. Auch auf die Qualität der Trainingsdaten kommt es an. Viele Fehler, Ausreißer und Rauschen können die Erkennung von Mustern in den Daten erschweren. Deswegen ist das Säubern der Trainingsdaten ein wichtiger Schritt, in den viel Zeit investiert werden sollte. Außerdem sollten die Trainingsdaten für gute Ergebnisse genügend relevante und nicht zu viele irrelevante Merkmale enthalten (Géron, 2019/2020).

2.2.2 Over- und Underfitting

Beim Machine Learning kann es zudem zu einer übermäßigen Verallgemeinerung kommen, welche als *Overfitting* (Überanpassung) bezeichnet wird. Die Überanpassung ist ein häufiges Problem beim Machine Learning. Durch eine Überanpassung funktioniert das Modell gut für die Trainingsdaten, kann jedoch nicht gut verallgemeinern und hat dadurch mit unbekannten Daten (Testdaten) Schwierigkeiten. Die Überanpassung bei einem Modell wird auch als *große Varianz* bezeichnet. Sie wird oftmals durch zu viele Parameter verursacht, die zu einem für die verwendeten Daten zu komplexen Modell führen. Bei einem kleinen oder verrauschten Trainingsdatensatz, erkennt das Modell möglicherweise Muster im Rauschen selbst. Des Weiteren kann auch das Gegenteil von einer Überanpassung auftreten, welches als *Underfitting* (Unteranpassung) bezeichnet wird. Zu einer Unteranpassung kann es kommen, wenn das Modell nicht komplex genug ist und in den Trainingsdaten keine Muster finden kann. Dies führt ebenfalls zu einer schlechten Erkennung bei unbekannten Daten (Géron, 2019/2020; Raschka & Mirjalili, 2019/2021).

2.3 Deep Learning

Das *Deep Learning* ist eine Unterkategorie des maschinellen Lernens, welches wiederum ein Teilgebiet der künstliche Intelligenz ist. Dies wird in Abbildung 2.2 veranschaulicht. Systeme des Deep Learnings sind durch intensive Analysen und höhere Automatisierung gekennzeichnet. Beim maschinellen Lernen werden verschiedene Techniken eingesetzt, wie die statistische Analyse oder die Suche nach Analogien in den Daten. Beim Deep Learning wird jedoch nur eine Technik verwendet, bei der die Funktionsweise des menschlichen Gehirns nachgeahmt wird.

Die Verarbeitung der Daten erfolgt mit Rechnereinheiten, die als *Neuronen* bezeichnet werden. Diese Neuronen sind in *Schichten* angeordnet. Das übergeordnete Konstrukt wird als *neuronales Netz* bezeichnet. Abhängig vom neuronalen Netz kann die Anzahl der Schichten sehr groß werden. Das Convolutional Neural Network (CNN) oder das rekurrente neuronale Netz (RNN) sind bekannte Vertreter für das Deep Learning (Mueller & Massaron, 2019/2020). Diese Netze werden für komplexe Aufgabenstellungen verwendet, wie für Bild- oder Spracherkennung. Bekannte Technologieunternehmen wie Facebook und Google verwenden Deep-Learning-Algorithmen für ihre Produkte wie Facebooks Gesichtserkennung DeepFace, Googles Bildersuche und Google Translate (Raschka & Mirjalili, 2019/2021).

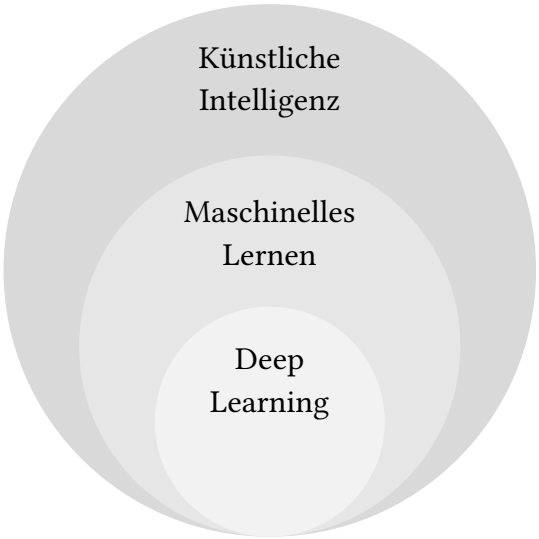

Abbildung 2.2: Deep Learning als Unterkategorie des maschinellen Lernens [eigene Grafik nach (Mueller & Massaron, 2019/2020, S. 30)]

2.4 Natural Language Processing

Beim *Natural Language Processing* (NLP) geht es um die Verarbeitung natürlicher Sprache in einem Computer. Vorliegende Textdaten sollen mit Algorithmen verarbeitet werden. Dafür müssen die Textdaten oftmals in numerische Form gebracht werden. Die Wörter müssen folglich so in Variablen umgewandelt werden, dass ein maschineller Lernalgorithmus sie als Eingabe akzeptiert. Viele Informationen liegen bereits im Zahlenformat vor, wie beispielsweise die Temperatur, die Einwohnerzahl einer Stadt oder das Einkommen einer Person. Bei Textdaten ist das Umwandeln der Daten in Zahlen schwierig. Die Wörter sind einerseits eigenständige Bedeutungsträger, stehen aber zudem in semantischen und grammatikalischen Relationen zueinander wie der Synonymie, der Antonymie oder der Hyperonymie. Um die Bedeutung eines einzelnen Wortes zu verstehen, muss meist die Bedeutung des restlichen verwendeten Vokabulars bekannt sein. Häufig ergibt sich diese Bedeutung aus dem Kontext und ist von der Reihenfolge der Wörter im Text anhängig. Erschwerend kommt hinzu, dass die Sprache einem ständigen Wandel unterliegt und somit sich auch die Bedeutungen der Wörter im Laufe der Zeit ändern.

Diese zusätzlichen Informationen lassen sich schwer in Zahlenwerten ausdrücken. Zumindest lässt sich eine solche Darstellung nicht mit einfachen Encodierungsverfahren erreichen. Aus diesem Grund setzen neuere Anwendungen spezielle Algorithmen ein, welche die Bedeutung und die Relationen der Wörter in einem Raum ausdrücken können. Dazu zählen beispielsweise Methoden wie die Worteinbettung oder rekurrente Netze. Bei diesen handelt es sich jeweils um ein eigenes maschinelles Lernverfahren, welches nur in einem System des Deep Learnings funktioniert. Für Lernalgorithmen wie die logistische Regression muss ein einfaches Encodierungsverfahren verwendet werden, bei dem bestimmte Informationen verloren gehen. Dafür werden einfache Vektorisierungsverfahren wie der Bag-of-Words-Ansatz verwendet (Hirschle, 2022; Mueller & Massaron, 2019/2020).

2.4.1 Bag-of-Words

Für die Verarbeitung von kategorialen Daten wie von Texten und von Wörtern, werden diese wie bereits in der Vorverarbeitung in numerische Daten umgewandelt. Auf diese Weise können Lernalgorithmen die Daten verarbeiten. *Bag-of-Words* (BoW) ist ein Modell, welches es erlaubt Text als numerischen Merkmalsvektor zu repräsentieren (Raschka & Mirjalili, 2019/2021). Für das Bag-of-Words-Modell werden alle Wörter in einen Gesamtkorpus tokenisiert. Bei einer solchen *Tokenisierung* werden Texte in ihre Wörter oder Token zerlegt, da diese als kleinste Analyseeinheit dienen. Von jedem Wort wird nur ein Exemplar behalten. Folglich werden die Duplikate entfernt. Daraus ergibt sich das sogenannte *Vokabular* (Hirschle, 2022).

Für das Bag-of-Words-Modell wird dementsprechend ein Vokabular mit eindeutigen Tokens für alle Textdokumente erstellt. Anschließend wird für jedes Textdokument ein Merkmalsvektor erstellt, in dem die Anzahl der Vorkommen jedes einzelnen Wortes aus dem Vokabular im jeweiligen Dokument enthalten sind. Wörter die nicht im Text vorkommen, werden mit einer Null codiert. Da ein Textdokument meist nicht alle Wörter des Vokabulars umfasst, enthalten die Merkmalsvektoren vorwiegend Nullen und werden aus diesem Grund als *dünn besetzt* bezeichnet (Raschka & Mirjalili, 2019/2021).

Ein Beispiel für das Bag-of-Words-Modell ergibt sich aus den drei kurzen Textdokumenten nach Hirschle (2022), die in Codeblock 2.1 dargestellt sind:

Codeblock 2.1: Textdokumente [nach (Hirschle, 2022, S. 43)]

```
korpus = ['Es ist sonnig', 'Es regnet', 'Die Sonne scheint']
```

Anschließend werden die Sätze tokenisiert. Dabei werden zudem alle Wörter in Kleinbuchstaben umgewandelt. Mit dieser Umwandlung wird versucht das Vokabular möglichst klein zu halten. Ein Problem bei BoW-Modellen ist, dass jedes Wort des Vokabulars für jedes Dokument encodiert werden muss, wodurch sehr große Matrizen entstehen, deren Größe ein Problem bei der Datenhaltung und beim Anlernen der Algorithmen darstellt. Deswegen wird versucht durch verschiedene Maßnahmen das Vokabular relativ klein zu halten. Zusätzlich zur häufigen Umwandlung aller Wörter in Kleinbuchstaben werden auch irrelevante Wörter aus dem Korpus entfernt. Diese werden als *Stopwords* bezeichnet und bestehen aus Konjunktionen, Pronomen, Artikeln und Modalverben. Zusätzlich werden häufig statt der kompletten Wörter nur die Wortstämme oder die Grundformen der Wörter verwendet. Im folgenden Codeblock 2.2 fand davon lediglich eine Umwandlung in Kleinbuchstaben statt, keine Entfernung der Stopwords oder eine Umwandlung der Wörter in die Wortstämme:

Codeblock 2.2: Tokenisierte Sätze [nach (Hirschle, 2022, S. 44)]

```
['es', 'ist', 'sonnig'] ['es', 'regnet'] ['die', 'sonne', 'scheint']
```

Daraus kann die Dokument-Wort-Matrix erzeugt werden, in welcher sich die erstellten Merkmalsvektoren befinden. Diese Dokument-Wort-Matrix ist in Codeblock 2.3 dargestellt:

Codeblock 2.3: Dokument-Wort-Matrix [nach (Hirschle, 2022, S. 44)]

```
doc-no. die es  ist regnet  scheint sonne   sonnig
1       0   1   1   0       0       0       1
2       0   1   0   1       0       0       0
3       1   0   0   0       1       1       0
```

2.4.2 Word Embedding

Wie auch das Bag-of-Words-Modell handelt es sich bei *Word Embeddings* (Worteinbettungen) um einen Ansatz zur Darstellung von Textdaten als Zahlenwerte. Beim BoW-Modell bleiben jedoch die Beziehungen zwischen den Wörtern nicht erhalten. Sowohl die semantischen Relationen wie die Synonymie, die Antonymie oder die Hyperonymie als auch die grammatikalischen Übereinstimmungen und Unterschiede wie die Wortart, das Geschlecht, der Modus oder der Tempus gehen verloren. Das BoW-Modell macht zudem das Anlernen schwieriger, weil die Bedeutung jedes Wortes anhand der Trainingsdaten erlernt werden muss und die Beziehung zwischen Synonymen wie „gut“ und „exzellent“ nicht bekannt ist.

Word Embeddings beheben dieses Probleme. Bei Word Embeddings sollen hierfür die Bedeutungen der Wörtern und die Relationen zwischen den Wörtern durch Vektoren dargestellt werden. Jedes Wort wird sozusagen in einen mehrdimensionalen Raum projiziert. Ein Wort wird auf diese Weise beispielsweise durch 30 stetige Variablen dargestellt. Jede dieser Variablen kann als eine semantische oder grammatikalische Dimension betrachtet werden, auf der das Wort eine gewisse Position einnimmt. Die Bedeutung eines Wortes ergibt sich als Konfiguration aller 30 Variablen. Die anderen Wörter können, da die Werte der Variablen stetig sind, auf einigen oder allen Dimensionen ähnliche oder sehr weit voneinander entfernte Positionen annehmen. Auf diese Weise kann semantische oder grammatikalische Nähe oder Entfernung ausgedrückt werden. Word Embeddings werden, anders als der BoW-Ansatz, mit der Hilfe eines statistischen Verfahrens und mit Trainingsdaten gelernt (Hirschle, 2022).

Word Embeddings werden in Deep Learning-Systemen eingesetzt, wie CNNs oder RNNs. Für solche neuronalen Netze sind *spärliche Datenmengen* ein Problem, also Datenmengen mit vielen Nullen. Das ist beim BoW-Ansatz jedoch der Fall. Durch spärliche Datenmengen wird die Berechnung einer guter Lösung mit einem neuronalen Netz sehr schwer. Zudem besitzen spärliche Datenmengen häufig mehr Spalten, wodurch in der Eingabeschicht sehr viel mehr Gewichte benötigt werden. Word Embeddings lösen dieses Problem, denn mit ihnen können dünn besetzte Matrizen in dicht besetzte Matrizen umgewandelt werden. Die Anzahl der Spalten in der Matrix kann somit erheblich reduziert werden. Sie kann von mehreren hunderttausend Matrizen auf wenige hundert reduziert werden.

Die Qualität der Word Embeddings hängt sehr von der Qualität der Trainingsdaten ab. Zum Training von Word Embeddings werden große Textmengen benötigt, die häufig automatisch und ohne vorherige Überprüfung aus dem Internet zusammengesammelt werden. Dies kann zu falschen Assoziationen zwischen Wörtern und einem unbeabsichtigten Bias führen (Mueller & Massaron, 2019/2020). Vorurteile der Gesellschaft finden sich dadurch ebenfalls in den Daten wieder. Word Embeddings lernen zum Beispiel, dass Mann zu König gehört wie Frau zu Königin. Zugleich lernen sie aber auch, dass Mann zu Doktor gehört wie Frau zu Krankenschwester, wobei es sich um ein sexistisches Vorurteil handelt (Géron, 2019/2020).

Word Embedding-Ansätze sind zum Beispiel *fastText*, *Vec2Word* und *GloVe* (Global Vectors). Von diesen Ansätzen wird in dieser Arbeit GloVe verwendet. Bei GloVe handelt es sich um ein Open-Source-Projekt der Stanford University, das eine ähnliche Vorgehensweise wie Methoden der statistischen Linguistik verwendet. Dieser Ansatz verwendet Statistiken über das gemeinsame Auftreten bestimmter Wörter aus einem Korpus und wandelt die daraus resultierende dünn besetzte Matrix mithilfe der Matrixfaktorisierung in eine dichte Matrix um (Mueller & Massaron, 2019/2020).

2.5 Textklassifikation

Das in Abschnitt 2.4 vorgestellte Natural Language Processing ermöglicht die Verarbeitung von natürlicher Sprache in einem Computer. Durch die Umwandlung von Textdaten in numerische Form können Algorithmen mit natürlicher Sprache arbeiten. Die Textklassifikation ist eine typische NLP-Anwendung. Bei der Textklassifikation werden Textdaten geeigneten Kategorien zugeordnet. Viele aktuelle Modelle zur Textklassifikation konzentrieren sich auf die Erfassung von mehr Kontextinformationen und der korrekten Wortreihenfolge. Dazu dienen die in Abschnitt 2.4.2 eingeführten Word Embeddings. Mit der Entwicklung des Deep Learnings wurden neuronale Modelle in den Bereich der Textklassifikation eingeführt, die die Fähigkeit zum Lernen von Textrepräsentationen besitzen.

Textklassifikation kann auf verschiedenen Ebenen erfolgen, wie auf der Dokumentebene, der Phrasenebene und der Satzebene. Insbesondere die Klassifikation auf der Dokumentebene kann sehr ungenau sein, weil sie davon ausgeht, dass das gesamte Dokument einer Kategorie zuzuordnen ist, auch wenn das Dokument Textabschnitte zu verschiedenen Kategorien enthält. Eine typische Aufgaben von der Textklassifikation ist die Stimmungsklassifikation (engl. sentiment classification). Bei dieser ist es das Ziel, die gefühlsmäßige Meinungsrichtung eines Textes zu klassifizieren, zum Beispiel in „positiv“ oder „negativ“ (Liu et al., 2020; Poria et al., 2020). Ähnlich verhält es sich mit der in dieser Arbeit vorgenommenen Klassifikation von Kommentaren hinsichtlich ihrer Toxizität, bei der die Kommentare in „toxisch“ und „nicht-toxisch“ klassifiziert werden.

2.5.1 Erkennung von Toxizität

Risch und Krestel (2020) unterscheiden zwischen fünf verschiedenen Arten von Toxizität: obszöne Sprache oder Profanität, Beleidigungen, Drohungen, Hassreden oder Hass gegen die Identität einer Person und anderweitige Toxizität. Bei der anderweitigen Toxizität handelt es sich um Kommentare, die nicht in die ersten vier Kategorien fallen, aber allgemein als toxisch gelten und eine Person wahrscheinlich dazu veranlassen eine Diskussion zu verlassen.

Für die sozialen Medien und andere Internetforen wird es immer wichtiger toxische Kommentare unter der großen Anzahl von täglich geposteten Inhalten zu identifizieren und zu entfernen. Herkömmliche Algorithmen verlassen sich darauf, dass toxische Inhalte von Nutzer*innen gemeldet werden. Jedoch dauert es auf diese Weise relativ lange bis die toxischen Inhalte gelöscht und eventuell notwendige Maßnahmen gegen die entsprechenden Nutzer*innen ergriffen werden können. In dieser Zeit können sich die toxischen Inhalte auf den Plattformen verbreiten und zu weiteren Auseinandersetzungen führen. Deswegen kann es sinnvoll sein, die Inhalte vor der Veröffentlichung auf ihre Toxizität zu prüfen (Ghosh et al., 2021). Dies kann jedoch nicht allein von menschlicher Hand erfolgen, weshalb der Einsatz von Systemen zur automatischen Erkennung von toxischen Kommentaren sinnvoll ist. Wenn ein bestimmter Beitrag automatisch und zuverlässig als toxisch eingestuft wird, kann er vorübergehend unter Quarantäne gestellt werden (Ullmann & Tomalin, 2020). Ein derartiges System kann zudem in Kombination mit menschlichen Moderator*innen eingesetzt werden. Bei einer solchen halbautomatischen Moderation klassifiziert ein maschinelles Lernmodell die Kommentare in „toxisch“ und „nicht toxisch“. Die vermutlich angemessenen Kommentare werden veröffentlicht und die vermutlich toxischen Kommentare werden menschlichen Moderator*innen vorgelegt (Risch & Krestel, 2020).

