



Datengenerator für Daten mit Bias als Grundlage für Data Science Projekte

Studienarbeit

für die Prüfung zum
Bachelor of Science

des Studiengangs Informatik
an der Dualen Hochschule Baden-Württemberg Stuttgart

von
Simon Jess, Timo Zaoral

Juni 2022

Bearbeitungszeitraum
Matrikelnummer, Kurs
Betreuer

04.10.2021 - 10.06.2022
8268544, 6146532, INF19C
Prof. Dr. Monika Kochanowski

Erklärung

Wir versicherern hiermit, dass wir die vorliegende Studienarbeit mit dem Thema: *Datengenerator für Daten mit Bias als Grundlage für Data Science Projekte* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, Juni 2022

Simon Jess

Timo Zaoral

Abstract

Fasst die Aufgabenstellung und Ergebnisse kompakt und übersichtlich in wenigen Zeilen zusammen (4-7 Zeilen).

Inhaltsverzeichnis

Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
1 Einleitung	1
1.1 Motivation	2
1.2 Zielsetzung	3
1.3 Aufbau der Arbeit	4
2 Stand der Technik	5
2.1 Daten als wertschöpfende Ressource	5
2.1.1 Daten	5
2.1.2 Datenqualität	7
2.2 Künstliche Intelligenz	9
2.2.1 Künstliche Intelligenz Allgemein	9
2.2.2 Teilgebiet maschinelles Lernen	10
2.2.3 Ethik in der künstlichen Intelligenz	16
2.3 Bias im Zusammenhang mit Künstliche Intelligenz (KI)	18
2.3.1 Bias	18
2.3.2 Diskriminierung durch Bias in Daten	18
2.3.3 Gegenmaßnahmen	18
3 Praktischer Teil	19
3.1 Szenarien	19
3.1.1 Szenario 1	19

3.1.2	Szenario 2	20
3.2	Konzeption	21
3.2.1	Grobkonzept	21
3.2.2	Feinkonzept	22
3.3	Umsetzung	25
3.3.1	Umsetzung des Szenario 1	25
3.3.2	Umsetzung des Szenario 2	35
3.4	Datenauswertung	46
3.5	Evaluation der Ergebnisse	46
4	Schluss	47
4.1	Zusammenfassung	47
4.2	Diskussion	47
4.3	Ausblick	47

Abkürzungsverzeichnis

KI	Künstliche Intelligenz
ML	Machine Learning
bzw.	beziehungsweise
M	Männlich
W	Weiblich
dt	deutsch

Abbildungsverzeichnis

2.1	Weltweit jährlich anfallende Datenmenge [3]	5
2.2	Big Data 3 V rule [4]	6
2.3	Data Quality Dimensions [4]	7
2.4	Exemplarisch dargestellte Disziplinen von KI[6]	9
2.5	Arten des maschinellen Lernen[6]	11
2.6	Der Prozess von überwachtem Lernen [17]	12
2.7	Veränderung der Performance durch die Datenmengen [18]	13
2.8	Overfitting und Underfitting [6]	14
2.9	Scheinkorrelationen in der Bilderkennung [19]	14
2.10	Gartner Hype Cycle für KI [22]	16
2.11	Ethische und rechtliche Anforderungen an KI [20]	17
3.1	Programmablaufplan der fünf Hauptschritte zur Generierung der Daten . .	22
3.2	Verbindungen zwischen den Attributen des zweiten Szenario	23
3.3	Erste Zelle Code des Szenario1	26
3.4	Codezeile zur Bestimmung des Geschlechts einer Person	27
3.5	Codezeilen zum Erstellen eines Dictionary mit den zur Bewertung relevanten Attributen	29
3.6	Programmablaufplan zur Generierung der Regeln von Szenario 1	30
3.7	Methode zur Initialisierung eines Bewertenden	31
3.8	Methode eines Bewertenden zum Bewerten von Anträgen.	32
3.9	Dictionaries gefüllt mit Standard Werten des Szenario 1	34
3.10	Letzte Zelle des Szenario 1 für die Benutzenden Interaktion	35
3.11	Codezeile zur Auswahl der Ausprägung der Wohnlage basierend auf angegebenen Wahrscheinlichkeiten	39

3.12	Codeausschnitt zum Hinzufügen eines Dateneintrags	40
3.13	Codeausschnitt zum Erstellen des Regel Dictionary	41
3.14	Codeausschnitt der Funktion zum Bewerten von Personen	42
3.15	Codeausschnitt für das Hinzufügen eines Bias beim Bewerten der Personen	43
3.16	Codeausschnitt für das Hinzufügen eines Bias beim Bewerten der Personen	44

Tabellenverzeichnis

2.1	Data Quality Dimensions Merkmale	8
3.1	Tabelle für die Auswirkung der Attributen von Szenario 1	20
3.2	Tabelle der Attribute und Auswirkungen von Szenario 2	20
3.3	Tabelle zur Bestimmung der Wahrscheinlichkeiten für das Geschlecht . . .	27
3.4	Tabelle der Wahrscheinlichkeiten für die Härte der Strafe nach Geschlecht .	28
3.5	Tabelle zur Bestimmung der Wahrscheinlichkeiten für die Hautfarbe unter Berücksichtigung des Geschlechts	28
3.6	Tabelle zur Bestimmung der Wahrscheinlichkeiten für die Altersgruppen . .	36
3.7	Tabelle der Wahrscheinlichkeiten für die Politische Orientierung nach Alter	37
3.8	Tabelle der Wahrscheinlichkeiten für den Bildungsabschluss nach Alter . .	38
3.9	Tabelle der Wahrscheinlichkeiten für das Soziale Engagement nach Politi- scher Orientierung	38
3.10	Tabelle der Wahrscheinlichkeiten für die Wohnlage	39

1 | Einleitung

Die fortschreitende Digitalisierung ist kaum noch aus unserem Alltag wegzudenken. Durch immer mehr Programme, die den Alltag erleichtern sollen, nutzen wir die Errungenschaften der Digitalisierung täglich. Häufig ist hier die Rede von künstlicher Intelligenz. Dabei ist uns meist nicht einmal Bewusst, dass im Hintergrund mit künstlicher Intelligenz gearbeitet wird. Egal ob als intelligenten Routenplaner oder Sprachsteuerung, hinter all diese Anwendung steckt heute nicht mehr nur ein Optimierungsalgorithmus sondern KI.

Mit der Digitalisierung hat man begonnen große Datenmengen zu sammeln. Durch den technischen Fortschritt im Bereich von Big Data, werden diese Datenmengen heutzutage unvorstellbar groß. Mit dem Erfassen und Speichern von Daten ist man in der Lage seine Produkte stetig zu verbessern und zudem neue Geschäftsmodelle zu schaffen. Zu diesen neuen Geschäftsmodellen gehört die nicht mehr aus unserem Alltag wegzudenkende KI. Sie ist in der Lage Entscheidungen und Vorhersagen auf Basis von Daten zu treffen, die durch einen Menschen nur mit großem Aufwand getätigt werden können. Egal ob eine Entscheidung oder eine Vorhersage von einer KI getroffen wird, sie basiert auf Daten der Vergangenheit. Aus diesem Grund sind Daten, sobald sie verarbeitet und genutzt werden, eine so wertvolle Ressource.

Für eine KI werden Daten zum Lernen genutzt. Entscheidend für die Qualität der KI ist somit die Datengrundlage auf der die KI basiert. Lernen bedeutet, dass Zusammenhänge und die dadurch abgebildeten Verhaltensweisen in den Daten von der KI erkannt und gelernt werden. Durch diese Art des Lernens, wie auch wir Menschen lernen, ergeben sich jedoch nicht nur Potentiale sondern auch Risiken. Abhängig von der Datenqualität und Richtigkeit bzw. Zuverlässigkeit der Daten werden zukünftige Entscheidungen und Vorhersagen getroffen. Eine KI betrachtet dabei die Daten vollkommen neutral ohne Hintergrundwissen und ethische Wertvorstellungen. Für manche Entscheidungen gibt es jedoch nicht zwingend Richtig oder Falsch. Häufig ist es ein schmaler Grad dazwischen. In diesen Fällen wird das menschliche Handeln durch Ethik gesteuert. Eine KI besitzt jedoch keine Ethik und so können Entscheidungen einer KI durch unterschiedliche Ursachen benachteiligend oder gar diskriminierenden sein.

Durch KI öffnen sich viele neue Möglichkeiten und Geschäftsmodelle. Sie wird in immer mehr Bereichen eingesetzt. Doch wenn eine KI vor moralischen Entscheidungen steht sollte man bedenken, dass eine Maschine keine Ethik und Moral besitzt. Dies kann zu fatalen Fehlentscheidungen führen und „the dark side of KI“ zum Vorschein bringen.

1.1 Motivation

Mit den Vorteilen der KI kommen immer auch Nachteile. Um die Schattenseite einer KI verstehen zu können, muss man das Thema KI etwas genauer betrachten. Eine KI ist meist ein Instrument zur Vorhersage oder Erkennung. Die Entscheidungen werden durch maschinelles Lernen getroffen. Beim Maschinellen lernen werden, vereinfacht gesagt, Verhaltensweisen und Zusammenhänge in Daten analysiert und diese für zukünftige Entscheidungen als Vorlage genutzt. Die besondere Eigenschaft hierbei ist, dass die Daten, auch Trainingsdaten genannt, Daten aus der Vergangenheit sind. Das Lernen funktioniert ähnlich wie bei uns Menschen, die KI bekommt Trainingsdaten die zeigen, wie Sie zu Entscheiden hat und übernimmt diese Verhaltensweise. Da eine KI auf diese Art und weiße lernt und Entscheidungen trifft, ist naheliegend, dass es wie beim Menschen durch diese Form des Lernens auch ungewünschte Effekte gibt. Bei uns Menschen lernen wir in der Regel von den Eltern, die einen erziehen. Bei einer KI sind die Eltern die Daten, die Verhaltensweisen beibringen.

Bei der KI und speziell dem Machine Learning (ML) ergeben sich mehrere zu berücksichtigende Probleme. Das häufigste Problem des ML ist das Under- und Overfitting. Dabei wird entweder zu wenig aus den Trainingsdaten gelernt und deshalb willkürlich entschieden oder die Trainingsdaten werden „auswendig“ gelernt und deshalb bei neuen Daten willkürlich entschieden.

Ein unbekannteres Problem von KI und ML ist die Verzerrung in den Trainingsdaten. Wenn Trainingsdaten aufgrund unterschiedlichster Ursachen unerwünschte Zusammenhänge beinhalten, wird von Bias gesprochen. So können zum Beispiel Entscheidungen aufgrund eines unbekannten Zusammenhang in den Trainingsdaten, häufig auf diskriminierenden Verhaltensmustern, basieren. Die Problematik liegt darin, dass den Endnutzer in der Regel nicht bekannt ist, dass es einen Bias in den Daten geben kann. In den meisten Fällen ist eine solche Verzerrung verborgen und wird erst im produktiven Betrieb der KI festgestellt.

Diese Verzerrungen führen meist zu Skandalen in der Medienwelt. Es wurde bereits diverse Male in der Presse darüber berichtet, dass bspw. in Unternehmen Bewerbungen durch ein KI vorsortiert wurden und dabei Personen mit Migrationshintergrund aus nicht nachvollziehbaren Gründen aussortiert wurden. Ein solches diskriminierendes Verhaltensmuster wurde daraufhin in den Trainingsdaten erkannt.

Diese Diskriminierungen sind jedoch nicht zu vergessen immer auf Trainingsdaten und so in der Regel auf reale Daten aus der Vergangenheit zurückzuführen. Das Problem des Bias in Daten ist daher, durch menschliches Verschulden, eine Schattenseite der KI

1.2 Zielsetzung

KI ist in allen Lebensbereichen vorhanden und auch nicht mehr wegzudenken. Jedoch die Schattenseite der KI, ist den meisten Menschen unbekannt. Dabei spielt die Ethik eine besondere Rolle, denn im Gegensatz zu uns Menschen, verfügt eine KI nicht über ethische Werte und Moral. Häufig spielt die Ethik jedoch in der Entscheidungsfindung eine nicht zu vernachlässigende Rolle. Die Folge aus der fehlenden Ethik bei einer KI kann zu Fehlentscheidungen und fatalen Folgen führen.

Aus diesem Grund soll mehr Bewusstsein für Bias in Daten geschaffen werden. Insbesondere die Entwickler von KI Lösungen müssen für die Thematik mehr sensibilisiert werden, sodass mögliche Benachteiligungen nicht erst in der Praxis festgestellt werden. Dafür soll ein Datengenerator, welcher Daten mit Bias erzeugt entwickelt werden. Um diese Daten in der Lehre einsetzen zu können soll zusätzlich eine Auswertung entwickelt werden, welche den Bias als Visualisierung veranschaulicht.