2.5.2 Bias in der Textklassifikation

Bei der Textklassifikation gibt es verschiedene Herausforderungen, wie die korrekte Erkennung und Einordnung von Sarkasmus, Ironie, Abkürzungen, Umgangssprache und Emoticons (Poria et al., 2020). Des Weiteren können solche Systeme zur automatischen Erkennung von toxischen Kommentaren einen unbeabsichtigten Bias beinhalten, welcher die Ergebnisse verzerrt. Nach der Definition von Dixon et al. (2018) enthält ein Modell zur Textklassifikation einen unbeabsichtigten Bias, wenn es bei Kommentaren über bestimme Gruppen besser abschneidet als bei Kommentaren über andere Gruppen.

Nach Poria et al. (2020) gibt es drei Hauptquellen für einen Bias in Systemen zur Sentimentanalyse. Diese lassen sich auf andere Systeme zur Textklassifikation übertragen. Die erste Quelle sind Word Embeddings, welche häufig mit öffentlich zugänglichen Textquellen wie Wikipedia trainiert werden. Bei Wikipedia werden jedoch nur 15 % der Beiträge von Frauen verfasst, wodurch die Sichtweise von Frauen unterrepräsentiert ist. Die zweite Quelle ist die Modellarchitektur, bei der verwendete Metainformationen wie Geschlechtsidentifikatoren und Indikatoren für demografische Merkmale wie Alter, Rasse, Nationalität und geografische Merkmale die Ursache für den Bias sein können. Eigentlich sollen diese Variablen bei der Klassifizierung helfen und werden beispielsweise bei der Twitter-Sentimentanalyse verwendet. Jedoch kann durch die Konditionierung auf diese Variablen ein Bias entstehen. Die dritte Hauptquelle sind die Trainingsdaten, aus denen ein System auf verschiedene Weise den Bias

übernehmen kann. So können in den Trainingsdaten bestimmte Sentimentwörter gemeinsam mit einem spezifischem Geschlecht auftreten, zum Beispiel „Frau“ gemeinsam mit „böse“. Zudem können bestimmte Personengruppen in den Trainingsdaten über- oder unterrepräsentiert sein und bestimmte demografischen Merkmale können häufig einer bestimmten Kategorie angehören, wie etwa weibliche Personen der Kategorie „positive Stimmung“. Neben diesen drei Hauptquellen kann auch der Schreibstil eines Autors eine Quelle für Bias sein. Eine Person kann beispielsweise sehr starke Gefühlswörter verwenden, um eine positive Meinung auszudrücken, aber für eine negative Meinung lieber mildere Gefühlswörter. Dies kann sich je nach Herkunft und Geschlecht einer Person unterscheiden, wodurch die Beseitigung von Bias erschwert wird.

Es gibt keine einfache Lösung für den Umgang mit diesem unbeabsichtigten Bias in maschinellen Lernsystemen, der durch menschliche Voreingenommenheit in die Systeme mit einfließt. Es ist schwierig diesen Bias genau zu identifizieren und zu messen, selbst wenn man die Untersuchung auf bestimmte Personengruppen beschränkt, wie beispielsweise das Geschlecht oder die *Race* einer Person (Kiritchenko & Mohammad, 2018).

In den letzten Jahren gab es bereits einige Arbeiten zum Auffinden eines unbeabsichtigten Bias in Modellen zur Textklassifikation. Mit dem Equity Evaluation Corpus (EEC) von Kiritchenko und Mohammad (2018) wurden 219 Systeme zur Sentimentanalyse auf einen Bias bezüglich des Geschlechts und der *Race* untersucht. Der EEC wurde aus 8.640 englischen Sätzen zusammengestellt, die einen möglichen Bias in den Systemen aufzeigen sollen. Diese Sätze bestehen aus einfachen Satzvorlagen wie: *<Person>* feels *<emotional state word>*. Für *<Person>* werden weibliche und männliche afrikanisch-amerikanische und europäisch-amerikanische Vornamen eingesetzt, aber auch Wörter die sich allgemein auf Frauen und Männer beziehen wie „my daughter“ und „my son“. Das *<emotional state word>* ist nicht in jeder Satzvorlage enthalten. Dort werden Gefühlszustände mit unterschiedlichen Intensitäten eingesetzt. Mit dem EEC konnte auf diese Weise in den untersuchten Systemen ein Bias in Bezug auf das Geschlecht und die *Race* aufgezeigt werden. Mehr als 75 % der Systeme neigten dazu, Sätze, die ein Geschlecht / eine *Race* betreffen, mit höheren Intensitätswerten zu bewerten als Sätze, die das andere Geschlecht / die andere *Race* betreffen.

2.6 Verfahren

Für die Klassifikation von Texten werden in dieser Arbeit verschiedene Verfahren verwendet: Naive Bayes, Entscheidungsbaum, Random Forest, logistische Regression und Convolutional Neural Network (CNN). Mit diesen Verfahren wird jeweils ein Modell trainiert. Naive Bayes ist ein einfaches Verfahren, das auf der Bayesschen Formel basiert. Entscheidungsbäume

bieten eine besonders gute Interpretierbarkeit. Bei Random Forests werden mehrere Entscheidungsbäume miteinander kombiniert, um die Anfälligkeit für eine Überanpassung bei Entscheidungsbäumen zu reduzieren. Die logistische Regression ist eines der meistgebrauchten Klassifikationsverfahren, welches linear trennbare Klassen besitzt. Es handelt sich um ein Wahrscheinlichkeitsmodell. Das Convolutional Neural Network gehört in den Bereich des Deep Learnings.

2.6.1 Naive Bayes

Das erste in dieser Arbeit verwendete Verfahren ist *Naive Bayes*. Dabei handelt es sich um einen einfachen Algorithmus, der die *Bayessche Formel* nutzt, um Wahrscheinlichkeitsprobleme zu lösen. Das Ziel ist es, die wahrscheinlichste Klasse vorherzusagen. Beim Naive Bayes-Algorithmus wird davon ausgegangen, dass alle Attribute unabhängig voneinander sind. Der Algorithmus zählt, wie oft welcher Merkmalswert bei der jeweiligen Zielklasse auftritt. Auf diese Weise kann für jedes Merkmal eine empirische Wahrscheinlichkeitsverteilung, differenziert nach Klassen, erstellt werden. Wenn ein Element klassifiziert werden soll, so wird für jede Zielklasse die Wahrscheinlichkeit der jeweiligen Klasse mit den Wahrscheinlichkeiten der vorhandenen Merkmalswerte multipliziert und die Klasse mit dem höchsten Wert ausgewählt (Biemann et al., 2022; Cleve & Lämmel, 2020).

Ein Vorteil von Naive Bayes ist, dass das Trainieren sehr schnell und einfach ist. Dadurch eignet sich Naive Bayes auch für inkrementelles Training. Jedoch sind die Ergebnisse von Naive Bayes im Vergleich häufig schlechter als die von anderen Algorithmen. Das liegt einerseits daran, dass die Merkmale keine Gewichtung haben, sodass alle Merkmale zu gleichen Teilen in die Berechnung eingehen. Andererseits liegt es daran, dass bei Naive Bayes die Abhängigkeiten zwischen Merkmalen nicht berücksichtigt werden (Biemann et al., 2022).

Der Naive Bayes-Algorithmus basiert, wie bereits erwähnt, auf der Bayesschen Formel. Bei der Bayesschen Formel ist es erlaubt in Berechnungen mit bedingten Wahrscheinlichkeiten die abhängigen Ereignisse zu vertauschen. Die Bayesschen Formel ergibt sich wie folgt:

$$P(X \mid Y) = \frac{P(Y \mid X) \times P(X)}{P(Y)} \tag{2.1}$$

Dabei ist Y ein Ereignis mit $P(Y) > 0$ und $P(X \mid Y)$ die bedingte Wahrscheinlichkeit von X unter der Bedingung Y. Durch die Bayesschen Formel ist die Berechnung von $P(X \mid Y)$ mit $P(Y \mid X)$ möglich. Somit kann mit dem Satz von Bayes die umgekehrte bedingte Wahrscheinlichkeit berechnet werden (Cleve & Lämmel, 2020).

Es gibt mehrere bayessche Verfahren, die die Bayessche Formel nutzen um Wahrscheinlichkeitsprobleme zu lösen. Die Algorithmen dieser Gruppe können sowohl für die Klassifikation als auch für die Regression eingesetzt werden. Beispiele für bayessche Verfahren sind neben

dem Naive Bayes-Klassifikator, der Gaußsche Naive Bayes-Klassifikator, der multinomiale Naive Bayes-Klassifikator und Bayessche Netze (Mueller & Massaron, 2019/2020). In dieser Arbeit wird der Bernoulli Naive Bayes verwendet, ein Bayessches Netz ohne Abhängigkeiten zwischen den Wörtern und mit binären Wortmerkmalen (McCallum & Nigam, 1998).

2.6.2 Entscheidungsbaum

Bei *Entscheidungsbäumen* (engl. Decision Trees) handelt es sich um Machine Learning-Algorithmen, die sowohl für die Klassifikation als auch für die Regression geeignet sind (Géron, 2019/2020). Klassifikationsmodelle die auf Entscheidungsbäumen beruhen können besonders gut eingesetzt werden, wenn die Interpretierbarkeit von Bedeutung ist. Entscheidungsbäume sind eine sehr gute Möglichkeit, den Weg zur Vorhersage oder zur Entscheidung zu visualisieren. Dieser Weg ist mit Hilfe eines Wetter-Beispiels in Abbildung 2.3 dargestellt (Cleve & Lämmel, 2020; Raschka & Mirjalili, 2019/2021).

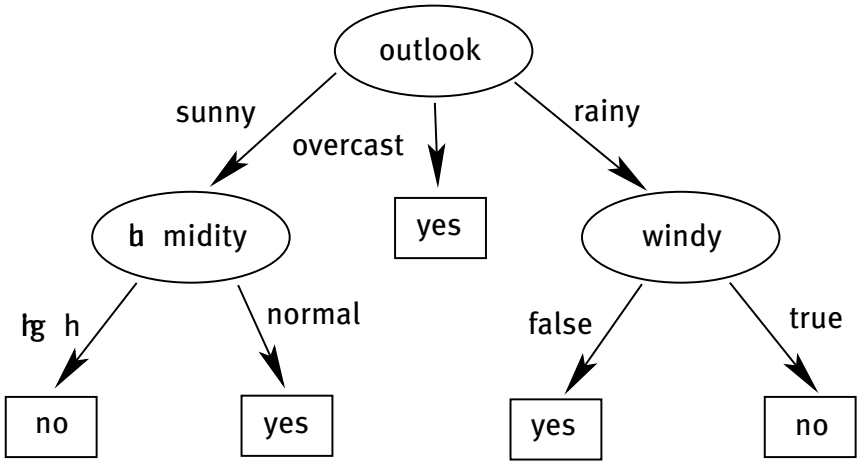

Abbildung 2.3: Entscheidungsbaum (Quelle: Cleve und Lämmel, 2020, S. 76)

Bei Entscheidungsbäumen werden mit Hilfe einer Reihe von Fragen Entscheidungen getroffen, indem mit diesen Fragen die Klassenbezeichnungen der Objekte in der Trainingsdatenmenge erlernt werden. Dabei wird bei der Wurzel des Baums angefangen und die Daten so aufgeteilt, dass sich der größtmögliche *Informationsgewinn* (IG, engl. Information Gain) ergibt. Dieser Vorgang wir bei jedem Kindknoten wiederholt, bis die Blattknoten erreicht werden. Dementsprechend gehören alle Exemplare der Blattknoten zu der selben Klasse. Durch dieses Vorgehen können sehr tief verschachtelte Baumstrukturen mit vielen Knoten entstehen, was zu einer Überanpassung führen kann. Um dies zu verhindern, wird beim sogenannten *Pruning* die maximale Tiefe des Baums begrenzt (Raschka & Mirjalili, 2019/2021).

Um die Knoten an den Merkmalen mit den höchsten Informationsgewinn aufteilen zu können, wird eine Zielfunktion definiert. Diese soll durch den Algorithmus des Entscheidungsbaums optimiert werden. Der Informationsgewinn soll bei jeder Aufteilung maximiert werden:

$$IG\left(D_p, f\right) = I\left(D_p\right) - \sum_{j=1}^{m} \frac{N_j}{N_p} I\left(D_j\right) \tag{2.2}$$

Dabei bezeichnet f das Merkmal, an dem die Aufteilung erfolgen soll, I ist das Maß der *Impurity* (Unreinheit), D_P und D_j bezeichnen jeweils die Datenmengen des Elternknotens und des j-ten Kindknotens, N_p bezeichnet die gesamte Anzahl der Exemplare des Elternknotens und N_j bezeichnet die Anzahl der Exemplare im j-ten Kindknoten. Aus Formel (2.2) geht hervor, dass der Informationsgewinn sich als Differenz aus dem Impurity-Wert des Elternknotens und der Summe der Impurity-Werte der Kindknoten ergibt. Dementsprechend führt ein geringerer Impurity-Wert der Kindknoten zu einem größeren Informationsgewinn. Viele Programmbibliotheken verwenden binäre Entscheidungsbäume, bei denen jeder Elternknoten in zwei Kindknoten D_{links} und D_{rechts} aufgeteilt wird:

$$IG\left(D_p, f\right) = I\left(D_p\right) - \frac{N_{links}}{N_p} I(D_{links}) - \frac{N_{rechts}}{N_p} I(D_{rechts}) \tag{2.3}$$

Um die Impurity zu bewerten, werden bei binären Entscheidungsbäumen drei Maße verwendet: die *Entropie* (I_H), der *Gini-Koeffizient* (I_G) und der *Klassifikationsfehler* (I_E). Die Entropie wird für alle nicht-leeren Mengen ($p(i \mid t) \neq 0$) definiert als:

$$I_H(t) = -\sum_{i=1}^{c} p(i \mid t) \log_2 p(i \mid t) \tag{2.4}$$

Hierbei steht $p(i \mid t)$ für den Anteil der Exemplare für den Knoten t die zur Klasse i gehören. Dementsprechend ist die Entropie 0, wenn alle Objekte eines Knotens zur selben Klasse gehören. Bei einer gleichmäßigen Verteilung der Objekte auf alle Klassen ist die Entropie maximal. Der Gini-Koeffizient gibt an in welchem Maß die Wahrscheinlichkeit einer Fehlklassifikation minimiert ist:

$$I_G(t) = \sum_{i=1}^{c} p(i \mid t)\left(1 - p(i \mid t)\right) = 1 - \sum_{i=1}^{c} p(i \mid t)^2 \tag{2.5}$$

Der Gini-Koeffizient ist maximal wenn die Klassen gleichmäßig verteilt sind, so wie es auch bei der Entropie der Fall ist. Gini-Koeffizient und Entropie liefern oft ähnliche Ergebnisse. Der Klassifikationsfehler ist ein weiteres Maß für die Impurity:

$$I_E(t) = 1 - max\{p(i \mid t)\} \tag{2.6}$$

Der Klassifikationsfehler wird beim Entscheidungsbaum als nützliches Kriterium zum Pruning verwendet. Er wird jedoch nicht für dessen Konstruktion verwendet, denn er ist gegenüber Änderungen der Klassenwahrscheinlichkeiten der Knoten weniger empfindlich (Raschka & Mirjalili, 2019/2021).

2.6.3 Random Forest

Bei einem *Random Forest* werden mehrere Entscheidungsbäume miteinander kombiniert. Es handelt sich um eine Ensemble-Methode. Random Forests zeichnen sich durch ihre gute Skalierbarkeit und durch einfache Handhabung aus. Ein Random Forest ist wie ein *Ensemble* aus Entscheidungsbäumen. Für ein solches Ensemble kann beispielsweise eine Gruppe an Entscheidungsbäumen auf unterschiedlichen Teilmengen der Trainingsdaten trainiert werden. Für das Treffen einer Vorhersage mit diesem Ensemble werden die Vorhersagen aller Bäume gesammelt. Dieses Ensemble sagt dann die Kategorie vorher, die die meisten Stimmen erhalten hat. Diese Methode kann zu einem stabileren Modell führen, das eine geringere Anfälligkeit für eine Überanpassung aufweist (Géron, 2019/2020; Raschka & Mirjalili, 2019/2021).

Der Random-Forest-Algorithmus lässt sich in vier Schritten zusammenfassen. Zuerst wird eine zufällige Stichprobe aus den Trainingsdaten der Größe n aus der Trainingsdatenmenge ausgewählt (mit Zurücklegen). Im zweiten Schritt wird mithilfe der Stichprobe ein Entscheidungsbaum erstellt. Für jeden Knoten werden zufällig d Merkmale ausgewählt (ohne Zurücklegen) und der Knoten wird an dem Merkmal aufgeteilt, welches in Bezug auf die Zielfunktion am besten geeignet ist. Im dritten Schritt werden der erste und der zweite Schritt k-mal wiederholt. Anschließend wird im vierten und letzten Schritt eine Mehrheitsentscheidung durchgeführt, in der die Vorhersagen der einzelnen Bäume zusammengefasst werden, um dadurch die Klassenbezeichnung zuzuweisen.

Der große Vorteil bei Random Forests ist, dass sich kaum mit der Auswahl der Hyperparameter auseinandergesetzt werden muss. Weil es sich um ein Ensemble-Modell handelt, welches nicht sehr anfällig ist in Bezug auf das „Rauschen“ von einzelnen Entscheidungsbäumen, muss normalerweise der Random Forest nicht zurechtgestutzt werden. Jedoch sollte die Anzahl k der Entscheidungsbäume bei der Auswahl eines Random Forest als Parameter beachtet werden. Auch Hyperparameter wie die Größe n der Stichprobe aus der Trainingsdatenmenge und die Anzahl d der zufällig ausgewählten Merkmale können optimiert werden, auch wenn dies weniger üblich ist. Ein niedrigerer Wert für n kann die Diversität der einzelnen Bäume erhöhen, da es dadurch unwahrscheinlicher wird, dass ein bestimmtes Objekt zur Stichprobe gehört. Außerdem wird die Zufälligkeit des Random Forests erhöht und eine Überanpassung hat geringere Auswirkungen. Jedoch führt dies ebenfalls zu einer schlechteren Leistung des Random Forests. Größere Werte für n hingegen können zu einer Überanpassung führen. Die Objekte in der Stichprobe sind einander ähnlicher und folglich sind auch die Entscheidungsbäume ähnlicher. Viele Implementierungen verwenden die Anzahl der Objekte im ursprünglichen Trainingsdatensatz als n der Stichprobe. Für die Anzahl d der Merkmale sollte ein Wert kleiner als die gesamte Zahl der Merkmale in der Trainingsdatenmenge gewählt werden. Ein Standardwert in vielen Implementierungen ist $d = \sqrt{m}$, mit m als Merkmalsanzahl in der Trainingsdatenmenge (Raschka & Mirjalili, 2019/2021).

2.6.4 Logistische Regression

Einige Regressionsalgorithmen lassen sich ebenfalls zur Klassifikation einsetzen. Dies ist auch bei der *logistischen Regression* der Fall. Die logistische Regression wird auch Logit-Regression genannt. Sie wird häufig verwendet um abzuschätzen wie wahrscheinlich es ist, dass ein Datenpunkt zu einer bestimmten Kategorie gehört. Wenn die Wahrscheinlichkeit bei über 50 % liegt sagt das Modell voraus, dass der Datenpunkt zu dieser Kategorie gehört. Es handelt sich dementsprechend um einen binären Klassifikator (Géron, 2019/2020).

Bei der logistischen Regression handelt es sich um ein Klassifikationsmodell, welches linear trennbare Klassen besitzt. Da es sich relativ einfach implementieren lässt, ist es einer der meistgebrauchten Klassifikationsalgorithmen. Die lineare Regression ist besonders geeignet, wenn von einem linearen Zusammenhang zwischen den Merkmalen und einer kontinuierlichen Zielvariablen ausgegangen werden kann. Bei maschinellen Lernalgorithmen wird versucht eine Beziehung von einem oder mehreren Merkmalen mit einer y-Variablen zu modellieren. Bei der linearen Regression wird eine Gerade durch eine Punktewolke gezogen. Dies ist jedoch für Klassifikationen mit nicht stetigen y-Variablen keine geeignete Herangehensweise. Für eine solche Klassifikation bietet es sich an, in diese Punktewolke ein Sigmoid, also eine s-förmige Kurve, zu legen. Diese gibt eine Zahl zwischen 0 und 1 aus (Hirschle, 2022; Raschka & Mirjalili, 2019/2021)

Um die Sigmoidfunktion zu verstehen, wird zuerst das Chancenverhältnis $\frac{p}{(1-p)}$ vorgestellt, welches eine wichtige Rolle bei der logistischen Regression spielt. Hierbei steht p für die Wahrscheinlichkeit des Positivereignisses, also das Ereignis, welches vorhergesagt werden soll. Angenommen dem Positivereignis wird die Klassenbezeichnung $y = 1$ zugeordnet und es wird die *logit*-Funktion definiert, als der Logarithmus des Chancenverhältnisses:

$$logit(p) = \log \frac{p}{(1-p)} \tag{2.7}$$

Es handelt sich hierbei um den natürlichen Logarithmus. Die logit-Funktion nimmt Werte zwischen 0 und 1 entgegen und bildet sie auf den gesamten Bereich der reellen Zahlen ab. Dadurch wird die Formulierung einer linearen Beziehung zwischen Merkmalswerten und dem Logarithmus des Chancenverhältnisses mit $p(y = 1 \mid x)$ als die bedingte Wahrscheinlichkeit für die Zugehörigkeit eines Objekts bei gegebenen Merkmal x zur Klasse 1 ermöglicht:

$$logit(p(y = 1 \mid x)) = w_0 x_0 + w_1 x_1 + ... + w_m x_m = \sum_{i=0}^{m} w_i \, x_i = w^T x \tag{2.8}$$

Um die Wahrscheinlichkeit dafür zu berechnen, dass ein bestimmtes Objekt zu einer Klasse gehört, wird der Kehrwert der *logit*-Funktion verwendet. Diese Funktion wird auch als *logistische Funktion* oder als *Sigmoidfunktion* bezeichnet:

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{2.9}$$

Bei z handelt es sich hierbei um die Netzeingabe, die eine Linearkombination aus Gewichten und Eingaben ist:

$$z = w^T x = w_0 + w_1 x_1 + ... + w_m x_m \tag{2.10}$$

Wenn z gegen unendlich geht, dann geht $\phi(z)$ gegen 1. Das liegt daran, dass e^{-z} bei großen z sehr klein wird. Umgekehrt geht $\phi(z)$ gegen 0, wenn z gegen minus unendlich geht, denn hierbei wird der Nenner sehr groß. Somit nimmt die Sigmoidfunktion reelle Zahlen entgegen und bildet sie auf das Intervall [0,1] ab. Die Sigmoidfunktion wird als Wahrscheinlichkeit $\phi(z) = P(y = 1 \mid x; w)$ ausgelegt, bei der ein ein Exemplar mit den Merkmalen x zur Klasse 1 gehört, wenn es mit es mit den Gewichten x ausgestattet ist (Raschka & Mirjalili, 2019/2021).

2.6.5 Convolutional Neural Network

Das *Convolutional Neural Network* (CNN) gehört „zu einer Modellfamilie, die sich an der Funktionsweise der Sehrinde (visueller Kortex) des menschlichen Gehirns bei der Erkennung von Objekten orientiert“ (Raschka & Mirjalili, 2019/2021). Es wird auch als *Konvolutionsnetz* bezeichnet. Der Name stammt von den Konvolutionen, die auch Faltungen genannt werden (Mueller & Massaron, 2019/2020). Bei der Leistung eines Lernalgorithmus ist das extrahieren von auffälligen (relevanten) Merkmalen besonders wichtig. Bei herkömmlichen Lernalgorithmen werden von Expertenwissen vorgegebene Eingabemerkmale oder Verfahren zur Berechnung der Merkmalsextraktion verwendet. Bestimmten Neuronalen Netzen wie CNNs sind im Stande, die wesentlichen Merkmale bei bestimmten Aufgaben lediglich anhand der Rohdaten automatisch zu erkennen. Neuronale Netze werden als Mechanismus zur Merkmalsextraktion betrachtet. Die nach der Eingabeschicht folgenden ersten Schichten extrahieren die Low-Level-Merkmale aus den Rohdaten. Anschließend verwenden die später folgenden Schichten diese Merkmale für die Vorhersage eines stetigen Zielwerts oder einer Klassenbezeichnung. Diese sind meist vollständig verbunden, wie bei dem mehrschichtigem Perzeptron MLP.

Von einigen mehrschichtigen neuronalen Netzen und besonders Deep Convolutional Networks wird eine sogenannte Merkmalshierarchie konstruiert. Dies geschieht, indem Low-Level-Merkmale schichtweise zu High-Level-Merkmalen kombiniert werden. Typischerweise bestehen CNNs aus mehreren Faltungsschichten, Pooling-Schichten und anschließend vollständig verknüpften Schichten. Letztere entsprechen im Wesentlichen einem MLP, bei dem alle Eingabeeinheiten i mit allen Ausgabeeinheiten j und den Gewichten w_{ij} verknüpft sind. Die Pooling-Schichten besitzen keine erlernbaren Parameter, es gibt zum Beispiel keine Gewichte. Die Faltungsschichten und die vollständig verknüpften Schichten besitzen diese jedoch. Sie werden bei ihnen während des Trainings optimiert.

In einem CNN ist die *diskrete Faltung* eine grundlegende Operation. Die diskrete Faltung von zwei eindimensionalen Vektoren x und w wird notiert als $y = x * w$, mit x als Eingabe (auch *Signal* genannt) und w als *Filter* oder *Kernel*. Die diskrete Faltung wird definiert als:

$$y = x * w \rightarrow y[i] = \sum_{k=-\infty}^{+\infty} x[i-k]\, w[k] \tag{2.11}$$

In der Definition stehen die eckigen Klammern für die Indizierung der Vektorelemente. Der Index i durchläuft die Elemente des Ausgabevektors y. Die Indizes der Summe laufen von $-\infty$ bis $+\infty$, auch wenn Anwendungen des Machine Learnings immer endliche Merkmalsvektoren besitzen. Wenn x 10 Merkmale mit den Indizes von 0 bis 9 besitzt, so sind die Indizes von $-\infty$ bis -1 und von 10 bis $+\infty$ nicht vorhanden. Deswegen wird für die korrekte Berechnung der Summe für x und w angenommen, dass die fehlenden Positionen Nullen enthalten. Dadurch ergibt sich ein unendlich großer Ausgabevektor y, der viele Nullen enthält. Da dies in der Praxis nicht umsetzbar ist, wird x mit einer begrenzten Anzahl an Nullen aufgefüllt. Dieses Verfahren wird als *Zero-Padding* oder einfach als *Padding* bezeichnet. Die Anzahl der aufgefüllten Nullen auf beiden Seiten wird als p bezeichnet. Angenommen die Eingabe x und der Filter w besitzen ursprünglich n bzw. m Elemente, mit $m \leq n$. Dann hat der mit Nullen aufgefüllte Vektor x^p eine Größe von $n+2p$ und es ergibt sich die folgende Gleichung für die diskrete Faltung:

$$y = x * w \rightarrow y[i] = \sum_{k=0}^{k=m-1} x^p\,[i+m-k]\, w[k] \tag{2.12}$$

Eine weitere Besonderheit ist die Indizierung von x mit $i+m-k$. Dabei werden x und w bei der Summierung in gegensätzlicher Richtung indiziert, wodurch einer der Vektoren nach dem Padding umgekehrt werden kann. Anschließend lässt sich das Skalarprodukt berechnen. Es gibt unterschiedliche Padding-Modi. Bei CNNs ist das Same-Padding am verbreitetsten. Das Same-Padding wird verwendet, wenn die Größe der Ausgabe mit der Größe der Eingabevektors x übereinstimmen soll. Für den Parameter p wird die entsprechende Größe des Filters berechnet.

Die Größe der Faltungsausgabe wird durch die Anzahl der Verschiebungen des Filters w entlang des Eingabevektors festgelegt. Mit einem Eingabevektor der Größe n und einem Filter der Größe m ergibt sich die Größe der Ausgabe von $x * w$ mit dem Padding p und der Schrittweite s folgendermaßen:

$$o = \left\lfloor \frac{n + 2p - m}{s} \right\rfloor + 1 \tag{2.13}$$

Die Abrundungsfunktion (Gaußklammer) liefert die größte ganze Zahl, die kleiner oder gleich der Eingabe ist.

Die zweidimensionale Faltung beruht auf den bereits vorgestellten Konzepten. Diese lassen sich auf zwei Dimensionen übertragen. Bei zweidimensionalen Eingaben mit einer Matrix $X_{n_1 \times n_2}$ und einer Filtermatrix $W_{m_1 \times m_2}$ und mit $m_1 \leq n_1$ und $m_2 \leq n_2$ ist das Ergebnis der zweidimensionalen Faltung von X und W die Matrix $Y = X * W$:

$$Y = X * W \rightarrow Y[i,j] = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} X[i-k_1, j-k_2]\, W[k_1, k_2] \tag{2.14}$$

Es handelt sich hierbei um dieselbe Formel wie bei der eindimensionalen Faltung, jedoch mit einer zusätzlichen Dimension. Zusätzlich lassen sich Verfahren wie das Zero-Padding, Filtermatrix umkehren und die Schrittweite anwenden, wenn diese für unabhängig voneinander auf beide Dimensionen erweitert werden.

In CNNs werden zwei Pooling-Operationen angewendet: das *Max-Pooling* und das *Mean-Pooling*, welches auch als *Average-Pooling* bezeichnet wird. Bei der Pooling-Schicht $P_{n_1 \times n_2}$ gibt der tiefgestellte Index die Nachbarschaftsgröße an, mit der die Max- oder Mean-Operation durchgeführt wird. Dies wird als *Pooling-Größe* bezeichnet. Beim Max-Pooling wird der maximale Wert der benachbarten Pixel genommen und beim Mean-Pooling wird der Mittelwert gebildet. Das Pooling führt zu einer lokalen Invarianz, sodass kleine Veränderungen in der lokalen Nachbarschaft das Ergebnis des Max-Poolings verändern. Zudem verringert es die Größe von Merkmalen, was zu einer effizienteren Berechnung führt und ebenfalls das Ausmaß der Überanpassung reduziert (Raschka & Mirjalili, 2019/2021).

3 Datensatz

In diesem Kapitel wird der verwendete Datensatz vorgestellt. Dieser Datensatz wird für die folgende Untersuchung auf einen Bias in der Textklassifikation verwendet. Dazu wird zunächst die Auswahl des Datensatzes erläutert. Anschließend wird der Datensatz mit den darin enthaltenen Daten vorgestellt. Schließlich wird der Datensatz mit den darin enthaltenen Daten visualisiert.

3.1 Auswahl eines Datensatzes

Passende Datensätze für die Untersuchung auf unbeabsichtigte Bias in der Textklassifikation sind nicht einfach zu finden. Obwohl täglich eine große Menge an Kommentaren auf verschiedenen sozialen Medienplattformen wie Twitter, Facebook und Reddit veröffentlicht wird, eignen sich diese Daten ohne Label nur für unüberwachtes Lernen und nicht für überwachtes Lernen, das in dieser Arbeit verwendet wird. Für diese Label müssen zuerst menschliche Kommentator*innen für jeden einzelnen Kommentar in einem aufwändigen Verfahren prüfen, ob er in eine der vordefinierten Klassen passt. Auf diese Weise können Kommentare beispielsweise hinsichtlich ihrer Toxizität bewertet werden. Herausforderungen wie mehrdeutige Kommentare, die Abhängigkeit vom Kontext des Kommentars, unterschiedliche Richtlinien für die Kennzeichnung der Label und eine generell niedrige Qualität der Kommentare erschweren diesen Prozess für die Kommentator*innen.

Das Sammeln einer großen Anzahl von kommentierten, toxischen Kommentaren ist zudem aus anderen Gründen kompliziert. Zum einen werden toxische Kommentare von Moderator*innen bearbeitet oder gelöscht. Dies kann kurz vor oder nach der Veröffentlichung erfolgen, so dass diese Kommentare der Öffentlichkeit gar nicht oder nur für kurze Zeit zugänglich sind. Zum anderen stellt die Wiederverwendbarkeit ein Problem beim Auffinden von kommentierten Datensätzen dar. Obwohl die Kommentare öffentlich zugänglich sind, ist es den Forschern in der Regel nicht gestattet, die von ihnen kommentierten Datensätze weiterzugeben (Risch & Krestel, 2020). Um auf unbeabsichtigte Bias zu prüfen, müssen die Kommentare zudem auf die erwähnten Personengruppen hin kommentiert werden, was die Auswahl weiter einschränkt.

Viele öffentlich zugängliche Datensätze für die Textklassifikation wie der Datensatz der Internet Movie Database (IMDB) mit 50 Tausend Filmbewertungen (Maas et al., 2011) oder der Sentiment140 Datensatz mit 1,6 Millionen Tweets (Go et al., 2009) wurden lediglich hinsichtlich der Stimmung des Kommentars bewertet. Mit diesen Datensätzen lässt sich eine Textklassifikation durchführen, für die Untersuchung auf einen Bias ist jedoch eine Bewertung der im Kommentar erwähnten Personengruppen hilfreich. Ist diese Bewertung nicht vorhanden, kann die Untersuchung mit einem zusammengestellten Korpus wie dem Equity Evaluation Corpus (EEC) von Kiritchenko und Mohammad (2018), der in Abschnitt 2.5.2 vorgestellt wurde, durchgeführt werden. In diesem Fall ist die Bewertung des Datensatzes auf die enthaltenen Personengruppen nicht notwendig. Jedoch sind die Satzvorlagen für den EEC sehr einfach und der EEC untersucht nur auf unbeabsichtigten Bias hinsichtlich des Geschlechts und der *Race*, weshalb ein kommentierter Datensatz von Vorteil wäre. Dies hat die Auswahl eines geeigneten Datensatzes für diese Arbeit erheblich erschwert.

3.2 Beschreibung des Datensatzes

Der für diese Untersuchung verwendete Datensatz stammt von Kaggle (Jigsaw / Conversation AI, 2019). Er wurde vom *Conversation AI* Team erstellt, einer Forschungsinitiative von Jigsaw und Google. Der Datensatz beinhaltet 1.804.874 englische Kommentare im Trainingsdatensatz und 194.640 englische Kommentare im Testdatensatz (zusammengesetzt aus einem öffentlichen und einem privaten Testdatensatz). Die Kommentare stammen von der Plattform *Civil Comments*, die Ende 2017 ihren Betrieb einstellte und im Anschluss die öffentlichen Kommentare in einem Archiv zur Verfügung stellte. Der Datensatz wurde zuerst für einen Wettbewerb veröffentlicht, in dem ein Modell erstellt werden sollte, welches die Erkennung der Toxizität ermöglicht und ebenfalls den unbeabsichtigten Bias in Bezug auf die erwähnten Identitäten minimiert.

Alle Kommentare im Datensatz wurden in Bezug auf ihre Toxizität (Toxicity) von mehreren menschlichen Kommentator*innen bewertet. Ein Teil der Kommentare wurde auch hinsichtlich der erwähnten Personengruppen bewertet. Diese Personengruppen werden als Identität bezeichnet. Die daraus resultierenden Werte für Toxizität und Identität liegen zwischen 0,0 und 1,0. Hierbei steht 0,0 für keine vorhandene Toxizität oder Identität, während 1,0 für eine erkennbare Toxizität oder Identität steht. Die Bewertung der Kommentare fand über die Crowd-Rating-Plattform Figure Eight statt. Die Kommentator*innen kamen aus der ganzen Welt und beherrschten alle die englische Sprache. Die Qualität der Bewertungen wurde mit Hilfe von Testfragen erzwungen. Es handelt sich um ein System, bei dem etwa 10 % der Kommentare, die die Kommentator*innen sehen, bereits korrekt gekennzeichnet sind. Kommentator*innen, die zu viele dieser Kennzeichnungen falsch bewerten, werden von der Bewertung ausgeschlossen.