Die Umsetzung liegt den folgenden Anforderungen zugrunde:

- Konzeption zweier Szenarien, die realitätsnah sind.
- Erstellung eines Datengenerators für zufallsgenerierte Daten.
 - Python Script zum generieren eines großen Datensets
 - Erzeugung eines Bias durch die Bewertung des Datensets
 - Bewertete Daten als CSV Datei bereitstellen
- Erstellung einer Auswertung zur Veranschaulichung des Bias.
 - Visuelle Auswertung in Tableau

Ziel ist es, einen Datengenerator für Daten mit Bias zu entwickeln und zusätzlich eine visualisierte Auswertung, die den Bias veranschaulicht. Dieser soll in der Lehre zum Einsatz kommen und für die Thematik von Bias in Daten sensibilisieren.

1.3 Aufbau der Arbeit

Der erste Abschnitt ist in drei Passagen aufgeteilt. Zu Beginn wird das allgemeine Thema der Daten als Grundlage für KI betrachtet. Dabei wird insbesondere auf die Datenqualität eingegangen. Des weiteren wird das Thema Bias, also die Verzerrung in den Daten, auf Basis der Literatur veranschaulicht. In der folgenden Passage wird auf KI und ML eingegangen. Ebenso wird die Ethik in der KI betrachtet. Die letzte Passage setzt sich dann mit Bias in KI Trainingsdaten auseinander. Dabei liegt der Fokus auf der möglicherweise entstehenden Diskriminierung. Im Gegensatz dazu werden zusätzlich Ansätze und Konzepte von Gegenmaßnahmen betrachtet. Im nächsten großen Abschnitt wird die praktische Umsetzung des Datengenerators näher betrachtet. Dafür werden zu Beginn die zwei Szenarien ausgearbeitet und näher beschrieben. Als nächstes werden die daraus entstehenden Anforderungen in Form eines Konzepts aufgestellt. Dieses unterscheidet sich in Fein und Grobkonzept und beschreibt die logischen Funktionen. Daraufhin folgt die Implementierung des beschriebenen Konzepts. Anschließend folgt die in den Anforderungen geforderte Auswertung der generierten Daten. Dazu wird die erstellte Auswertung in Tableau herangezogen. Zum Schluss dieses Abschnitts wird das Ergebnis des Datengenerators und der Auswertung vorgestellt und evaluiert. Abschließend werden alle Erkenntnisse gesammelt und zusammengefasst. Hier wird auch das Ergebnis der Arbeit kritisch reflektiert und evaluiert. Beendet wird die Arbeit mit einem Ausblick darüber, welche Relevanz Bias in der KI zukünftig haben könnte.

2 | Stand der Technik

In diesem Kapitel wird der Stand der Technik näher beleuchtet. Der Fokus liegt dabei auf den Themen Daten, KI und Bias. Zu Beginn wird auf basis der Literatur erläutert, was Daten sind, was Datenqualität bedeutet und worum es sich bei einem Bias handelt. Daraufhin wird näher auf KI, das Teilgebiet ML und die Ethik in der KI eingegangen. Zuletzt werden die Themen in einen gemeinsamen Kontext gebracht und der Einfluss eines Bias auf eine KI betrachtet sowie Gegenmaßnahmen untersucht.

2.1 Daten als wertschöpfende Ressource

2.1.1 Daten

Dass Daten eine wertvolle Ressource seien, meinte bereits 2006 der britische Mathematiker Clive Humby mit dem berühmten Zitat: „Data is the new oil“.[1] Hiermit ist gemeint, dass Daten in ihre Rohform nicht sonderlich wertvoll sind, diese jedoch an Wert gewinnen, sobald man beginnt sie zu verarbeiten. Denn lange Zeit waren Daten nur ein Nebenprodukt der Digitalisierung. Daten wurde gesammelt und gespeichert, aber nicht weiter verwendet. Mit dem technologischen Fortschritt im Bereich von Datenanalysen und mit aufkommen der KI wurden Daten von Zeit zu Zeit immer wertvoller. So wurden neue Datengetriebene Geschäftsfeld ermöglicht, die einen Mehrwert aus Rohdaten schaffen können. Insbesondere das rasante Aufkommen des Internet of Things hat diese Entwicklung stark vorangetrieben. Seither steigt die Menge der jährlich gesammelten Daten exponentiell an.[2]

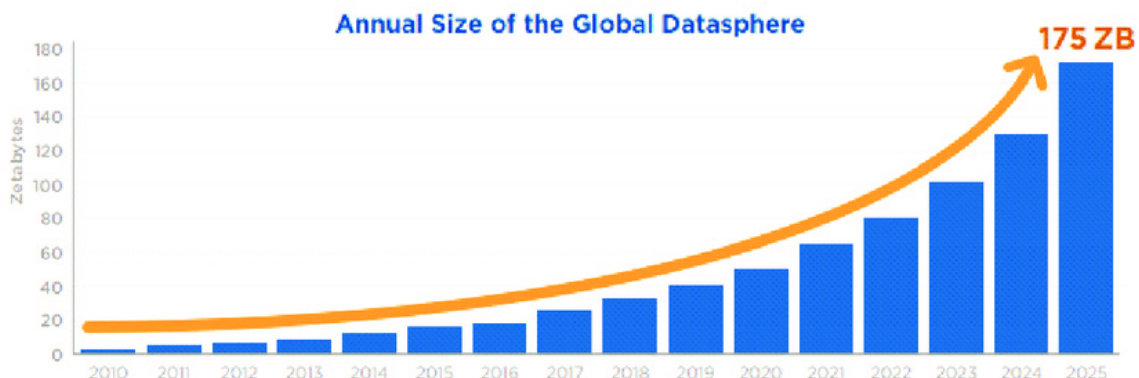


Abbildung 2.1: Weltweit jährlich anfallende Datenmenge [3]

Abbildung 2.1 stammt aus dem Jahr 2018 und verdeutlicht, dass bereits damals erwartet wurde, dass bis im Jahr 2025 rund 175 Zetabyte Daten jährlich gesammelt werden. Im Vergleich dazu waren es 2018 gerade einmal 33 Zetabyte weltweit.[3][4]

Der Begriff Daten selbst wird im ISO2382-1 Standard wie folgt definiert: „reinterpretierbare Darstellung von Informationen in einer formalisierten Weise, die für die Kommunikation, Interpretation oder Verarbeitung geeignet ist“.[5] Daraus lässt sich ableiten, dass Daten Informationen der Vergangenheit repräsentieren und für zukünftige Verwendung die Informationen aus der Vergangenheit in einer einheitlichen Form repräsentieren. Damit ist jedoch nicht die einheitliche Form der Daten selbst gemeint.