Da Toxizität und Identität subjektiv sein können, werden bis zu 10 Personen pro Kommentar befragt. Dadurch wird sichergestellt, dass ein breites Meinungsspektrum erfasst wird. Einige Kommentare wurden jedoch von viel mehr als 10 Kommentator*innen (bis zu Tausenden) gesehen. Dies ist auf die Stichprobenbildung und die Strategien zurückzuführen, mit denen die Genauigkeit der Kommentator*innen sichergestellt werden soll. Die Kommentator*innen wurden gefragt, wie sie die Toxizität des Kommentars bewerten würden:

- sehr toxisch (ein sehr hasserfüllter, aggressiver oder respektloser Kommentar, der sie sehr wahrscheinlich dazu bringt, die Diskussion zu verlassen oder die Perspektive aufzugeben),
- toxisch (ein unhöflicher, respektloser oder unangemessener Kommentar, der sie mit einiger Wahrscheinlichkeit dazu bringt, die Diskussion zu verlassen oder die Perspektive aufzugeben),
- schwer zu sagen oder
- nicht toxisch.

Diese Bewertungen wurden dann zusammengefasst, wobei der Zielwert (Toxizität) den Anteil der Anmerkungen darstellt, die in die beiden erstgenannten Kategorien fallen. Für die Auswertung werden im Wettbewerb die Werte größer als oder gleich 0,5 in die positive (toxische) Klasse eingestuft. Zudem sind in den Daten zusätzliche Untergruppen für die Toxizität enthalten, die jedoch nicht im Wettbewerb vorhergesagt werden sollten. Die Bezeichnungen für diese sind: Severe Toxicity, Obscene, Threat, Insult, Identity Attack und Sexual Explicit. Diese wurden ebenfalls bewertet und haben einen Wert zwischen 0,0 und 1,0 zugeordnet bekommen.

Tabelle 3.1: Identitäten im Datensatz

Kategorie	Identität
Geschlecht	**Male**, **Female**, Transgender, other Gender
Sexuelle Orientierung	Heterosexual, **Homosexual Gay or Lesbian**, Bisexual, other Sexual Orientation
Religion	**Christian**, **Jewish**, **Muslim**, Hindu, Buddhist, Atheist, other Religion
Race oder Ethnie	**Black**, **White**, Asian, Latino, other Race or Ethnicity
Behinderung	Physical Disability, Intellectual or Learning Disability, **Psychiatric or Mental Illness**, other Disability

Für ungefähr 22 % der Kommentare im Trainingsdatensatz sowie im Testdatensatz wurde zusätzlich bestimmt, ob eine Identität in diesem Kommentar erwähnt wurde. Bei der Identität handelt es sich wie in Tabelle 3.1 dargestellt um Bezeichnungen, die sich in die Kategorien Geschlecht, sexuelle Orientierung, Religion, *Race* oder Ethnie sowie in die Kategorie Behinderung einordnen lassen. Bestimmte Bezeichnungen für die Identität in Tabelle 3.1 sind fett gedruckt. Dies sind die Identitäten, die in mehr als 500 Beispielen im Testdatensatz (öffentlich und privat) erwähnt werden. Nur Identitäten mit mehr als 500 Beispielen in der Testmenge werden in die Berechnung der Bewertung mit einbezogen. Nur diese Identitäten werden auf einen unbeabsichtigten Bias untersucht, da in den Datensätzen ausreichend Daten für sie vorhanden sind.

Um die Identitäten zu bewerten, wurde eine Kombination aus Modellvorhersagen und Wortabgleich verwendet, um etwa 250.000 Kommentare zu finden, die wahrscheinlich eine Identität enthalten. Diese Daten wurden mit mit etwa 250.000 zufällig ausgewählten Kommentaren kombiniert, um eine Menge von etwa 500.000 zu kennzeichnenden Kommentaren zu erhalten. Bei diesen Daten konnte davon ausgegangen werden, dass etwa 50 % der Kommentare Hinweise auf Identität enthielten. Um die Identitäten zu bewerten, wurden die Kommentator*innen gefragt, alle im Kommentar erwähnten Identitäten anzugeben. Beispielsweise wurde hierfür gefragt, welche Geschlechter in dem Kommentar erwähnt wurden. Mögliche Antworten waren Mann, Frau, Transgender, anderes Geschlecht oder kein Geschlecht. Diese Werte wurden erneut zu Anteilswerten zusammengerechnet, die den Anteil der Kommentator*innen darstellen, die angaben, dass die Identität im Kommentar erwähnt wurde.

Ein Beispiel für einen Kommentar im Datensatz ist der Satz „i'm a white woman in my late 60's and believe me, they are not too crazy about me either!!". Dieser Kommentar hat eine Bewertung von 0,0 für die Toxizität (Toxicity) erhalten und jeweils eine 1,0 für Frau (Female) und Weiß (White). Alle anderen Identitäten wurden als 0,0 eingestuft. Ein anderes Beispiel ist der Satz „Continue to stand strong LGBT community. Yes, indeed, you'll overcome and you have.", welcher eine Bewertung von 0,0 für die Toxizität (Toxicity) erhalten hat. Für die Identität hat der Kommentar eine 0,8 für Homosexuell Schwul oder Lesbisch (Homosexual Gay Or Lesbian), eine 0.6 für Bisexuell (Bisexual) und eine 0,3 für Transgender erhalten. Alle anderen Identitäten wurden erneut als 0,0 eingestuft.

Zudem enthält der Datensatz zusätzliche Informationen, die in derer Tabelle 3.2 dargestellt werden. Zum einen handelt es sich um Informationen zum Kommentar wie die Anzahl der Reaktionen, beispielsweise die Anzahl der Likes. Aber auch Informationen wie das Erstellungsdatum des Kommentars sind im Datensatz enthalten. Außerdem wurde die Anzahl der Kommentator*innen erfasst, die die Toxizität und Identität bewertet haben.

Tabelle 3.2: Spalten im Datensatz

Kategorie	Spaltennamen
Text	Comment Text
Toxizität	Toxicity/Target, Severe Toxicity, Obscene, Sexual Explicit, Identity Attack, Insult, Threat
Identität	**Male**, **Female**, Transgender, other Gender, Heterosexual, **Homosexual Gay or Lesbian**, Bisexual, other Sexual Orientation, **Christian**, **Jewish**, **Muslim**, Hindu, Buddhist, Atheist, other Religion, **Black**, **White**, Asian, Latino, other Race or Ethnicity, Physical Disability, Intellectual or Learning Disability, **Psychiatric or Mental Illness**, other Disability
Reaktionen	Funny, Wow, Sad, Likes, Disagree
Kommentarinformationen	ID, Created Date, Publication ID, Parent ID, Article ID
Genehmigt/Abgelehnt	Rating
Anzahl der Kommentator*innen	Toxicity Annotator Count, Identity Annotator Count

3.3 Visualisierung des Datensatzes

Für die Visualisierung des Datensatzes wurden die Bibliotheken *Matplotlib* und *Seaborn* verwendet. Seaborn bietet gemeinsam mit Matplotlib gute Werkzeuge zur Visualisierung von Daten (Frochte, 2019). Zuerst wurde der Datensatz auf fehlende Werte untersucht. Die Untersuchung hat ergeben, dass nur bei den Identitäten und bei der „Parent ID“ Werte fehlen. In der vorliegenden Arbeit ist die „Parent ID“ nicht relevant. Für alle Identitäten fehlen jeweils 1.399.744 Werte in den Trainingsdaten und 151.770 Werte in den Trainingsdaten. Damit ist für jeweils ca. 78 % der Kommentare kein Wert für die Identität vorhanden. Dies entspricht der Erwartung, weil nur für einen Teil der Daten die Identitäten bewertet wurden. Die Toxizitätswerte und die vorhandenen Identitätswerte liegen alle im erwarteten Bereich zwischen 0,0 und 1,0. Über die Spalte „Created Date“ lässt sich zudem herausfinden, dass die erhobenen Daten aus dem Zeitraum von Oktober 2015 bis November 2017 stammen.

Nachdem die Vollständigkeit der Daten festgestellt wurde, kann es hilfreich sein, Teile des Datensatzes zu visualisieren. Insbesondere die in Tabelle 3.1 fett gedruckten Identitäten, die auf unbeabsichtigten Bias untersucht werden, sind für eine solche Visualisierung interessant. Zunächst wird die Verteilung der Identitäten mit einem Wert größer oder gleich 0,5 betrachtet. Jede Identität, die einen Wert größer oder gleich 0,5 hat, gilt als positiv bzw. als im Kommentar

vorhanden. Diese Verteilung ist in Abbildung 3.1 für die Trainingsdaten dargestellt. Die Identität „Female“ ist mit 53.429 Kommentaren am häufigsten vertreten. Die Identitäten „Male“ und „Christian“ sind mit einer Anzahl von 44.484 und 40.423 ebenfalls häufig vertreten. Am seltensten wird mit einer Anzahl von 4.889 die Identität „Psychiatric or Mental Illness“ in den Kommentaren erwähnt. Auch die Identitäten „Jewish“ und „Homosexual Gay or Lesbian“ werden mit einer Anzahl von 7.651 und 10.997 weniger häufig genannt. Die Identitäten „Muslim“, „Black“ und „White“ kommen mit einer mittleren Häufigkeit von 21.006, 14.901 und 25.082 vor.

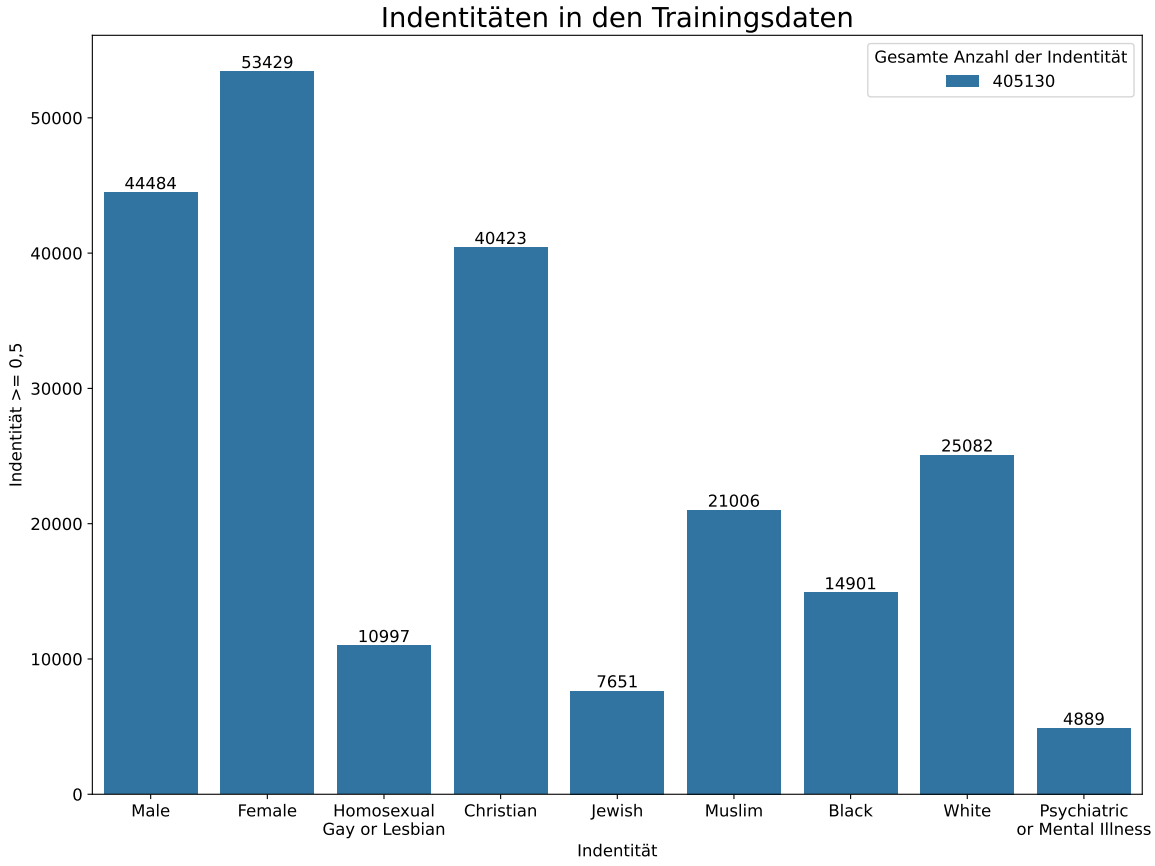

Abbildung 3.1: Anzahl der Identitäten in den Trainingsdaten

Zudem ist für eine Untersuchung interessant, ob diese Verteilung ebenfalls in den Testdaten wiederzufinden ist. In Abbildung 3.2 werden die Anzahl der Identitäten mit einem Wert größer oder gleich 0,5 in den Trainingsdaten mit der Anzahl der Identitäten in den Testdaten verglichen. Wie in der Abbildung zu erkennen ist, ist die Verteilung der Identitäten in den Trainingsdaten und in den Testdaten sehr ähnlich. Der Trainingsdatensatz und der Testdatensatz repräsentieren sich also gegenseitig korrekt. Zumindest ist die Repräsentation bei der Verteilung der vorhandenen Identitäten vergleichbar.

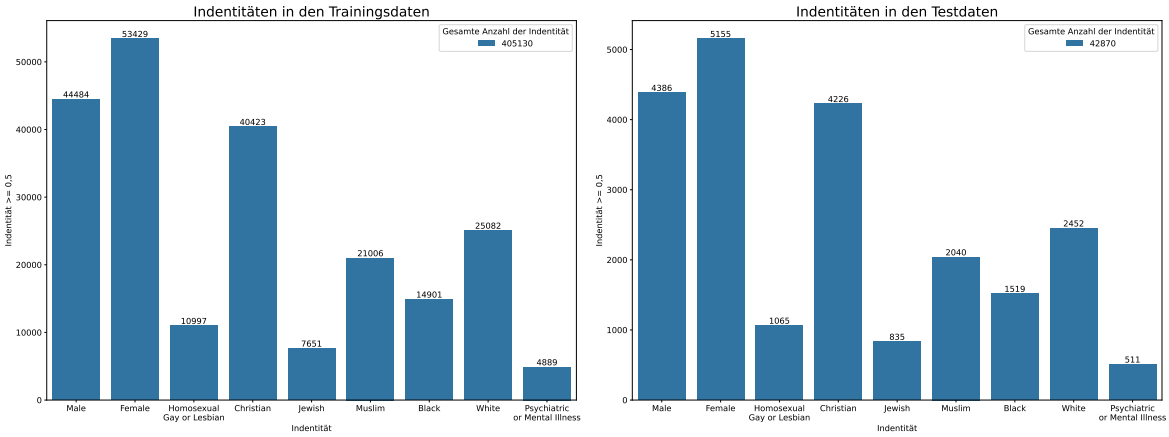

Abbildung 3.2: Anzahl der Identitäten in den Trainings- und Testdaten

Weiterhin ist eine Visualisierung der toxischen Kommentare in Bezug auf diese Identitäten sinnvoll. In der Abbildung 3.3 wird dargestellt, wie häufig die einzelnen Identitäten mit toxischen Kommentaren in Verbindung gebracht werden. Dafür wird jede Identität mit der Toxizität multipliziert und anschließend durch die Anzahl des Auftretens der Identität geteilt. Daraus ergibt sich für jede Identität eine gewichtete Toxizität. Die in dieser Arbeit nicht verwendeten Identitäten wurden zum Vergleich ebenfalls dargestellt. Die Identität mit der höchsten Toxizität ist „White“, dicht gefolgt von „Black“. „Christian“ hat hingegen die geringste Toxizität unter den verwendeten Identitäten.

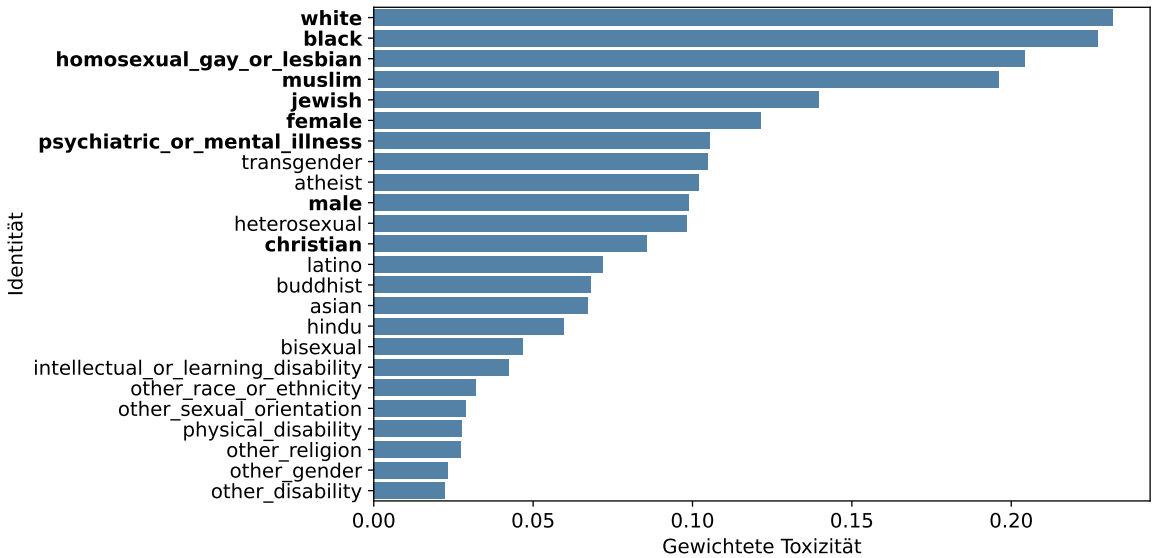

Abbildung 3.3: Toxizität in Bezug auf die Identitäten [eigene Grafik nach (NZ, 2019)]

4 Methoden zur Modellbewertung

In den vorangegangenen Kapiteln wurden die Klassifikationsverfahren und der Datensatz vorgestellt, mit denen die Modelle in dieser Arbeit erstellt werden. Um die Leistungen dieser Modelle anschließend beurteilen und miteinander vergleichen zu können, werden Methoden zur Modellbewertung benötigt. Die Bewertung der Vorhersagen von Modellen ist ein wesentlicher Bestandteil des maschinellen Lernens (Borkan, Dixon et al., 2019). Durch die Auswahl von geeigneten Metriken lassen sich die Vorhersageleistungen der einzelnen Modelle auswerten, um sie anschließend miteinander vergleichen zu können. In diesem Kapitel werden von den vielen bekannten Methoden zur Bewertung von Modellen einige ausgewählte Methoden vorgestellt. Die Konfusionsmatrix bietet einen guten Überblick über die Qualität eines Modells, indem die richtig positiven, falsch negativen, falsch positiven und richtig negativen Vorhersagen dargestellt werden. Für den Vergleich anhand eines kompakten Wertes wird die ROC-Kurve mit dem ROC AUC eingeführt. Im Rahmen des verwendeten Datensatzes werden zudem der Subgroup AUC, der BPSN AUC und der BNSP AUC zum Vergleichen der Modelle vorgestellt, die zu einem Gesamtmaß kombiniert werden und aus denen zuletzt eine endgültigen Metrik zur Modellbewertung erstellt wird.

4.1 Konfusionsmatrix

Die *Konfusionsmatrix* ist eine Möglichkeit zur Auswertung der Vorhersageleitung eines Klassifikators. Mit dieser quadratischen Matrix werden die Anzahlen der richtig positiven, falsch negativen, falsch positiven und richtig negativen Vorhersagen eines Klassifikators dargestellt. Für die Berechnung der Konfusionsmatrix wird ein Satz Vorhersagen mit den korrekten Zielwerten verglichen. Wie in Abbildung 4.1 zu sehen, steht in einer Konfusionsmatrix jede Zeile für eine tatsächliche Klasse und jede Spalte für eine vorhergesagte Klasse. Die erste Zeile der Matrix enthält die tatsächlich positiven Datenpunkte: in der ersten Spalte die korrekt als positiv vorhergesagten (richtig Positive, TP) und in der zweiten Spalte die fälschlicherweise als negativ klassifizierten (falsch Negative, FN) Datenpunkte. In der zweiten Zeile der Matrix befinden sich die tatsächlich negativen Datenpunkte: in der ersten Spalte die fälschlicherweise als positiv vorhergesagten (falsch Positive, FP) und in der zweiten Spalte die korrekt als negativ klassifizierten (richtig Negative, TN) Datenpunkte.

		Vorhergesagte Klasse	
		P	N
Tatsächliche Klasse	P	Richtig Positive (TP)	Falsch Negative (FN)
	N	Falsch Positive (FP)	Richtig Negative (TN)

Abbildung 4.1: Die Konfusionsmatrix [eigene Grafik nach (Raschka & Mirjalili, 2019/2021, S. 234)]

Die Konfusionsmatrix bietet viele Informationen, jedoch ist mitunter ein kompakteres Maß für die Qualität von Vorteil. Die *Relevanz* (engl. Precision) des Klassifikators ist das Maß für die Genauigkeit der positiven Vorhersagen. Sie wird mit der Anzahl der richtig Positiven TP geteilt durch die Summe aus der Anzahl der richtig Positiven und der Anzahl der falsch Positiven $TP + FP$ berechnet:

$$\text{Relevanz (Precision)} = \frac{TP}{TP + FP} \tag{4.1}$$

Für eine perfekte Relevanz kann mit einer einzigen Vorhersage gesorgt werden, bei der sichergestellt wird, dass sie richtig ist. Dadurch ergibt sich eine Relevanz von $1/1 = 100\%$. Da dies nicht sonderlich vorteilhaft ist, wird als ein weiteres Maß die *Sensitivität* (engl. Recall) verwendet. Diese wird auch als *Trefferquote* oder *Richtig-positiv-Rate* (engl. True Positive Rate, TPR) bezeichnet. Die Sensitivität wird mit der Anzahl der richtig Positiven TP geteilt durch die Summe aus der Anzahl der richtig Positiven und der Anzahl der falsch Negativen $TP + FN$ berechnet:

$$\text{Sensitivität (Recall)} = TPR = \frac{TP}{TP + FN} \tag{4.2}$$

Mit der Sensitivität wird dementsprechend der Anteil der positiven Datenpunkte berechnet, die der Klassifikator erkannt hat. Ein weiteres Leistungskriterium für einen Klassifikator ist neben der Richtig-positiv-Rate auch die *Falsch-positiv-Rate* (engl. False Positive Rate, FPR). Bei der FPR handelt es sich um den Anteil der negativen Datenpunkte, die fälschlicherweise als positiv klassifiziert werden. Diese wird mit der Anzahl der falsch Positiven FP geteilt durch die Summe aus der Anzahl der falsch Positiven und der Anzahl der richtig Negativen $FP + TN$ berechnet:

$$FPR = \frac{FP}{FP + TN} \tag{4.3}$$

Die TPR und die FPR sind Leistungskriterien, die insbesondere bei unausgeglichenen Klassifikationsaufgaben von Nutzen sind. Ein Beispiel hierfür ist die Diagnose von Tumoren, bei der die Erkennung von bösartigen Tumoren wichtiger ist als die Erkennung von gutartigen, um die betroffenen Patienten angemessen behandeln zu können. Außerdem sollte die Anzahl der falschen Positiven gering gehalten werden, damit gutartige Tumore nicht fälschlicherweise als bösartig eingestuft und die Patienten dadurch unnötig alarmiert werden (Géron, 2019/2020; Raschka & Mirjalili, 2019/2021). Dies lässt sich ebenfalls auf die Klassifikation von Kommentaren hinsichtlich ihrer Toxizität übertragen. Hierbei ist die Erkennung von toxischen Kommentaren besonders wichtig.

4.2 ROC-Kurve

Die *ROC-Kurve* (Receiver Operating Characteristic) ist besonders bei der binären Klassifikation als Hilfsmittel verbreitet. Die Kurve zeigt die TPR (4.2), also die Sensitivität, gegen die FPR (4.3). Die FPR kann auch aus der Differenz von eins und der *Richtig-negativ-Rate* (engl. True Negative Rate, TNR) berechnet werden. Die TNR gibt den Anteil der korrekt als negativ klassifizierten Datenpunkte an und wird auch als *Spezifität* bezeichnet. Bei der ROC-Kurve wird dementsprechend die Sensitivität gegen 1 − Spezifität gezeigt. Die Rate in der ROC-Kurve wird durch Verschiebung der Entscheidungsgrenze des Klassifikators berechnet.

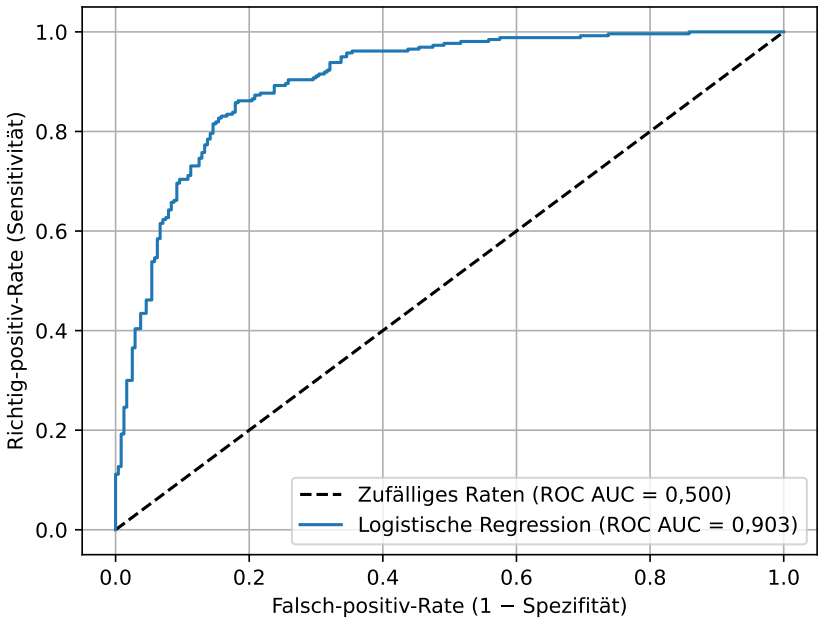

Abbildung 4.2: Die ROC-Kurve [eigene Grafik nach (Brownlee, 2021; Géron, 2019/2020, S. 101)]

Die blaue Linie in Abbildung 4.2 zeigt die ROC-Kurve für ein Modell, das mit logistischer Regression und mit einem zufällig generierten Datensatz nach Brownlee (2021) trainiert wurde. Das Modell dient lediglich als Beispiel für die Abbildung. Wie in der Abbildung 4.2 zu erkennen, erhöhen sich mit höherer Sensitivität (TPR) ebenfalls die falschen Positiven (FPR). Mit der gestrichelten Diagonalen wird zufälliges Raten dargestellt. Befindet sich ein Klassifikationsmodell unter dieser Linie, so ist es schlechter als zufälliges Raten. Je besser der Klassifikator ist, desto mehr entfernt er sich von dieser Linie und bewegt sich zunehmend in die Richtung der linken oberen Ecke. Ein perfekter Klassifikator würde sich in der linken oberen Ecke befinden und hätte eine TPR von 1 und eine FPR von 0 (Géron, 2019/2020; Raschka & Mirjalili, 2019/2021).

Die Fläche unter der ROC-Kurve wird als *Area under the Curve* (AUC) bezeichnet und wird zur Bestimmung der Leistung eines Klassifikationsmodells verwendet. Mit der AUC-Metrik wird die Wahrscheinlichkeit gemessen, dass ein zufällig ausgewähltes negatives Beispiel eine niedrigere Punktzahl erhält als ein zufällig ausgewähltes positives Beispiel. Bei einem ROC AUC-Wert von 1 handelt es sich um einen perfekten Klassifikator, bei dem alle negativen und positiven Paare richtig geordnet sind. Bei einem Wert von 0,5 dagegen handelt es sich um einen völlig zufälligen Klassifikator. Ein solcher zufälliger Klassifikator ist auch in der Abbildung 4.2 als gestrichelte Linie dargestellt. Ein großer Vorteil der ROC AUC-Metrik ist, dass sie unabhängig von einem Schwellenwert ist. Ein Wert bei AUC von 1,0 bedeutet, dass es möglich ist einen Schwellenwert zu wählen, der perfekt zwischen negativen und positiven Beispielen unterscheiden kann. Das bedeutet, dass die Klassen über den Modell-Score perfekt trennbar sind (Borkan, Dixon et al., 2019; Géron, 2019/2020).

4.3 AUC-basierte Metriken

Beim Wettbewerb von Jigsaw / Conversation AI (2019), aus welchem auch der in Kapitel 3 vorgestellte Datensatz stammt, wurde eine neu entwickelte Metrik zur Bewertung der Modelle verwendet. Diese Metrik setzt sich aus mehreren Submetriken zusammen, die auf der ROC AUC-Metrik basieren. Es werden dabei drei Teilmengen für jede Identität untersucht, wobei jede Teilmenge einen anderen Aspekt vom unbeabsichtigten Bias untersuchen soll. Diese Submetriken werden als *Bias AUCs* bezeichnet. In einem Konferenzpapier von Borkan, Dixon et al. in (2019) werden diese verwendeten Metriken detailliert vorgestellt. In den folgenden Abschnitten werden die drei Bias AUCs und die Berechnung der endgültigen Metrik mithilfe dieser drei Submetriken vorgestellt.

Wie bereits im vorherigen Abschnitt erläutert, ist ein wesentlicher Vorteil der AUC-Metrik, dass sie unabhängig von einem Schwellenwert ist. Dies ist bei der Untersuchung auf unbeabsichtigten Bias von Vorteil. In der Praxis liefern Klassifizierungsmodelle oft Scores anstelle

von binären Klassifizierungsentscheidungen. Bei diesen Modellen können Metriken, die von einem Schwellenwert abhängig sind, die Untersuchung auf einen unbeabsichtigten Bias beeinträchtigen und dadurch irreführend sein. Die im Folgenden vorgestellten Submetriken basieren auf der AUC-Metrik und sind somit ebenfalls unabhängig von einem Schwellenwert, wodurch sie sich gut für die Untersuchung auf einen unbeabsichtigten Bias eignen.

Ein weiterer Vorteil der im Folgenden vorgestellten Metriken ist, dass die Daten bei diesen Metriken in Untergruppen (engl. subgroups) aufgeteilt werden. Anstatt die Metriken jedoch, wie bei vielen anderen Metriken für die Untersuchung auf unbeabsichtigten Bias, ausschließlich für die Daten der jeweiligen Untergruppe zu berechnen, vergleichen die im Folgenden vorgestellten Metriken die Untergruppe mit den übrigen Daten. Diese übrigen Daten werden als „Hintergrund"-Daten (engl. „background" data) bezeichnet.

Der Datensatz kann dementsprechend in Hintergrund- und Identitätsuntergruppen (eng. background and identity subgroups) sowie in negative und positive Klassifizierungen unterteilt werden, wodurch vier verschiedene Untergruppen entstehen: negative Beispiele im Hintergrund, positive Beispiele im Hintergrund, negative Beispiele in der Untergruppe und positive Beispiele in der Untergruppe. Dadurch können drei Bias AUCs zur Messung der negativen/positiven falschen Einordnung zwischen diesen vier Untergruppen definiert werden: der Subgroup AUC, der BPSN (Background Positive, Subgroup Negative) AUC und der BNSP (Background Negative, Subgroup Positive) AUC.

4.3.1 Subgroup AUC

Beim *Subgroup AUC* wird der Datensatz auf die Beispiele beschränkt, in denen die spezifische Identitätsuntergruppe erwähnt wird. Für diese Identitätsuntergruppe wird der ROC AUC-Wert berechnet. Durch den Subgroup AUC kann ein Verständnis über das Modell und die Trennbarkeit innerhalb der Untergruppe erlangt werden. Ein niedriger Wert bei dieser Metrik bedeutet, dass das Modell schlecht zwischen toxischen und nicht-toxischen Kommentaren, die die spezifische Identität erwähnen, unterscheiden kann.

4.3.2 BPSN (Background Positive, Subgroup Negative) AUC

Beim *BPSN* (*Background Positive*, *Subgroup Negative*) *AUC* wird der AUC-Wert für die positiven Beispiele aus dem Hintergrund und die negativen Beispiele aus der Untergruppe berechnet. Das heißt, dass der Testdatensatz auf die nicht-toxischen Beispiele, die die Identität erwähnen, und die toxischen Beispiele, die die Identität nicht erwähnen, beschränkt wird. Dieser Wert würde sich verringern, wenn die Werte für negative Beispiele in der Untergruppe höher sind als die Werte für andere positive Beispiele. Ein niedriger Wert bei dieser Metrik

bedeutet, dass das Modell nicht-toxische Beispiele, die die Identität erwähnen, mit toxischen Beispielen verwechselt, die die Identität nicht erwähnen. Dies bedeutet wiederum, dass das Modell wahrscheinlich höhere Toxizitätswerte für nicht-toxische Beispiele, die die Identität erwähnen, vorhersagt, als es sollte. Diese nicht-toxischen Beispiele würden wahrscheinlich bei vielen Schwellenwerten in der Untergruppe als falsch Positive erscheinen.

4.3.3 BNSP (Background Negative, Subgroup Positive) AUC

Beim *BNSP* (*Background Negative*, *Subgroup Positive*) *AUC* wird der AUC-Wert für die negativen Beispiele aus dem Hintergrund und die positiven Beispiele aus der Untergruppe berechnet. Das heißt, dass der Testdatensatz auf die toxischen Beispiele, die die Identität erwähnen, und die nicht-toxischen Beispiele, die die Identität nicht erwähnen, beschränkt wird. Dieser Wert würde sich verringern, wenn die Werte für positive Beispiele in der Untergruppe niedriger sind als die Werte für andere negative Beispiele. Hierbei bedeutet ein niedriger Wert, dass das Modell toxische Beispiele, die die Identität erwähnen, mit nicht-toxischen Beispielen, die die Identität nicht erwähnen, verwechselt. Dementsprechend sagt das Modell für toxische Beispiele, in denen die Identität erwähnt wird, wahrscheinlich niedrigere Toxizitätswerte voraus, als es sollte. Diese toxischen Beispiele würden wahrscheinlich bei vielen Schwellenwerten in der Untergruppe als falsch Negative erscheinen.

4.3.4 Die drei AUC-Metriken

Die gemeinsame Betrachtung dieser drei vorgestellten Metriken für eine beliebige Identitätsuntergruppe zeigt, inwieweit das untersuchte Modell die Beispiele in den Testdaten nicht korrekt zuordnet und ob diese Fehlordnungen wahrscheinlich zu falsch positiven oder falsch negativen Ergebnissen führen, wenn ein Schwellenwert ausgewählt wird.

Eine wichtige Eigenschaft der AUC-Metrik besteht darin, dass sie gegenüber Datenungleichgewichten hinsichtlich der Anzahl negativer und positiver Beispiele im Testdatensatz robust ist. Dies ist insbesondere bei der Messung von unbeabsichtigtem Bias wichtig, da in realen Daten die Menge der Beispiele in jeder Identitätsuntergruppe und das Gleichgewicht zwischen negativen und positiven Beispielen zwischen den Gruppen stark variieren kann. Wenn man für jeden AUC erzwingt, dass entweder alle negativen oder alle positiven Beispiele oder beim Subgroup AUC alle negativen und positiven Beispiele aus einer Identitätsuntergruppe stammen, führt das dazu, dass Fehlordnungen, die diese bestimmte Untergruppe betreffen, nicht durch Ergebnisse aus anderen Gruppen übertönt werden können. Dadurch wird sichergestellt, dass diese Metriken robust gegenüber Datenungleichgewichten sind, die in realen Daten wahrscheinlich auftreten.

Zusammengefasst können die verschiedenen Metriken kombiniert werden, um verschiedene Arten von Bias zu erkennen. Der Subgroup AUC und die BPSN und BNSP AUCs identifizieren jeden Bias der signifikant genug ist, um eine falsche Zuordnung von negativen und positiven Beispielen zu verursachen (Borkan, Dixon et al., 2019; Jigsaw / Conversation AI, 2019).

4.3.5 Verallgemeinerter Mittelwert der Bias AUCs

Die vorgestellten Bias AUCs werden jeweils für die einzelnen Identitätsuntergruppen berechnet. Anschließend werden diese Submetriken zu einem Gesamtmaß kombiniert. Dafür wird der verallgemeinerte Mittelwert von den Bias AUCs berechnet:

$$M_p(m_s) = \left(\frac{1}{N}\sum_{s=1}^{N} m_s^p\right)^{\frac{1}{p}} \tag{4.4}$$

Dabei ist M_p das p-te Mittel, m_s ist die für die Untergruppe s berechnete Bias-Metrik m und N ist die Anzahl der Identitätsuntergruppen. Für den Wettbewerb wurde ein Wert von -5 für p verwendet, um die Teilnehmer zu ermutigen, das Modell für die Identitätsuntergruppen mit der niedrigsten Modellleistung zu verbessern. Je niedriger der Wert für p gewählt wird, desto mehr ziehen die Untergruppen mit den niedrigsten Leistungen den Mittelwert nach unten.