Daten gibt es in unterschiedlichen Formen. Es wird zwischen strukturierten und unstrukturierten Daten unterschieden. Strukturierte Daten sind Datensätze bestehend aus einzelnen Variablen die eindeutige Größen darstellen. Beispiel hierfür sind Sensordaten oder Unternehmenszahlen aus einem ERP System. Sie werden tabellarisch gespeichert und können einfach weiter verarbeitet werden. Oftmals sind diese Daten heterogen, was bedeutet, dass sich die Variablen unterscheiden und bspw. Spalte 1 vollkommen andere Daten beinhaltet als Spalte 2. Ein Beispiel hierfür wären Sensoren für Luftfeuchtigkeit und Helligkeit in einem Büro. Als unstrukturierte Daten bezeichnet man Daten, die nicht in sinnvolle einheitliche Variablen unterteilt werden können. Zu dieser Art von Daten zählt man Bilder, Videos, Audio und Textdaten. Sie sind meist homogen, denn die Pixel in einem Bild nehmen zwar unterschiedliche RGB Werte an, jedoch repräsentieren sie alle einen Pixel. Dabei ist es egal ob es sich um Pixel 1 oder Pixel 42 handelt.[6] Bei dieser unvorstellbar großen Datenmenge die jährlich generiert wird, wird davon ausgegangen, dass rund 80% als unstrukturierte Daten vorliegen.[2]

Häufig wird in dem Zusammenhang mit Daten auch von Big Data gesprochen. Eine einheitliche Definition für den Begriff Big Data existiert jedoch nicht. Denn der Begriff Big Data umfasst die gesamte Wertschöpfungskette. Diese beinhaltet die Datenerzeugung, das Sammeln und Speichern der Daten bis hin zur Verarbeitung und Nutzung für Analysen oder Visualisierungen.[4][7] Es handelt sich dabei um Informationen mit hohem Volumen (high-volume), hoher Geschwindigkeit (high-velocity) und hoher Vielfalt (high-variety). Diese drei charakteristischen Eigenschaften werden in der Literatur auch als die „3 V rule“ bezeichnet und ist in den meisten Definitionen wiederzufinden.[4][8]

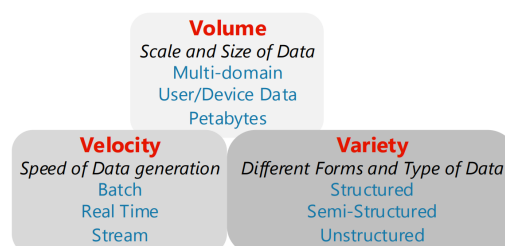


Abbildung 2.2: Big Data 3 V rule [4]

In der Abbildung 2.2 werden diese Eigenschaften die Big Data mit sich bringt näher beschrieben. Unter „Volume“ wird die große Menge an Daten, also der damit verbundene benötigte Speicher, und ihre Skalierbarkeit betrachtet. Bei der „Velocity“ liegt der Fokus auf der Geschwindigkeit in der die Daten erzeugt werden. Dies hat wiederum je nach Geschwindigkeit der Datenerzeugung einfluss auf das Volumen. Abschließend in dem Dreieck gibt es die „Variety“. Big Data ist von natur aus eine Vielfalt an strukturierten und unstrukturierten Daten. All diese drei Eigenschaften beeinflussen sich gegenseitig und bilden die grundlegenden Eigenschaften von Big Data.

In Rohform sind diese Daten, wie eingangs erwähnt, jedoch nicht sonderlich von Wert. Egal ob Big Data oder nicht, einen Mehrwert und Informationen liefern sie erst, sobald man sie nutzt. Dabei ist es egal ob für Simulationen, Monitoring oder KI. In der Vergangenheit sind Daten ein Nebenprodukt der Digitalisierung gewesen. Heute sind sie ein eigenes Geschäftsfeld und „Enabler“ für viele bisher nicht möglich gewesen Anwendungen.[2][9]

2.1.2 Datenqualität

Im Zusammenhang mit Daten fällt immer häufiger auch der Begriff Datenqualität. Hier trifft Quantität auf Qualität. Wie bereits erwähnt, ist die Menge an Daten die bereits zur Verfügung steht, riesig. Quantität ist daher nicht das Problem. Die Qualität der Daten hat hier jedoch sehr großen Einfluss. Nicht selten können Daten nicht verwendet werden, da die Qualität nicht ausreichend ist. Gerade für Analysen, Auswertungen und Vorhersagen, wie sie durch die KI getroffen werden sollen, wird hohe Datenqualität benötigt.[10] Datenqualität selbst lässt sich auf unterschiedliche Arten und Weisen verstehen.[8] Eine gängige Definition für Datenqualität in der Literatur ist: „fitness for use“.[7] Es bedeutet, dass die Datenqualität von Nutzungskontext und Anwendungsfall abhängt und in erster Linie nicht allgemeine Qualitätsanforderungen erfüllt werden, sondern die für den Use Case benötigten Qualitätsanforderungen.[7][8]

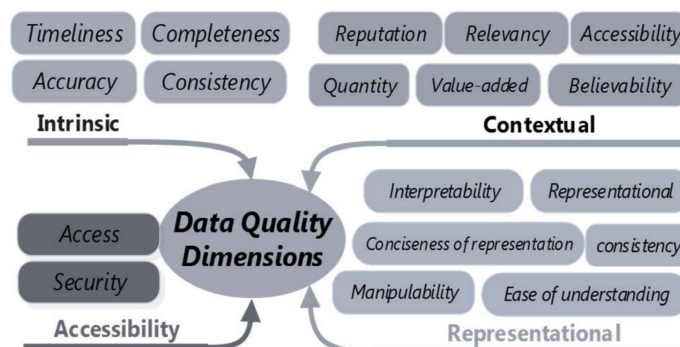


Abbildung 2.3: Data Quality Dimensions [4]

Für die Validierung der Datenqualität gibt es die sogenannten Data Quality Dimensions. In Abbildung 2.3 sind die vier unterschiedlichen Dimensionen: Intrinsic (deutsch (dt.): Intrinsisch), Contextual (dt.: Kontextbezogen), Accessibility (dt.: Zugänglichkeit) und Representational (dt.: Repräsentativ) dargestellt. Jede Dimension besitzt eigene Merkmale an denen die Datenqualität gemessen werden kann. Näher erläutert werden die Dimensionen in der Tabelle 2.1.