4.3.6 Endgültige Metrik

Für die endgültige Metrik wird der Gesamt-AUC des Modells AUC_{gesamt} mit den verallgemeinerten Mittelwerten der Bias-AUCs kombiniert. Auf diese Weise wird der Wert für die endgültige Modellbewertung berechnet:

$$score = \omega_0 AUC_{gesamt} + \sum_{a=1}^{A} \omega_a M_p\left(m_{s,a}\right) \tag{4.5}$$

Dabei ist A die Anzahl der Submetriken (in diesem Fall also 3), $m_{s,a}$ ist die Bias-Metrik für die Identitätsuntergruppe s unter Verwendung der Submetrik a, ω_0 ist eine Gewichtung für den Gesamt-AUC und ω_a ist eine Gewichtung für die jeweilige Relevanz der einzelnen Submetriken. Alle vier ω-Werte, also die Gewichtung für den Gesamt-AUC plus die Gewichtung für die drei Submetriken, werden auf 0,25 gesetzt. Dieser finale Wert wurde für die Rangliste im Wettbewerb verwendet. Jedoch sind neben diesem Wert die Ergebnisse der Submetriken ebenfalls relevant und sollten für die endgültige Beurteilung des jeweiligen Modells betrachtet werden (Jigsaw / Conversation AI, 2019).

5 Implementierung

In diesem Kapitel wird die Implementierung der verschiedenen Modelle für die Klassifikation von Kommentaren hinsichtlich ihrer Toxizität beschrieben. Dafür werden zuerst die Daten vorverarbeitet. Anschließend wird das Training der Modelle mit verschiedenen Verfahren durchgeführt. Hierfür werden die Verfahren Naive Bayes, Entscheidungsbaum, Random Forest, logistische Regression und ein Convolutional Neural Network (CNN) verwendet.

5.1 Daten vorverarbeiten

Bevor mit dem eigentlichen Training der Modelle begonnen werden kann, müssen die Daten vorverarbeitet werden, sowohl die Trainingsdaten als auch die Testdaten. Die Vorverarbeitung ist ein wesentlicher Schritt, bei dem die Daten bereinigt werden (Frochte, 2019). In Abschnitt 3.3 wurden die Daten bereits visualisiert und auf ihre Vollständigkeit überprüft, weshalb dieser Schritt nicht mehr durchgeführt werden muss.

Die Vorverarbeitung der Daten sowie das Training der Modelle werden mit Python durchgeführt, welches zahlreiche Funktionen und Klassen für den Umgang mit Textdaten bietet. Für die Vorverarbeitung der Daten werden die Bibliotheken *NumPy*, *Pandas*, *Scikit-Learn* und *Keras* verwendet. NumPy ist insbesondere bei der Arbeit mit mehrdimensionalen Arrays hilfreich (Hirschle, 2022). Pandas bietet gute Werkzeuge für den Umgang mit großen Datensätzen. Vor allem bei der Vorverarbeitung und bei dem Umgang mit Textdaten werden durch Pandas gute Werkzeuge zur Verfügung gestellt. Scikit-Learn bietet viele traditionelle Werkzeuge für das maschinelle Lernen. Bei Keras handelt es sich um eine Open-Source-Bibliothek für neuronale Netzwerke (Choo et al., 2020; Frochte, 2019).

Als Word Embedding wird GloVe von Pennington et al. (2014) verwendet. Es handelt sich um ein Word Embedding, das mit Daten von Common Crawl trainiert wurde und 42 Milliarden Tokens, ein Vokabular in der Größe von 1,9 Millionen und 300-dimensionale Vektoren beinhaltet. Es ist zudem *uncased*, der Text wurde dementsprechend in Kleinbuchstaben umgewandelt.

Für das Training der Modelle müssen die Daten an einigen Stellen unterschiedlich vorbereitet werden. Für das CNN wird GloVe als Word Embedding verwendet, während für die anderen Modelle mit dem Bag-of-Words-Ansatz gearbeitet wird. Durch die Anwendung

dieser verschiedenen Ansätze müssen die Textdaten für GloVe und das BoW-Modell jeweils unterschiedlich vorverarbeitet werden. Für den BoW-Ansatz müssen, wie in Abschnitt 2.4.1 beschrieben, die Stopwords entfernt und die Wörter in die Wortstämme oder Grundformen umgewandelt werden. Darüber hinaus werden Symbole wie „?“ und die Zahlen entfernt. Für GloVe ist dies nicht der Fall, da die Symbole und Zahlen größtenteils bekannt sind und nur teilweise entfernt werden müssen. Bei der Vorverarbeitung für GloVe werden jedoch auch für das BoW-Modell nützliche Schritte durchgeführt. Aus diesem Grund werden in dieser Arbeit die Daten zunächst für GloVe, bis auf ein paar finale Schritte, vorbereitet und anschließend werden für das BoW-Modell die Textdatenanteile, die für das BoW-Modell störend sind, entfernt.

Die Vorverarbeitung wird sowohl für die Trainingsdaten als auch für die Testdaten durchgeführt. Bei der Vorverarbeitung wird zunächst ein Vokabular aus den Kommentartexten erstellt und überprüft, wie viel des Vokabulars und des gesamten Textes das GloVe-Modell abdeckt. Anfangs wurden Embeddings für 8,92 % des Vokabulars und 78,80 % des ganzen Textes gefunden. Anschließend wird mit den folgenden Schritten versucht für möglichst große Anteile des Vokabulars und des Textes Embeddings zu finden. Weil es sich um ein uncased Modell handelt, werden alle Kommentare in Kleinbuchstaben umgewandelt.

Danach werden Kurzformen wie „isn’t“ oder „let’s“ in ihre Langformen wie „is not“ oder „let us“ umgeformt. Zudem werden alle Zeichen, mit denen wahrscheinlich ein Apostroph gemeint ist, in ein einheitliches Apostroph umgewandelt. Für GloVe unbekannte Symbole wie das Wurzelzeichen „√“, werden in eine für GloVe erkennbare Form umgeändert, wie bei diesem Beispiel „sqrt“. Zudem werden zwischen den Symbolen und den restlichen Textteilen Leerzeichen eingefügt. Dadurch wird erreicht, dass Worte neben denen sich direkt Symbole befinden, nicht mehr als gesondertes Wort gezählt werden. Andernfalls werden „yes,“ und „yes“ als gesonderte Worte gezählt und auch von GloVe würde „yes,“ nicht als Embedding gefunden werden. Zudem werden häufig falsch geschriebene Wörter korrigiert, beispielsweise „theguardian“ zu „the guardian“. Zuletzt werden unbekannte Symbole wie beispielsweise „*“ gelöscht und Leerzeichen entfernt, wenn mehr als eines in direkter Folge vorkommt. Ein Beispiel für das Ergebnis der Vorverarbeitung im Vergleich zum Text vor der Vorverarbeitung ist in Tabelle 5.1 dargestellt. Durch diese Schritte werden Embeddings für 59,23 % des Vokabulars und für 99,77 % des ganzen Textes gefunden.

Tabelle 5.1: Vorverarbeitung der Kommentare für GloVe

Vor der Vorverarbeitung	Nach der Vorverarbeitung
„This is so cool. It’s like, ’would you want your mother to read this??’ Really great idea, well done!“	„this is so cool . it is like , ’ would you want your mother to read this ? ? ’ really great idea , well done ! “

Für das BoW-Modell werden anschließend die Buchstaben, Zahlen und Stopwords entfernt, so dass nur die eigentlichen Wörter bestehen bleiben. Zusätzlich wird der PorterStemmer aus dem Natural Language Toolkit (NLTK) verwendet, welcher morphologische Affixe aus den Wörtern entfernt, so dass nur der Wortstamm übrig bleibt (NLTK Project, 2023). In Tabelle 5.2 ist ein Beispiel für das Ergebnis dieses Vorgangs dargestellt. Als Beispiel wurde erneut der Kommentar aus Tabelle 5.1 nach der Vorverarbeitung verwendet. Für die Umsetzung des BoW-Modells wird der CountVectorizer von Scikit-Learn verwendet, welcher die Daten in eine Wort-Dokument-Matrix umwandelt. In dieser ist verzeichnet, wie häufig jedes Wort aus dem gesamten Korpus in jedem Dokument vorkommt (Hirschle, 2022).

Tabelle 5.2: Vorverarbeitung der Kommentare für Bag-of-Words

Vor der Vorverarbeitung	Nach der Vorverarbeitung
„this is so cool . it is like , ' would you want your mother to read this ? ? ' really great idea , well done ! "	„cool like would want mother read realli great idea well done"

Nach diesem Vorgang ist die Vorverarbeitung der Textdaten fast abgeschlossen. Es fehlen noch die finalen Schritte für GloVe bzw. für das CNN. Die Kommentare werden noch mit dem Tokenizer von Keras auf die gleiche Länge gekürzt oder aufgefüllt. Dabei wird eine maximale Länge aller Sequenzen von 250 verwendet. Bevor das Training der Modelle beginnt, werden alle Toxizitätswerte und alle Identitätswerte in binäre Werte umgewandelt. Alle Werte größer oder gleich 0,5 werden zu „True" und alle Werte kleiner als 0,5 werden zu „False". Der Grenzwert von 0,5 für die Toxizität und die Identitäten wurde ebenfalls im Wettbewerb von Jigsaw / Conversation AI (2019) verwendet. Für das CNN werden diese Toxizitätswerte mit to_categorical von Keras in eine binäre Klassenmatrix umgewandelt.

5.2 Modelle trainieren

Die Verfahren für das Training der Modelle wurden in Abschnitt 2.6 vorgestellt. Zuerst werden vier Modelle mit den Verfahren Naive Bayes, Entscheidungsbaum, Random Forest und Logistische Regression trainiert. Dafür wird erneut die Bibliothek Scikit-Learn verwendet. Alle vier Verfahren werden mit Hilfe dieser Bibliothek implementiert. Im vorherigen Schritt wurden die Daten bereits als Bag-of-Words vorverarbeitet, welcher für diese Verfahren verwendet wird. Für das Training des Naive Bayes-Modells wird der BernoulliNB verwendet, welcher sich besonders für die binäre Klassifikation eignet. Das Entscheidungsbaum-Modell wird mit dem DecisionTreeClassifier trainiert, das Random Forest-Modell mit dem RandomForestClassifier und das Logistische Regressions-Modell mit dem LogisticRegression-Klassifikator.

Für das Convolutional Neural Network (CNN) wird das Benchmark Kernel von Borkan, Elliott et al. (2019) als Grundlage verwendet. Für dieses Modell wird *Keras* als Bibliothek verwendet. Bevor das Modell trainiert werden kann, werden die Trainingsdaten in eine Trainingsmenge und eine Validierungsmenge aufgeteilt. Dabei werden 80 % der Daten zum Training und 20 % zur Validierung verwendet. Bei neuronalen Netzen wird auf der Trainingsmenge die Backpropagation durchgeführt. Zudem werden auf dieser Menge die Gewichte angepasst. Die Daten der Validierungsmenge werden zur Auswahl des besten Netzwerkes aus der Menge an Netzwerken verwendet. Dies ist das Netzwerk mit der besten Leistung auf diesen Daten. Darüber hinaus werden mit Hilfe der Validierungsmenge bei Bedarf die Parameter angepasst. Die Testdaten werden nicht beim Training verwendet und dienen nach dem Training zur Beurteilung der Leistung des Modells (Frochte, 2019).

Als Embedding wird GloVe verwendet, welches bereits in Abschnitt 5.1 vorgestellt wurde. Es handelt sich um ein Embedding mit 300-dimensionalen Vektoren. Die Dropout-Rate wird auf 0,3 und die Lernrate auf 0,00005 festgelegt. Es wird eine Batch-Größe von 128 verwendet. Das Modell wird in 10 Durchläufen bzw. Epochen trainiert. Nach diesen 10 Epochen gibt es keine wesentlichen Verbesserungen mehr, weshalb diese Anzahl für das Modell ausreichend ist. Der *loss*, also der Wert der Kostenfunktion für die Trainingsdaten, konnte von 0,1774 in der ersten Epoche auf 0,1341 in der zehnten Epoche verbessert werden.

6 Resultate

In diesem Kapitel werden die trainierten Modelle aus dem vorangegangenen Kapitel hinsichtlich eines unbeabsichtigten Bias untersucht. Hierfür werden die Metriken aus Abschnitt 4.3 verwendet. Es werden jeweils die Subgroup AUCs, BPSN (Background Positive, Subgroup Negative) AUCs und die BNSP (Background Negative, Subgroup Positive) AUCs für alle Modelle betrachtet. Diese werden werden für die jeweiligen Identitätsuntergruppen untersucht: Male, Female, Homosexual Gay or Lesbian, Christian, Jewish, Muslim, Black, White und Psychiatric or Mental Illness. Im Folgenden wird die Untergruppe „Homosexual Gay or Lesbian“ mit „Homosexual“ und die Untergruppe „Psychiatric or Mental Illness“ mit „Mental Illness“ abgekürzt. Zudem werden die verallgemeinerten Mittelwerte und die finalen Metriken für alle Modelle betrachtet. Anschließend werden die Modelle hinsichtlich des unbeabsichtigten Bias in einer Übersicht miteinander verglichen.

6.1 Untersuchung der Modelle

Bevor die Modelle miteinander verglichen werden, erfolgt eine separate Betrachtung der Ergebnisse für jedes Modell. Es wird jedes Modell einzeln auf unbeabsichtigte Bias untersucht. Auf diese Weise kann die Leistung der einzelnen Modelle besser bewertet werden. Für die jeweiligen Spalten mit den Subgroup AUC-Werten, den BPSN AUC-Werten und den BNSP AUC-Werten wurde jeweils der höchste und der niedrigste Wert des Modells fett gedruckt.

6.1.1 Naive Bayes

Die Ergebnisse des Naive Bayes-Modells sind in Tabelle 6.1 dargestellt. Zudem wurden für das Naive Bayes-Modell der folgende ROC AUC-Wert für das gesamte Modell und die folgende endgültige Metrik berechnet:

- Gesamt AUC des Modells: 0,720258
- Endgültige Metrik: 0,704581

Wie aus der Tabelle 6.1 hervorgeht, weisen alle drei Bias AUCs sehr unterschiedliche Werte für die einzelnen Identitätsuntergruppen auf. Der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe beträgt beim Subgroup AUC 0,106472, beim BPSN AUC 0,144528 und beim BNSP AUC 0,112987. Bei diesem Modell ist insbesondere der große Unterschied zwischen den BPSN AUC-Werten sehr auffällig. Der verallgemeinerte Mittelwert für den BPSN AUC ist ebenfalls der geringste von den drei Mittelwerten und der Wert für die Identität „White“ beim BPSN AUC ist mit 0,562458 der niedrigste in der ganzen Tabelle. Ein niedriger Wert beim BPSN AUC bedeutet, dass das Modell wahrscheinlich höhere Toxizitätswerte für nicht-toxische Beispiele, die die Identität erwähnen, vorhersagt, als es sollte. In diesem Fall erscheinen wahrscheinlich nicht-toxische Beispiele als falsch Positive. Dies ist relativ wahrscheinlich der Fall für die Identität „White“.

Tabelle 6.1: Ergebnisse des Naive Bayes-Modells

Identität	Größe	Subgroup AUC	BPSN AUC	BNSP AUC
Male	4386	0,737568	0,665826	0,789976
Female	5155	0,728833	0,674050	0,773705
Homosexual	1065	**0,637264**	0,649485	**0,708575**
Christian	4226	0,703525	**0,706986**	0,717181
Jewish	835	**0,743736**	0,641824	**0,821562**
Muslim	2040	0,682953	0,635327	0,767126
Black	1519	0,665415	0,596589	0,787535
White	2452	0,688786	**0,562458**	0,841980
Mental Illness	511	0,712520	0,692527	0,740176
Verallg. Mittelwert	22189	0,695141	0,637822	0,765100

Zudem ist bei diesem Modell auffällig, dass viele Identitäten, insbesondere beim BPSN AUC, schlechter als der Gesamt AUC des Modells abschneiden. Besser ist das Modell insgesamt beim BNSP AUC. Hierbei handelt es sich um den höchsten der drei verallgemeinerten Mittelwerte. Viele Identitätsuntergruppen in dieser Spalte schneiden zwar unterschiedlich ab, der verallgemeinerte Mittelwert ist aber besser als die endgültige Metrik und auch besser als der AUC des gesamten Modells. Beim BNSP AUC ist der Wert für die Identität „Homosexual“ am niedrigsten. Dies ist ebenfalls der Fall beim Subgroup AUC. Außerdem ist auffällig, dass der BNSP AUC hinsichtlich des verallgemeinerten Mittelwerts einen höheren Wert hat als der AUC des gesamten Modells.

6.1.2 Entscheidungsbaum

Die Ergebnisse des Naive Bayes-Modells sind in Tabelle 6.2 dargestellt. Zudem wurden für das Entscheidungsbaum-Modell der folgende ROC AUC-Wert für das gesamte Modell und die folgende endgültige Metrik berechnet:

- Gesamt AUC des Modells: 0,720305
- Endgültige Metrik: 0,666583

Auch bei diesem Modell weisen die drei Bias AUCs unterschiedliche Werte für die einzelnen Identitätsuntergruppen auf. Der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe beträgt beim Subgroup AUC 0,105057, beim BPSN AUC lediglich 0,060944 und beim BNSP AUC 0,083989. Bei diesem Modell hat der Subgroup AUC den niedrigsten verallgemeinerten Mittelwert und auch der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe ist beim Subgroup AUC der größte. Der Subgroup AUC für die Identität „Black“ ist zudem der niedrigste in der gesamten Tabelle. Ein niedriger Subgroup AUC bedeutet, dass das Modell schlecht zwischen toxischen und nicht-toxischen Kommentaren, die die spezifische Identität erwähnen, unterscheiden kann. Dies ist relativ wahrscheinlich der Fall für die Identität „Black“.

Tabelle 6.2: Ergebnisse des Entscheidungsbaum-Modells

Identität	Größe	Subgroup AUC	BPSN AUC	BNSP AUC
Male	4386	0,659402	0,699308	0,682611
Female	5155	0,650969	0,703791	0,670331
Homosexual	1065	0,590489	0,656801	0,655520
Christian	4226	0,623732	**0,711685**	0,634966
Jewish	835	0,591770	0,697688	**0,615426**
Muslim	2040	0,607511	0,677401	0,652996
Black	1519	**0,566024**	0,653473	0,636101
White	2452	0,591921	**0,650741**	0,665008
Mental Illness	511	**0,671081**	0,692180	**0,699415**
Verallg. Mittelwert	22189	0,611472	0,680344	0,654212

Außerdem sind die verallgemeinerten Mittelwerte alle niedriger als der AUC des gesamten Modells. Insgesamt sind die Unterschiede zwischen dem jeweils höchsten und niedrigsten

Wert der Bias AUCs bei diesem Modell nicht so groß wie beim Naive Bayes-Modell, die einzelnen Identitätsuntergruppen schneiden jedoch deutlich schlechter ab, insbesondere im Vergleich zum AUC des gesamten Modells. Keine einzige Identitätsuntergruppe ist beim Wert des Subgroup AUCs so hoch, wie der AUC des gesamten Modells. Am besten schneidet im Verhältnis die Untergruppe „Mental Illness“ ab, sowohl beim Subgroup AUC als auch beim BNSP AUC handelt es bei ihr um den jeweils höchsten Wert.

6.1.3 Random Forest

Die Ergebnisse für das Random Forest-Modell sind in Tabelle 6.3 dargestellt. Zudem wurden für das Random Forest-Modell der folgende ROC AUC-Wert für das gesamte Modell und die folgende endgültige Metrik berechnet:

- Gesamt AUC des Modells: 0,641967
- Endgültige Metrik: 0,593821

Bei diesem Modell weisen die drei Bias AUCs ebenfalls unterschiedliche Werte für die einzelnen Identitätsuntergruppen auf. Der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe beträgt beim Subgroup AUC 0,042604, beim BPSN AUC lediglich 0,010585 und beim BNSP AUC 0,049146. Diese Unterschiede sind im Vergleich zu den vorherigen Modellen relativ gering.

Tabelle 6.3: Ergebnisse des Random Forest-Modells

Identität	Größe	Subgroup AUC	BPSN AUC	BNSP AUC
Male	4386	0,566565	0,646983	0,565004
Female	5155	0,554353	**0,648498**	0,551980
Homosexual	1065	0,534271	0,644984	0,533329
Christian	4226	**0,525460**	0,647208	**0,523462**
Jewish	835	0,529022	0,642266	0,529737
Muslim	2040	0,546799	0,645022	0,546816
Black	1519	0,533334	0,645033	0,533909
White	2452	0,552040	0,646144	0,552385
Mental Illness	511	**0,568064**	**0,637913**	**0,572608**
Verallg. Mittelwert	22189	0,544326	0,644853	0,544138

Auch wenn die Unterschiede in den Werten der einzelnen Identitätsuntergruppen bei den Bias AUCs nicht groß sind, so ist dieses Modell insgesamt nicht gut. Die einzelnen AUC-Werte liegen sehr an einem Wert von 0,5. Ein Wert von 0,5 beim ROC AUC würde zufälliges Raten bedeuten. Außerdem ist der verallgemeinerte Mittelwert für den Subgroup AUC deutlich niedriger als der AUC für das gesamte Modell. Dies deutet darauf hin, dass das Modell schlechter zwischen toxischen und nicht-toxischen Kommentaren, die eine der spezifischen Identitäten erwähnen, unterscheiden kann, als das Modell insgesamt. Der verallgemeinerte Mittelwert des BNSP AUCs ist ebenfalls sehr niedrig und nicht weit von dem Wert 0,5 entfernt. Dies deutetet darauf hin, dass das Modell für toxische Beispiele, in denen die jeweilige Identität erwähnt wird, wahrscheinlich niedrigere Toxizitätswerte voraussagt als es sollte und dass diese nicht-toxischen Beispiele wahrscheinlich in den jeweiligen Untergruppen als falsch Positive erscheinen werden.

6.1.4 Logistische Regression

Die Ergebnisse für das Logistische Regressions-Modell sind in Tabelle 6.4 dargestellt. Außerdem wurden für das Logistische Regressions-Modell der folgende ROC AUC-Wert für das gesamte Modell und die folgende endgültige Metrik berechnet:

- Gesamt AUC des Modells: 0,721644
- Endgültige Metrik: 0,694289

Auch bei diesem Modell weisen die drei Bias AUCs unterschiedliche Werte für die einzelnen Identitätsuntergruppen auf. Der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe beträgt beim Subgroup AUC 0,079066, beim BPSN AUC 0,046204 und beim BNSP AUC 0,047935. Wie aus diesen Werten zu erkennen, ist bei diesem Modell insbesondere der größere Unterschied zwischen den Subgroup AUC-Werten sehr auffällig. Zudem weist der Subgroup AUC den niedrigsten verallgemeinerten Mittelwert auf, welcher deutlich niedriger ist als der AUC des gesamten Modells.

Die Identitätsuntergruppe „Homosexual“ schneidet bei den Subgroup AUC-Werten besonders schlecht ab und ist ebenfalls beim BNSP AUC der niedrigste Wert der Spalte. Bei der Identitätsuntergruppe mit den höchsten Werten beim Subgroup AUC und auch beim BNSP AUC handelt es sich um „Female“. Der Unterschied von 0,079066 beim Subgroup AUC zwischen „Female“ und „Homosexual“ ist jedoch nicht ganz so groß wie bei einigen der vorherigen Modelle. Außerdem ist bei diesem Modell auffällig, dass die höchsten Werte der jeweiligen Spalten kleiner sind, als der AUC des gesamten Modells. Dementsprechend schneiden diese Untergruppen im Vergleich zum gesamten Modell relativ schlecht ab.

Tabelle 6.4: Ergebnisse des Logistischen Regression-Modells

Identität	Größe	Subgroup AUC	BPSN AUC	BNSP AUC
Male	4386	0,701337	0,707394	0,716156
Female	5155	**0,707620**	0,711280	**0,718417**
Homosexual	1065	**0,628554**	0,680879	**0,670482**
Christian	4226	0,696180	**0,715807**	0,702685
Jewish	835	0,653255	0,694144	0,681229
Muslim	2040	0,647441	0,696752	0,674106
Black	1519	0,647714	**0,669603**	0,700769
White	2452	0,667680	0,681647	0,708740
Mental Illness	511	0,671966	0,698245	0,695598
Verallg. Mittelwert	22189	0,666089	0,694167	0,695257

6.1.5 Convolutional Neural Network (CNN)

Die Ergebnisse für das Convolutional Neural Network (CNN) sind in Tabelle 6.5 dargestellt. Zudem wurden für das CNN der folgende ROC AUC-Wert für das gesamte Modell und die folgende endgültige Metrik berechnet:

- Gesamt AUC des Modells: 0,936840
- Endgültige Metrik: 0,888840

Auch hier weisen die drei Bias AUCs unterschiedliche Werte für die einzelnen Identitätsuntergruppen auf. Der Unterschied zwischen der besten und der schlechtesten Identitätsuntergruppe beträgt beim Subgroup AUC 0,109711, beim BPSN AUC 0,143108 und beim BNSP AUC 0,032234. Besonders auffällig sind hierbei die hohen Unterschiede zwischen den niedrigsten und den höchsten Werten für die jeweiligen Identitätsuntergruppen des Subgroup AUCs und des BPSN AUCs. Lediglich der die Werte des BNSP AUC weisen keinen großen Unterschied auf. Bei einer alleinigen Betrachtung des AUCs für das gesamte Modell und der endgültigen Metrik schneidet das CNN am besten von den untersuchten Modellen ab. Auch die verallgemeinerten Mittelwerte liegen jeweils bei über 0,8 und schneiden im Vergleich zu den anderen Modellen gut ab. Umso auffälliger sind die Unterschiede in den Werten der einzelnen Identitätsuntergruppen, insbesondere beim Subgroup AUC und beim BPSN AUC.

Zwischen den Identitätsuntergruppen „Christian“ und „Black“ besteht bei den Werten des BPSN AUCs ein Unterschied von 0,143108. Die Untergruppe „Christian“ schneidet somit deutlich besser ab als die Untergruppe „Black“. Der BPSN AUC bedeutet, dass das Modell wahrscheinlich höhere Toxizitätswerte für nicht-toxische Beispiele, die die Identität erwähnen, vorhersagt, als es sollte. Dies geschieht in diesem Fall häufiger bei der Identität „Black“ als bei der Identität „Christian“. Allgemein schneidet die Identität „Christian“ in diesem Modell sehr gut ab und weist bei allen drei Bias AUCs die höchsten Werte auf. Die Identitätsuntergruppen „Homosexual“ und „Black“ weisen in diesem Modell beim Subgroup AUC und auch beim BPSN AUC jeweils die niedrigsten bzw. die zweitniedrigsten Werte auf.

Tabelle 6.5: Ergebnisse des CNNs

Identität	Größe	Subgroup AUC	BPSN AUC	BNSP AUC
Male	4386	0,882221	0,879783	0,944180
Female	5155	0,882412	0,894552	0,933038
Homosexual	1065	**0,793141**	0,778829	0,950145
Christian	4226	**0,902852**	**0,916497**	**0,930508**
Jewish	835	0,882457	0,858701	0,958900
Muslim	2040	0,815510	0,810708	0,950527
Black	1519	0,795324	**0,773389**	0,955432
White	2452	0,810170	0,778138	**0,962742**
Mental Illness	511	0,871733	0,840884	0,954843
Verallg. Mittelwert	22189	0,842298	0,827650	0,948572

6.2 Vergleichen der Modelle

Nachdem für die einzelnen Modelle jeweils die Subgroup AUC-Werte, BPSN AUC-Werte, BNSP AUC-Werte, die verallgemeinerten Mittelwerte und die endgültigen Metriken vorgestellt und auf Unterschiede in den Ergebnissen für die jeweiligen Identitätsuntergruppen untersucht wurden, werden in diesem Abschnitt die Modelle miteinander verglichen. Dies geschieht anhand ihrer Subgroup AUC-Werte, ihrer BPSN AUC-Werte und ihrer BNSP AUC-Werte. Zudem werden die verallgemeinerten Mittelwerte und die endgültigen Metriken miteinander verglichen.

6.2.1 Subgroup AUC

Der in Abschnitt 4.3.1 vorgestellte Subgroup AUC berechnet den ROC AUC-Wert für eine Untergruppe, in diesem Fall die jeweilige im Kommentar enthaltene Identität einer Person. Ein niedriger Wert bedeutet, dass das Modell schlecht zwischen toxischen und nicht-toxischen Kommentaren, die diese Identität erwähnen, unterscheiden kann. Tabelle 6.6 zeigt die Ergebnisse für alle untersuchten Modelle.

Tabelle 6.6: Subgroup AUC für alle Modelle

Identität	Naive Bayes	Entscheidungsbaum	Random Forest	Logistische Regression	CNN
Male	0,737568	0,659402	0,566565	0,701337	0,882221
Female	0,728833	0,650969	0,554353	**0,707620**	0,882412
Homosexual	**0,637264**	0,590489	0,534271	**0,628554**	**0,793141**
Christian	0,703525	0,623732	**0,525460**	0,696180	**0,902852**
Jewish	**0,743736**	0,591770	0,529022	0,653255	0,882457
Muslim	0,682953	0,607511	0,546799	0,647441	0,815510
Black	0,665415	**0,566024**	0,533334	0,647714	0,795324
White	0,688786	0,591921	0,552040	0,667680	0,810170
Mental Illness	0,712520	**0,671081**	**0,568064**	0,671966	0,871733
Verallg. Mittelw.	0,695141	0,611472	0,544326	0,666089	0,842298

Vergleicht man die Identitätsuntergruppen anhand der Subgroup AUC-Werte für alle Modelle, so wird besonders deutlich, dass die Identitätsuntergruppe „Homosexual“ besonders häufig die niedrigsten Werte aufweist und damit am schlechtesten abschneidet. Dies ist wenig überraschend, weil diese Untergruppe bereits bei den einzelnen Modellen häufiger niedrige Werte aufgewiesen hat. Bei den Modellen Entscheidungsbaum und Random Forest weist „Homosexual“ nicht die niedrigsten Werte auf, jedoch handelt es sich ebenfalls um vergleichsweise niedrige Werte für das jeweilige Modell. In Abschnitt 3.3 in der Abbildung 3.3 ist zu erkennen, dass es sich bei der Identitätsuntergruppe „Homosexual“ um die Untergruppe mit der drittgrößten Toxizität handelt. Eine höhere Toxizität in Verbindung mit Kommentaren zu der jeweiligen Untergruppe weisen nur die Identitäten „White“ und „Black“ auf. Besonders „Black“ hat auch beim Subgroup AUC im Vergleich sehr niedrige Werte erhalten. Die Untergruppe „White“ hat jedoch eher durchschnittliche Werte im Vergleich zu den anderen Untergruppen.

In der Abbildung 3.3 hat zudem die Identitätsuntergruppe „Christian“ die niedrigsten Toxizitätswerte erhalten. Dies spiegelt sich teilweise in der Tabelle 6.6 wieder. Die Untergruppe „Christian“ hat beim CNN den höchsten Wert und beim Entscheidungsbaum-Modell und beim Logistischen Regressions-Modell vergleichsweise hohe Werte erhalten. Beim Random Forest-Modell handelt es sich jedoch um den niedrigsten Wert, auch wenn es sich nicht um einen großen Unterschied zum höchsten Wert handelt. Auch die Untergruppe „Mental Illness“ zeigt im Vergleich bei einigen Modellen hohe Werte.

Insgesamt lassen sich aus der Tabelle 6.6 recht große Unterschiede in den Werten für die jeweiligen Identitätsuntergruppen ablesen. Die Modelle weisen dementsprechend einen unbeabsichtigten Bias auf. Bei vielen Modellen schneidet die Identität „Homosexual“ schlechter ab als viele andere Identitäten. Die niedrigen Werte bedeuten in diesem Fall, dass die Modelle schlecht zwischen toxischen und nicht-toxischen Kommentaren, die die Identität „Homosexual“ erwähnen, unterscheiden kann. Diese Unterscheidung fällt bei beispielsweise der Untergruppe „Christian“ sehr viel besser aus.

6.2.2 BPSN (Background Positive, Subgroup Negative) AUC

Bei dem in Abschnitt 4.3.2 vorgestellten BPSN (Background Positive, Subgroup Negative) AUC wird der ROC AUC-Wert für die nicht-toxischen Beispiele, die die Identität erwähnen, und die toxischen Beispiele, die die Identität nicht erwähnen, berechnet. Ein niedriger Wert bedeutet, dass das Modell nicht-toxische Beispiele, die die Identität erwähnen, mit toxischen Beispielen verwechselt, die die Identität nicht erwähnen. Dadurch sagt das Modell wahrscheinlich höhere Toxizitätswerte für nicht-toxische Beispiele, die die Identität erwähnen, vorher, als es sollte. Tabelle 6.7 zeigt die Ergebnisse für alle Modelle.

In dieser Tabelle ist sehr auffällig, dass erneut die Identitätsuntergruppe „Christian“ die höchsten Werte aufweist. Die Werte für den BPSN AUC sind lediglich beim Random Forest-Modell nicht die höchsten. Jedoch ist ebenfalls beim Random Forest-Modell der Wert für „Christian“ im Vergleich recht hoch. Die niedrigsten Werte weisen in dieser Tabelle insbesondere die Identitätsuntergruppen „White“ und „Black“ auf. Bei diesen handelt es sich ebenfalls in der Abbildung 3.3 um die Identitäten mit den höchsten Toxizitätswerten.

Auch aus dieser Tabelle lässt sich insofern ein unbeabsichtigter Bias ablesen. Bestimmte Identitätsuntergruppen wie „Christian“ schneiden bei vielen Modellen besser ab, als Untergruppen wie „White“ und „Black“. Die Modelle sagen für diese schlechteren Identitätsuntergruppen wahrscheinlich höhere Toxizitätswerte für nicht-toxische Beispiele, die diese Identitätsuntergruppen erwähnen, vorher, als sie sollten. Bei Identitätsuntergruppen mit höheren Werten wie „Christian“ ist dies nicht so häufig der Fall wie bei den Identitätsuntergruppen mit den deutlich niedrigeren Werten.

Tabelle 6.7: BPSN (Background Positive, Subgroup Negative) AUC für alle Modelle

Identität	Naive Bayes	Entscheidungsbaum	Random Forest	Logistische Regression	CNN
Male	0,665826	0,699308	0,646983	0,707394	0,879783
Female	0,674050	0,703791	**0,648498**	0,711280	0,894552
Homosexual	0,649485	0,656801	0,644984	0,680879	0,778829
Christian	**0,706986**	**0,711685**	0,647208	**0,715807**	**0,916497**
Jewish	0,641824	0,697688	0,642266	0,694144	0,858701
Muslim	0,635327	0,677401	0,645022	0,696752	0,810708
Black	0,596589	0,653473	0,645033	**0,669603**	**0,773389**
White	**0,562458**	**0,650741**	0,646144	0,681647	0,778138
Mental Illness	0,692527	0,692180	**0,637913**	0,698245	0,840884
Verallg. Mittelw.	0,637822	0,680344	0,644853	0,694167	0,827650

6.2.3 BNSP (Background Negative, Subgroup Positive) AUC

Beim in Abschnitt 4.3.3 vorgestellten BNSP (Background Negative, Subgroup Positive) AUC wird der ROC AUC-Wert für die toxischen Beispiele, die die Identität erwähnen, und die nicht-toxischen Beispiele, die die Identität nicht erwähnen, berechnet. Ein niedriger Wert bedeutet, dass das Modell toxische Beispiele, die die Identität erwähnen, mit nicht-toxischen Beispielen, die die Identität nicht erwähnen, verwechselt. Dementsprechend sagt das Modell wahrscheinlich niedrigere Toxizitätswerte für toxische Beispiele, in denen die Identität erwähnt wird, vorher, als es sollte. Tabelle 6.8 zeigt die Ergebnisse für alle Modelle.