Intrinsic	Zur intrinsischen Datenqualität gehören, wie in Abbildung 2.3 zu sehen, die Merkmale: Zeitlos, Vollständigkeit, Genauigkeit, Konsistenz. Die intrinsische Dimension bildet die Qualitätsmerkmale der Daten selbst ab. und gibt Aufschluss über Objektivität und Glaubwürdigkeit der Daten.[8]
Contextual	Bei der kontextbezogenen Datenqualität wird betrachtet in welchem Ausmaß die Daten für den Nutzenden einsetzbar sind. Darin werden Eigenschaften wie: Datenmenge (Quantität), Zugänglichkeit, Aktualität und Mehrwert der Daten betrachtet um die Nutzbarkeit zu messen.[2]
Representational	Repräsentativ bedeutet im Kontext der Datenqualität, dass die Daten Interpretierbarkeit und dabei primär einfach zu Verstehen sind, Konsistent sind und Manipulierbarkeit, also Veränderbar sind.[10]
Accessibility	Die vierte Dimension, die Zugänglichkeit, bezieht sich Insbesondere auf das Speichersystem der Daten. Diese Dimension wird an der möglichst einfachen Zugänglichkeit und der entgegenstehenden Sicherheit der Daten gemessen.[10]

Tabelle 2.1: Data Quality Dimensions Merkmale

Datenqualität lässt sich in unterschiedlichsten Dimensionen und anhand unterschiedlicher Qualitätsmerkmale messen. Die in Abbildung 2.3 dargestellten und in Tabelle 2.1 erläuterten Dimensionen sind dabei nur eine der gängigen Modelle aus der Literatur.

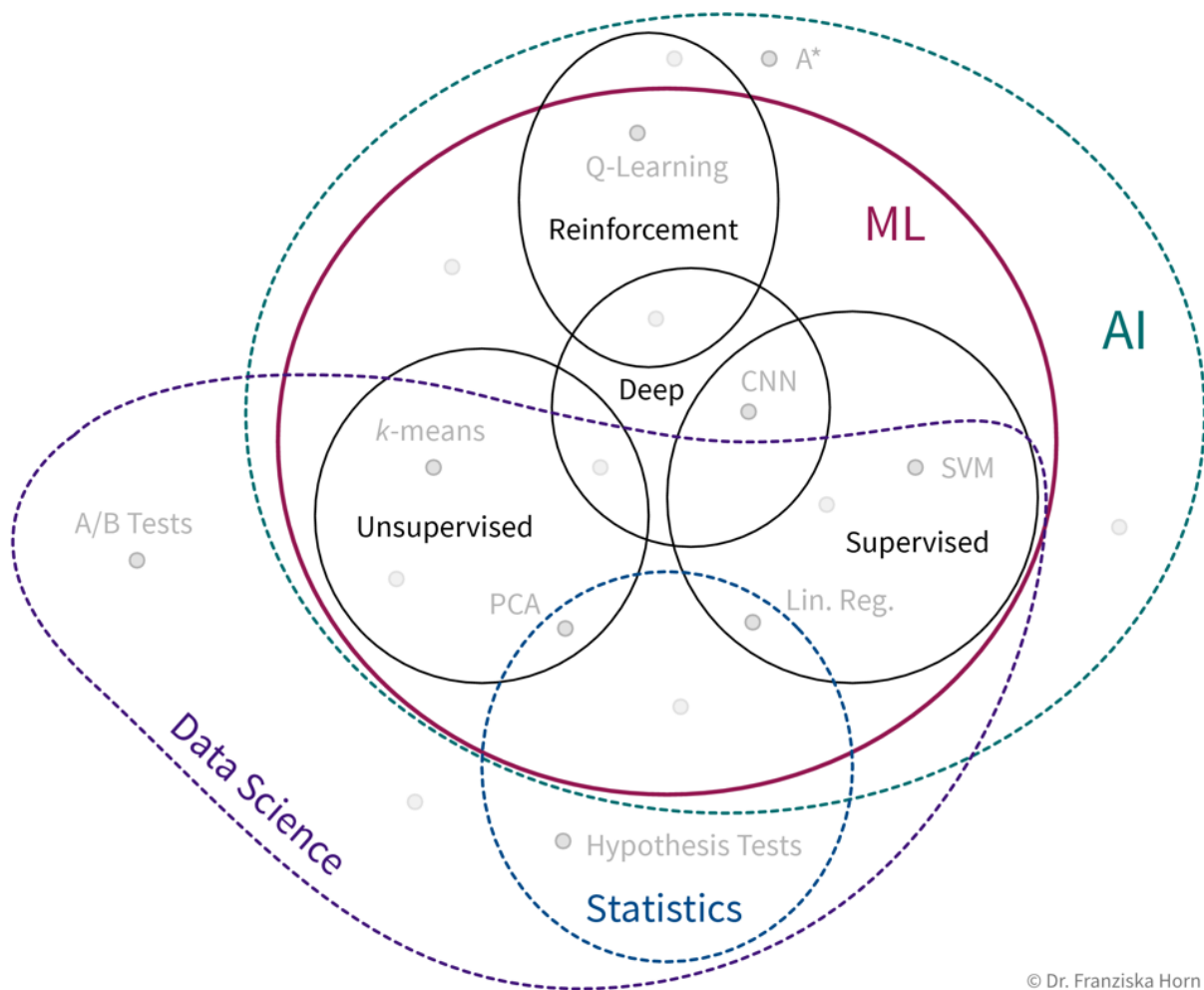
Trotz des Bewusstseins für Datenqualität, besteht rund 80 Prozent der Arbeit eines Data Scientist daraus, Daten aufgrund von Qualitätsanforderungen vorzuverarbeiten.[6] Angefangen beim einfachen Umwandeln bis hin zu komplexeren Bereinigungen und Encodierungen der Daten. Daten werden in den meisten Fällen nicht die gewünschten Qualitätsanforderungen erfüllen können. Datenqualität ist keine verallgemeinerbare Formel, sondern immer abhängig vom Anwendungsfall und dem Kontext in dem die Daten genutzt werden sollen.

Dass Datenqualität eine wichtige Rolle spielt, unabhängig von dem Verwendungszweck, ist keine Frage. In der Wissenschaft sowie Wirtschaft wird sich immer intensiver mit der Thematik der Datenqualität auseinandergesetzt, da heutzutage die Quantität kam noch ein Problem darstellt, sondern viel mehr die Qualität der Daten.[11] Denn Quantität ist nicht gleich Qualität!

2.2 Künstliche Intelligenz

2.2.1 Künstliche Intelligenz Allgemein

KI als Begriff wird vielseitig und unterschiedlich verwendet. Die Vision von KI ist es, die intellektuellen und menschlichen Fähigkeiten als ein KI-System nachzubilden.[11] Letztendlich beschreibt KI ein Forschungsbereich aus der Informatik, der aus einer Vielzahl aus Technologien besteht.[12]



© Dr. Franziska Horn

Abbildung 2.4: Exemplarisch dargestellte Disziplinen von KI[6]

Die Abbildung 2.4 zieht exemplarisch, welche Position KI in der Informatik einnimmt. Zur KI gehören bekannte Teilgebiete wie das ML, Optimierungsalgorithmen, aber auch in der Robotik kommt KI zum Einsatz. Selbst wird die Disziplin aber auch von anderen übergeordneten Disziplinen genutzt.[12] Ein Beispiel hierfür ist Data Science. Die Abbildung 2.4 veranschaulicht, dass einige Aspekte aus der KI im Bereich Data Science aufgegriffen werden, jedoch deutlich mehr als nur KI hinter Data Science steckt.