Die in dieser Tabelle dargestellten Ergebnisse unterscheiden sich deutlich von den Ergebnissen in den anderen Tabellen. Die Unterschiede zwischen dem jeweils höchsten und dem niedrigsten Wert für die jeweilige Identitätsuntergruppe sind, außer beim BNSP AUC-Wert für das Naive Bayes-Modell, vergleichsweise gering. Dies gilt insbesondere für das Random Forest-Modell, das Logistische Regressions-Modell und das CNN. Außerdem lässt sich auf den ersten Blick keine einzelne Identitätsuntergruppe mit den geringsten oder den höchsten Werten bei allen Modellen erkennen. Bei der Identität „Homosexual“ handelt es sich bei zwei Modellen um den niedrigsten Wert, während die anderen Modelle in dieser Untergruppe durchschnittlich abschneiden. Auch die Identitätsuntergruppe „Christian“ besitzt zwei der niedrigsten Werte. Diese Identitätsuntergruppe hatte in den vorherigen Tabellen vergleichsweise deutlich besser abgeschnitten.

Tabelle 6.8: BNSP (Background Negative, Subgroup Positive) AUC für alle Modelle

Identität	Naive Bayes	Entscheidungsbaum	Random Forest	Logistische Regression	CNN
Male	0,789976	0,682611	0,565004	0,716156	0,944180
Female	0,773705	0,670331	0,551980	**0,718417**	0,933038
Homosexual	**0,708575**	0,655520	0,533329	**0,670482**	0,950145
Christian	0,717181	0,634966	**0,523462**	0,702685	**0,930508**
Jewish	**0,821562**	**0,615426**	0,529737	0,681229	0,958900
Muslim	0,767126	0,652996	0,546816	0,674106	0,950527
Black	0,787535	0,636101	0,533909	0,700769	0,955432
White	0,841980	0,665008	0,552385	0,708740	**0,962742**
Mental Illness	0,740176	**0,699415**	**0,572608**	0,695598	0,954843
Verallg. Mittelw.	0,765103	0,654212	0,544138	0,695257	0,948572

Bei der Identitätsuntergruppe „Jewish“ handelt es sich beim Entscheidungsbaum-Modell um den höchsten Wert und beim Random Forest-Modell um den niedrigsten. Aus diesen Ergebnissen lässt sich somit nicht die beste oder die schlechteste Untergruppe herausarbeiten. Jedoch besteht trotzdem ein Unterschied in den Werten der jeweiligen Modelle. Dieser wurde bereits bei der Untersuchung der einzelnen Modelle in Abschnitt 6.1 aufgezeigt. So besteht beispielsweise beim Naive Bayes-Modell ein großer Unterschied zwischen dem niedrigstem und dem höchstem Wert. Dementsprechend sagt das Naive Bayes-Modell wahrscheinlich niedrigere Toxizitätswerte für toxische Beispiele, in denen die Identität „Homosexual“ erwähnt wird, vorher, als es sollte. Die Untergruppe „Jewish“ hat einen sehr viel höheren Wert, wodurch diese Untergruppe vergleichsweise besser abschneidet. Daraus lässt sich ebenfalls unbeabsichtigter Bias vergleichsweise gut erkennen.

6.2.4 Endgültige Metrik

Die Werte dieser drei Bias AUCs werden jeweils zu einem Gesamtmaß kombiniert. Dafür wird, wie in Abschnitt 4.3.5 vorgestellt, der verallgemeinerte Mittelwert berechnet. Diese drei Werte für die jeweiligen Modelle sind in Tabelle 6.9 unter Subgroup AUC, BPSN AUC und BNSP AUC dargestellt. Diese werden gemeinsam mit dem ROC AUC-Wert des gesamten Modells zu einer endgültigen Metrik zusammengerechnet, wie es in Abschnitt 4.3.6 beschrieben wurde.

Die ROC AUC-Werte für das gesamte Modell und die endgültigen Werte der Modelle sind ebenfalls in der Tabelle 6.9 dargestellt.

Wie aus der Tabelle 6.9 hervorgeht, hat das CNN mit einem großen Abstand die höchste endgültige Metrik. Sowohl die verallgemeinerten Mittelwerte als auch der AUC-Wert des gesamten Modells sind höher als bei den anderen Modellen. Am zweitbesten hat das Naive Bayes-Modell abgeschnitten, dicht gefolgt vom Logistischen Regressions-Modell. Beim Random Forest-Modell handelt es sich um das Modell mit der geringsten endgültigen Metrik. Obwohl das CNN vergleichsweise eine gute endgültige Metrik erhalten hat, konnte in den vorherigen Abschnitten aufgezeigt werden, dass die verschiedenen Identitätsuntergruppen auch bei diesem Modell sehr unterschiedliche Werte aufweisen und dementsprechend einen unbeabsichtigten Bias beinhalten.

Tabelle 6.9: Berechnung der endgültigen Metrik für alle Modelle

Metrik	Naive Bayes	Entscheidungsbaum	Random Forest	Logistische Regression	CNN
Subgroup AUC	0,695141	0,611472	0,544326	0,666089	0,842298
BPSN AUC	0,637822	0,680344	0,644853	0,694167	0,827650
BNSP AUC	0,765103	0,654212	0,544138	0,695257	0,948572
Gesamt AUC des Modells	0,720258	0,720305	0,641967	0,721644	0,936840
Endgültige Metrik	0,704581	0,666583	0,593821	0,694289	0,888840

Insgesamt konnten für alle Modelle Identitätsuntergruppen herausgearbeitet werden, die im Vergleich besser bzw. schlechter als andere Identitätsuntergruppen im jeweiligen Modell abschneiden. Nach der Definition aus Abschnitt 2.5.2 von Dixon et al. (2018) enthält ein Modell zur Textklassifikation einen unbeabsichtigten Bias, wenn es bei Kommentaren über bestimme Gruppen besser abschneidet als bei Kommentaren über andere Gruppen. Dies ist bei den hier analysierten Modellen gegeben. Somit konnte unbeabsichtigter Bias bei allen Untersuchten Modellen herausgearbeitet werden.

6.3 Mögliche Verbesserungen

In dieser Arbeit wurden lediglich eine begrenzte Anzahl an Verfahren für das Training der Modelle verwendet. In anderen Arbeiten mit diesem Datensatz wurden bereits weitere Ansätze

getestet. Reina (2019) hat ein Modell mit BERT trainiert. Bei BERT (Bidirectional Encoder Representations from Transformer) handelt es sich um eine Variante des Transformers (Görz et al., 2021), welcher in dieser Arbeit nicht eingeführt wurde. Mit BERT konnte eine endgültige Metrik von 0,929878 erreicht werden, welche um 0,041038 höher ist als der Wert des in dieser Arbeit trainierten CNNs. Jedoch enthält auch dieses Modell deutlichen unbeabsichtigten Bias. Der niedrigste Subgroup AUC-Wert liegt bei 0,842154 für „Homosexual“ und der höchste Wert bei 0,955801 für „Mental Illness“. Das ergibt einen deutlichen Unterschied von 0,113647. In der Zukunft könnten dementsprechend weitere Verfahren für diesen Datensatz ausprobiert bzw. bestehende Modelle verbessert werden.

Außerdem gibt es bereits erste Ansätze für die Beseitigung oder zumindest Reduzierung des unbeabsichtigten Bias. Der primäre Ansatz besteht darin, einen Text durch Wortsubstitution zu verändern, um auf diese Weise gegensätzliche Fälle in den Trainingsdaten zu erzeugen. Diese generierten Szenarien können dann verwendet werden, um das Lernen des Modells zu korrigieren bzw. positiv zu beeinflussen, entweder indem die Einbettungen gegenüber den Störungen resistent gemacht werden oder indem der Unterschied in den Vorhersagen zwischen den richtigen und den gestörten Szenarien minimiert wird. Ein anderer Ansatz besteht darin, Begriffe, die zum Bias beitragen, während des Trainings auszublenden. Dieser Ansatz könnte jedoch dazu führen, dass durch das Ausblenden der Begriffe semantische Lücken entstehen. Abgesehen von dem bisher erläuterten traditionellen Bias in Modellen kann Bias auch auf einer höheren Ebene entstehen, wenn Forschungsentscheidungen getroffen werden. Ein einfaches Beispiel dafür ist die Tendenz, auf englischsprachige Korpora zurückzugreifen, vor allem aufgrund der größeren Popularität und breiteren Akzeptanz. Durch diese Tendenzen wird die Forschung zu Randthemen und das Studieren anderer Sprachen vernachlässigt (Poria et al., 2020).

7 Fazit

In der vorliegenden Arbeit wurde jeweils ein Modell mit den Verfahren Naive Bayes, Entscheidungsbaum, Random Forest, Logistische Regression und mit einem Convolutional Neural Network (CNN) trainiert. Für das Training der Modelle wurde ein Datensatz mit Kommentaren verwendet, der auf das Vorkommen verschiedener Personengruppen im jeweiligen Kommentartext bewertet wurde. Diese Modelle wurden anschließend auf unbeabsichtigten Bias untersucht, welcher die Genauigkeit der Vorhersagen dieser Modelle für Kommentare zu diesen Personengruppen negativ beeinflussen kann.

Mit Hilfe von verschiedenen Metriken konnte gezeigt werden, dass die trainierten Modelle unterschiedlich starken unbeabsichtigten Bias aufweisen. Mit dem Subgroup AUC-Werten konnte aufgezeigt werden, dass die Identitätsuntergruppe „Homosexual“ bei den Modellen im Vergleich zu den anderen Untergruppen sehr niedrige Werte aufweist und somit schlechter abschneidet. Auch die Identitäten „Black“ und „White“ erhielten im Vergleich zu den anderen Identitätsuntergruppen niedrige Werte beim Subgroup AUC. Die Identitätsuntergruppe „Christian“ schnitt im Vergleich zu diesen Identitäten deutlich besser ab. Dieses Muster konnte konnte ebenfalls in den Werten des BPSN (Background Positive, Subgroup Negative) AUCs nachgewiesen werden. Auch bei dieser Metrik erhielten die Identitäten „Black“ und „White“ deutlich niedrigere Werte als die besser abschneidende Identität „Christian“. In den Werten des BNSP (Background Negative, Subgroup Positive) AUCs konnte dieses Muster nicht für alle Modelle nachgewiesen werden. Allerdings konnte mit dieser Metrik ebenfalls unbeabsichtigter Bias in den Modellen aufgezeigt werden, auch wenn keine bestimmte Identitätsuntergruppe für alle Modelle hervorstach.

Von den trainierten Modellen hat das CNN bei der endgültigen Metrik am besten abgeschnitten. Jedoch konnte durch die Untersuchung der einzelnen Identitäten mit den Werten des Subgroup AUCs, des BPSN AUCs und des BNSP AUCs ebenfalls unbeabsichtigter Bias nachgewiesen werden. Die Modelle, die mit den Verfahren Naive Bayes, Logistische Regression und Entscheidungsbaum trainiert wurden, schnitten in der endgültigen Metrik im Vergleich deutlich schlechter ab. Das Random Forest Modell schnitt in der finalen Metrik am schlechtesten ab. Auch in diesen Modellen konnte ein unbeabsichtigter Bias herausgearbeitet werden.

In zukünftigen Arbeiten können andere Verfahren, die in dieser Arbeit nicht verwendet wurden, zum Training der Modelle verwendet werden, um diese ebenfalls auf unbeabsichtig-

ten Bias zu untersuchen. In der vorliegenden Arbeit wurde zudem nur mit einem einzigen Datensatz und ebenfalls nur einem Word Embedding gearbeitet. Für zukünftige Arbeiten können weitere Erkenntnisse über unbeabsichtigte Bias gewonnen werden, indem andere Datensätze auf unbeabsichtigte Bias untersucht werden und die Ergebnisse für verschiedene Identitätsuntergruppen verglichen werden. Die Verwendung unterschiedlicher Word Embeddings, die ebenfalls ein Grund für unbeabsichtigte Bias sein können, ist ebenfalls interessant für zukünftige Arbeiten.

Ein weiteres wichtiges Thema für zukünftige Arbeiten ist die Beseitigung des unbeabsichtigten Bias. Auf diese Weise können in Zukunft Systeme zur Textklassifikation für alle Personengruppen gerechter gestaltet werden. Dadurch könnten Systeme geschaffen werden, die mit wenig oder gar keiner menschlicher Hilfe die Toxizität von Kommentaren bewerten, ohne dabei Kommentare von bestimmten Personengruppen als toxischer einzustufen als Kommentare von anderen Personengruppen.

Literatur

Biemann, C., Heyer, G., & Quasthoff, U. (2022). *Wissensrohstoff Text: Eine Einführung in das Text Mining* (2. überarb. Aufl.). Springer. https://doi.org/10.1007/978-3-658-35969-0

Borkan, D., Dixon, L., Sorensen, J., Thain, N., & Vasserman, L. (2019). Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. *Companion Proceedings of The 2019 World Wide Web Conference*, 491–500. https://doi.org/10.1145/3308560.3317593

Borkan, D. [@dborkan], Elliott, J. [@juliaelliott], inversion [@inversion], Sorensen, J. [@sorenj], Vasserman, L. [@lucyvasserman], & nithum [@nithum]. (2019). *Benchmark Kernel*. Verfügbar 25. Juli 2023 unter https://www.kaggle.com/code/dborkan/benchmark-kernel

Brownlee, J. (2021). *How to use ROC curves and precision-recall curves for classification in Python*. Verfügbar 17. August 2023 unter https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Choo, K., Greplova, E., Fischer, M. H., & Neupert, T. (2020). *Machine Learning kompakt: Ein Einstieg für Studierende der Naturwissenschaften*. Springer. https://doi.org/0.1007/978-3-658-32268-7

Cleve, J., & Lämmel, U. (2020). *Data Mining* (3. Aufl.). De Gruyter. https://doi.org/10.1515/9783110676273

Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and Mitigating Unintended Bias in Text Classification. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 67–73. https://doi.org/10.1145/3278721.3278729

Frochte, J. (2019). *Maschinelles Lernen: Grundlagen und Algorithmen in Python* (2. akt. Aufl.). Hanser.

Géron, A. (2020). *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme. Aktuell zu TensorFlow 2* (K. Rother & T. Demmig, Übers.; 2. Aufl.). O'Reilly. (Original erschienen 2019)

Ghosh, S., Kumar, S., Lepcha, S., & Jain, S. S. (2021). Toxic Text Classification. In D. S. Jat, S. Shukla, A. Unal & D. K. Mishra (Hrsg.), *Data Science and Security* (S. 251–260). Springer. https://doi.org/10.1007/978-981-15-5309-7_27

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Stanford CS224N Project Report*.

Görz, G., Schmid, U., & Braun, T. (Hrsg.). (2021). *Handbuch der Künstlichen Intelligenz* (6. Aufl.). De Gruyter. https://doi.org/10.1515/9783110659948

Hirschle, J. (2022). *Deep Natural Language Processing: Einstieg in Word Embedding, Sequence-to-Sequence-Modelle und Transformer mit Python*. Hanser.

Internet World Stats. (2023). *Internet Growth Statistics: Today's road to e-Commerce and Global Trade Internet Technology Reports*. Verfügbar 22. Juni 2023 unter https://www.internetworldstats.com/emarketing.htm

Jigsaw / Conversation AI. (2019). *Jigsaw Unintended Bias in Toxicity Classification: Detect toxicity across a diverse range of conversations*. Verfügbar 21. Juni 2023 unter https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification

Kemp, S. (2023). *Digital 2023 April Global Statshot Report*. Verfügbar 24. Juni 2023 unter https://datareportal.com/reports/digital-2023-april-global-statshot

Kiritchenko, S., & Mohammad, S. M. (2018). Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems. https://arxiv.org/abs/1805.04508

Liu, Z., Lin, Y., & Sun, M. (2020). *Representation Learning for Natural Language Processing*. Springer. https://doi.org/10.1007/978-981-15-5573-2

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150.

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *AAAI Conference on Artificial Intelligence*.

Merriam-Webster. (2023). Race. In *Merriam-Webster.com*. Verfügbar 21. August 2023 unter https://www.merriam-webster.com/dictionary/race

Mueller, J., & Massaron, L. (2020). *Deep Learning kompakt für Dummies* (S. Linke, Übers.). Wiley. (Original erschienen 2019)

NLTK Project. (2023). *Documentation*. Verfügbar 1. September 2023 unter https://www.nltk.org/howto/stem.html

NZ [@nz0722]. (2019). *Simple EDA Text Preprocessing - Jigsaw*. Verfügbar 25. Juli 2023 unter https://www.kaggle.com/code/nz0722/simple-eda-text-preprocessing-jigsaw

Ortiz-Ospina, E. (2019). The rise of social media. *Our World in Data*. Verfügbar 23. Juni 2023 unter https://ourworldindata.org/rise-of-social-media

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://aclanthology.org/D14-1162/

Poria, S., Hazarika, D., Majumder, N., & Mihalcea, R. (2020). Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research. *IEEE Transactions on Affective Computing*. https://arxiv.org/abs/2005.00357

Raschka, S., & Mirjalili, V. (2021). *Machine Learning mit Python und Keras, TensorFlow 2 und Scikit-learn: Das umfassende Praxis-Handbuch für Data Science, Deep Learning*

und Predictive Analytics (K. Lorenzen, Übers.; 3. akt. und erw. Aufl.). mitp. (Original erschienen 2019)

Reddit Inc. (2022). *Revealing This Year's Reddit Recap, Where We Highlight How Redditors Kept It Real in 2022*. Verfügbar 22. Juni 2023 unter https://www.redditinc.com/blog/reddit-recap-2022-global

Reina, Y. [@yuval6967]. (2019). *Toxic BERT plain vanila*. Verfügbar 29. August 2023 unter https://www.kaggle.com/code/yuval6967/toxic-bert-plain-vanila

Risch, J., & Krestel, R. (2020). Toxic Comment Detection in Online Discussions. In B. Agarwal, R. Nayak, N. Mittal & S. Patnaik (Hrsg.), *Deep Learning-Based Approaches for Sentiment Analysis* (S. 85–109). Springer. https://doi.org/10.1007/978-981-15-1216-2_4

Ullmann, S., & Tomalin, M. (2020). Quarantining online hate speech: technical and ethical perspectives. *Ethics Inf Technol*, *22*, 69–80. https://doi.org/10.1007/s10676-019-09516-z

Vogels, E. A. (2021). *The State of Online Harassment*. Pew Research Center.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel

Systematische Analyse von unbeabsichtigtem Bias in der Textklassifikation

selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Hamburg, 8. September 2023

Kirsten Grahl