In der Literatur wurde in der Vergangenheit oft zwischen „schwacher“ und „starker“ KI unterschieden. Eine schwache KI ist dabei für ein spezifisches Anwendungsproblem konzipiert. Der KI wird eine Aufgabe gegeben und diese versucht mittels Algorithmen und mathematischen Funktionen die Aufgabe zu lösen. Diese Art der KI versucht Aufgaben zu lösen und Ergebnisse zu liefern, wie sie durch menschliche Intelligenz entstehen können.[11] Ein Beispiel hierfür ist die Bilderkennung, bei der Entschieden wird, ob auf dem Bild eine Erdbeere zu sehen ist, oder nicht.

Als starke KI wird ein System verstanden, welches in der Lage ist ein breites Spektrum an Aufgaben zu erledigen.[13] Sie verfügt über weitere Methoden zur Datenverarbeitung. In dieser Form der KI wird mehr versucht die menschlichen Fähigkeiten und Intelligenz abzubilden. Es wird nicht nach eine spezifischen Vorgehensweise vorgegangen, sondern Aufgaben werden mit individuellen Lösungswegen erledigt.[11]

Mit dem Begriff KI wird demzufolge nur ein grundlegendes Prinzip von KI-Systemen definiert. Die Aufgabe dieser KI-Systeme ist es, Daten jeglicher Form zu interpretieren, Schlussfolgerungen daraus zu ziehen und eine Aussage über das zu erreichende Ziel zu liefern. Unabhängig davon, ob es sich um einen optimalen Lösungsweg, eine Vorhersage oder eine Schlussfolgerung handelt. [12]

2.2.2 Teilgebiet maschinelles Lernen

Maschinelles Lernen ist das Teilgebiet der KI, welches häufig als KI bezeichnet wird. Unter ML wird verstanden, dass ein ML Modell von Lernalgorithmen entwickelt wird, welches in der Lage ist, das Wissen aus Erfahrungen anzuwenden. Dabei soll dem Modell ein unbekannter Input geliefert, dieses verarbeitet die Eingabewerte mit dem Wissen welches es besitzt und liefert ein Ergebnis als Output zurück. Der Output ist dabei abhängig von der zu lösenden Aufgabe. Klassische Anwendungsfälle von ML sind daher Vorhersagen, Schlussfolgerungen, Optimierungen, Entscheidungen, Sprach- und Bilderkennung beziehungsweise (bzw.) Verarbeitung und weitere ähnliche Aufgaben.[14]

Um das Teilgebiet ML genauer zu betrachten, unterscheidet man in der Literatur zwischen „supervised learning“(dt.: überwachtes Lernen), „unsupervised learning“(dt.: unüberwachtes Lernen) und dem „reinforcement learning“(dt.: bestärktes Lernen).[13] Diese Unterscheidung wird anhand der Art des Lernens getroffen.[6] In Abbildung 2.5, werden diese unterschiedlichen Ansätze des Lernens vereinfacht dargestellt.

Das „unsupervised learning“(dt.: unüberwachtes Lernen), in der Abbildung 2.5 links dargestellt, basiert auf dem Prinzip, dass dem Modell ein Input ohne zusätzliche Informationen gegeben wird und das Modell selbstständig Zusammenhänge erlernt. Lediglich die zu suchende Struktur ist dem Modell zu liefern, damit ist gemeint, ob nach bspw. Ausreißern

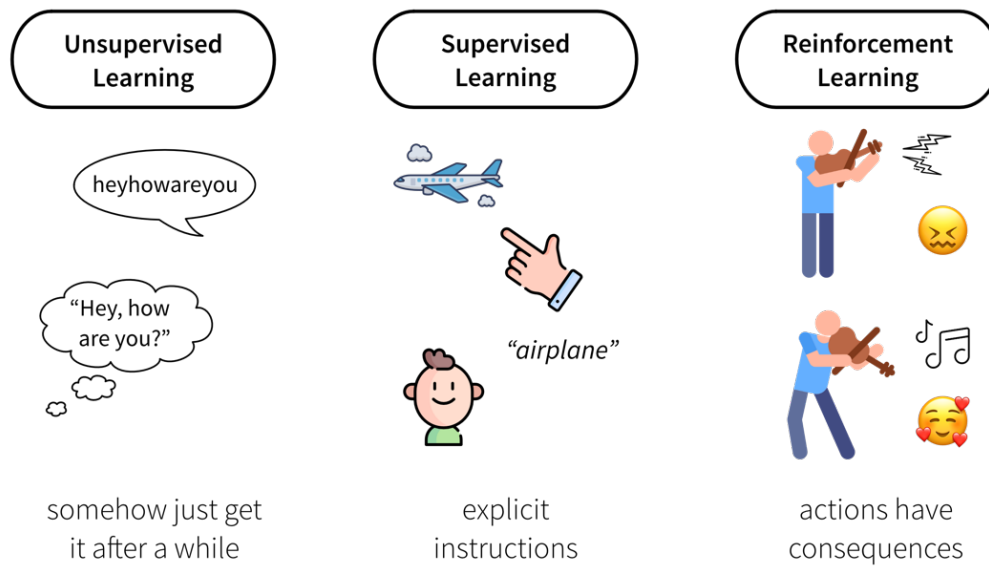


Abbildung 2.5: Arten des maschinellen Lernen[6]

gesucht werden soll. In der Abbildung 2.5, wird dies an dem Beispiel eines Texts veranschaulicht. Dem Modell wird lediglich die Buchstabenkette gegeben und dieses erkennt darin die einzelnen Wörter und, dass es sich um eine Frage handelt. Ein häufiger Anwendungsfall für das unüberwachte Lernen ist das Clustering, da das Modell in den Daten Zusammenhänge erkennt, die ein Mensch nicht direkt sieht.[13][15]

Bei dem „supervised learning“ (dt.: überwachtes Lernen), wird ein sehr Menschen ähnliches Lernverfahren genutzt. Vereinfacht gesagt wird bei dieser Art des Lernens etwas vorgemacht/erklärt und das Modell merkt sich diesen Zusammenhang. Auf diese Art und Weise erlernt das Modell Strukturen und kann dies auf ihm unbekannte Daten anwenden. Das Beispiel aus Abbildung 2.5 veranschaulicht diesen Lernverfahren. Hier wird auf ein Flugzeug gezeigt und dem Kinde gesagt, dass es sich um ein Flugzeug handelt. Das Kind merkt sich diesen Zusammenhang und ist beim nächsten mal selbst in der Lage das Flugzeug zu erkennen und es so zu benennen. Die gängigen Anwendungsfelder des überwachten Lernens sind Vorhersagen und Entscheidungen.[15]

Die dritte und bisher am wenigsten verbreitete Art des Lernens ist das „reinforcement learning“ (dt.: bestärktes Lernen). Es handelt sich dabei um Feedback basiertes Lernen. Es bedeutet, dass das Modell eine Auswahl an Handlungen hat, aus denen das Modell frei entscheiden darf, welche ausgeführt wird. Die Entscheidungen werden erst im Nachgang bewertet und in Form einer meist numerischen Bewertung an das Modell zurück gegeben. So lernt das Modell, wenn es etwas gut gemacht hat oder auch falsche Handlungen ausgeführt hat. In Abbildung 2.5, wird dies anhand einer musizierenden Person dargestellt. Die Person spielt etwas auf ihrem Instrument und bekommt nach dem Spielen eines Tons das Feedback der Zuhörenden ob es sich gut anhört oder nicht. So lernt die Person wann

sich etwas gut anhört und verbessert durch das Feedback ihr können. Häufig findet das bestärkende Lernen deshalb Einsatz in der Robotik, in der ein Roboter anhand von Feedback Bewegungsabläufe erlernt. Bekannt ist hier eine Roboter, der eine menschliche Hand nachbildet und erlernt einhändig einen Rubik's Cube zu lösen.[6][15][16]

Im Rahmen dieser Arbeit wird der Fokus speziell auf das supervised Learning gelegt. Denn überwachtes Lernen kommt meist dann zum Einsatz, wenn Vorhersagen oder Entscheidungen getroffen werden. Dieser Aspekt wird im Verlauf der Arbeit immer wieder eine Rolle spielen. Einzigartig am überwachten Lernen ist, wie bereits erwähnt, dass zum Lernen von Zusammenhängen immer ein „Lehrer“ benötigt wird. Im Falle eines ML Modells handelt es sich dabei nicht um eine Lehrer, sondern um Daten, genauer gesagt Trainingsdaten. Das Prinzip nach dem das überwachte Lernen funktioniert, ist in Abbildung 2.6 anhand eines Beispiels dargestellt.

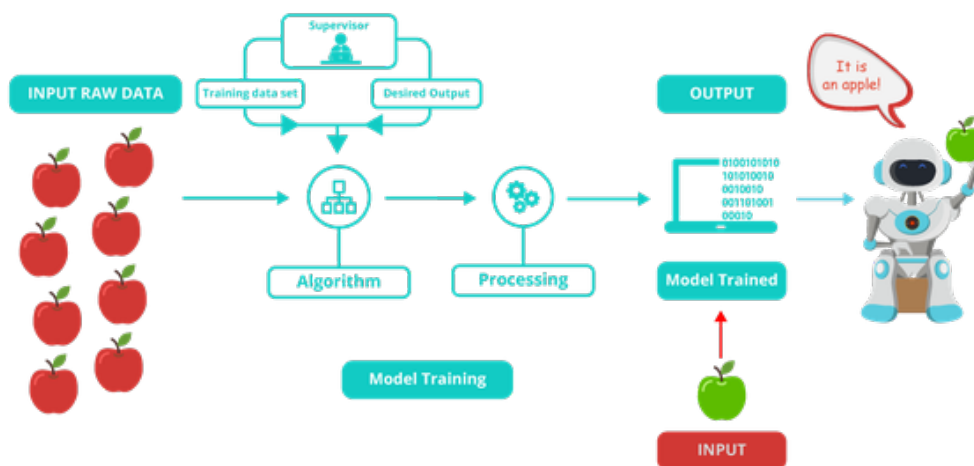


Abbildung 2.6: Der Prozess von überwachtem Lernen [17]

Zu Beginn wird das zu lösende Problem definiert, dabei geht es darum die Lernaufgabe des Modells festzulegen. Beim überwachten Lernen wird zwischen Regressionen und Klassifikation unterschieden. Regressionen kommen meist bei Schlussfolgerungen oder Vorhersagen zum Einsatz und Klassifikationen bei Entscheidungen. Anhand des Problems wird dann ein Algorithmus als Grundlage für das Modell ausgewählt. Um den Modell nun Wissen beizubringen werden sogenannte Trainingsdaten verwendet. Dabei handelt es sich um Daten in dem Format, wie sie auch später im Betrieb als Eingaben X geliefert werden. Diese werden jedoch davor mit Labels versehen. Diese Labels werden durch Experten für die Trainingsdaten erstellt. Im nächsten Schritt wird dann das Modell auf Basis der Trainingsdaten trainiert. Diese Phase ist die entscheidende Lernphase des Modells. Im Prinzip ist das Modell nach dem Training bereit für den Einsatz. Meist wird jedoch noch mit einem, dem Modell bisher unbekannten und nicht gelabelten, Datensatz überprüft

ob das Modell richtig funktioniert und die Qualität anhand unterschiedlicher Metirken, wie der Genauigkeit bewertet. Danach ist das Modell einatzbereit und kann mit Eingaben X gespeist werden und schließt dann auf Basis des erlernten Wissens auf eine Ausgabe Y. Auf diese Art und weiße werden beim überwachten Lernen die input Output Probleme gelöst.

Es gibt aber auch einige negative Eigenschaften bei ML Modelle zu berücksichtigen sind und zu möglichen Problemen führen können. Die meisten dieser Risiken lassen sich jedoch inzwischen durch Gegenmaßnahmen verhinder. Zu den Häufigsten Problemen gehören zu geringe Datenmengen, die Passgenauigkeit, Scheinkorrelationen und Verzerrungen.

1. Die Datenmenge ist ein entscheidender Faktor im überwachten Lernen. Viele Modelle scheitern daran, dass sie auf Basis von zu wenigen Trainingsdaten trainiert wurden und so nicht in der Lage sind treffende Schlüsse zu ziehen. [13][18] In Abbildung 2.7

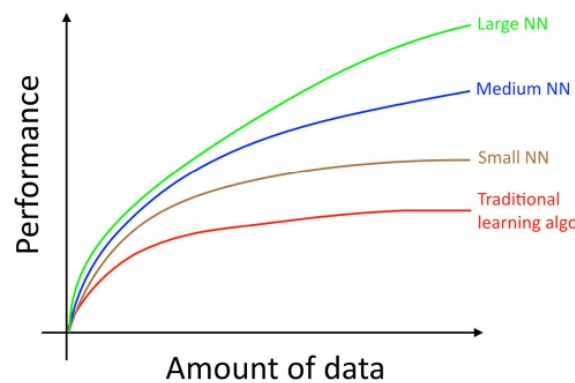


Abbildung 2.7: Veränderung der Performance durch die Datenmengen [18]

ist der Zusammenhang zwischen der Datenmenge und der Performance des Modells dargestellt. Hier wird zwar des spezielle Fall von unterschiedlich großen Neuronalen Netzen betrachtet, jedoch lässt sich dies auch auf allgemein ML übertragen. Der Verlauf ähnelt dem eines beschränkten Wachstums. Das bedeutet für das Modell, dass es zu Beginn mit einer ausreichenden Datenmenge möglich ist eine gute Performance zu erzielen, jedoch um später wenige Prozente an Verbesserung zu erzielen werden sehr große Datenmengen benötigt.[18] In einigen Szenarien existieren aber von Beginn an zu wenig Daten für eine gute Performance. Als Gegenmaßnahme können lediglich mehr Daten generiert und gesammelt werden.

2. Bei der Passgenauigkeit wird ein Problem bei dem Modell selbst adressiert, nämlich das overfitting (dt.: „überangepasst“) und underfitting (dt.: „unterangepasst“). Vereinfacht gesagt, wird beim overfitting der Trainingsdatensatz durch das Modell zu genau repräsentiert. Erkennbar wird overfitting, wenn man die Performance von dem Modell bei einem Trainings- und eine Testdatensatz betrachtet. Ist die Performance

bei den Trainingsdaten akzeptabel und bei den Testdaten deutlich schlechter, ist es nahezu sicher, dass das Modell zu sehr auf die Trainingsdaten angepasst ist. Beim underfitting hingegen, ist die Performance bei beiden Datensätzen nicht akzeptabel. Das liegt daran, dass beim underfitting zu wenig Zusammenhänge erlernt wurden und so keine realistischen Ergebnisse reproduziert werden können. Diese Problema-

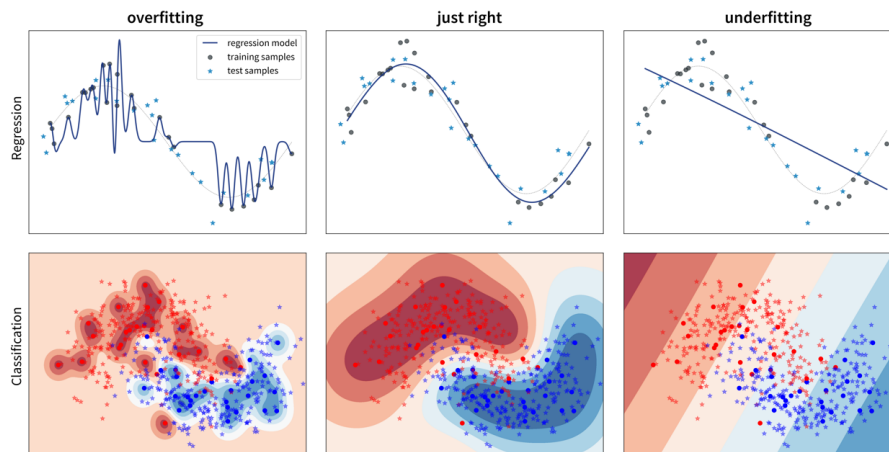


Abbildung 2.8: Overfitting und Underfitting [6]

tik wird in Abbildung 2.8 veranschaulicht. Auf der linken Seite ist ein überangepasstes Modell, welches den Trainingsdatensatz „auswendig gelernt“ hat und nicht auf neue Datenpunkte verallgemeinern kann. Rechts ist das unterangepasste Modell zu sehen, welches den Zusammenhang von Input und Output nicht abbilden kann und zu ungenau ist, da es „zu wenig gelernt“ hat.[6]

3. Nicht nur overfitting und underfitting sorgen in Modellen für Ungenauigkeiten auch Scheinkorrelationen sind häufig eine Fehlerursache. Bei Scheinkorrelationen, werden von dem Modell Zusammenhänge erlernt, die keinerlei Zusammenhang mit der tatsächlichen Entscheidung haben.

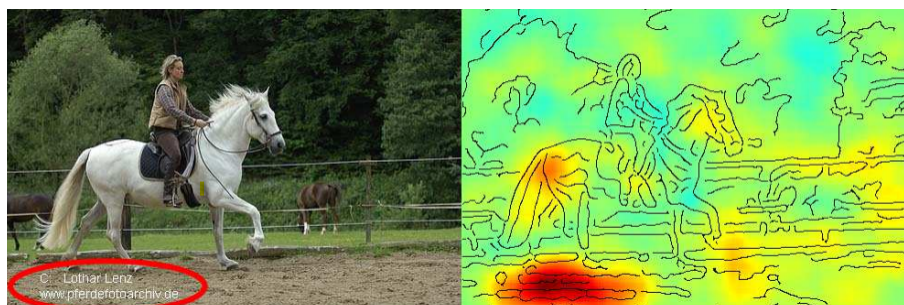


Abbildung 2.9: Scheinkorrelationen in der Bilderkennung [19]

Ein Beispiel dafür ist in Abbildung 2.9 zu sehen. In der ein Pferdebild nicht anhand der Kontur des Pferdes, sondern anhand des Wasserzeichens in dem Bild erkannt

wurde. Grund dafür war, dass einige der Trainingsdaten dieses Wasserzeichen beinhaltet haben und das Modell daraus eine Scheinkorrelation erlernt hat.

4. Ein letztes und schwerwiegendes Problem ist die Verzerrung von Trainingsdaten, auch Bias genannt. Dieser Aspekt wird später in Kapitel 2.3.1 genauer betrachtet. Allgemein kann man sagen, existieren Verzerrungen in Trainingsdaten, so wird das Modell diese Verzerrungen reproduzieren.[20]

Zusammenfassend wird bei ML, speziell beim überwachten Lernen auf Basis von Daten, Trainingsdaten genannt, Wissen aus der Vergangenheit erlernt. Dazu werden Zusammenhänge in den Daten genutzt und die dazugehörigen Bewertungen in Form der Labels. Paper die im hintergrund stehen[6][15][18][21]

2.2.3 Ethik in der künstlichen Intelligenz

Ein Aspekt der bisher nicht beleuchtet wurde ist die Ethik. Die Ethik spielt in der KI eine immer größer werdende Rolle. In einer Gesellschaft die für Inklusion steht, werden häufig Entscheidungen hinterfragt und kritisiert. Dies gilt insbesondere für Entscheidungen und Vorhersagen, die durch KI getroffen werden und so ohne direkten menschlichen Einfluss getroffen werden.

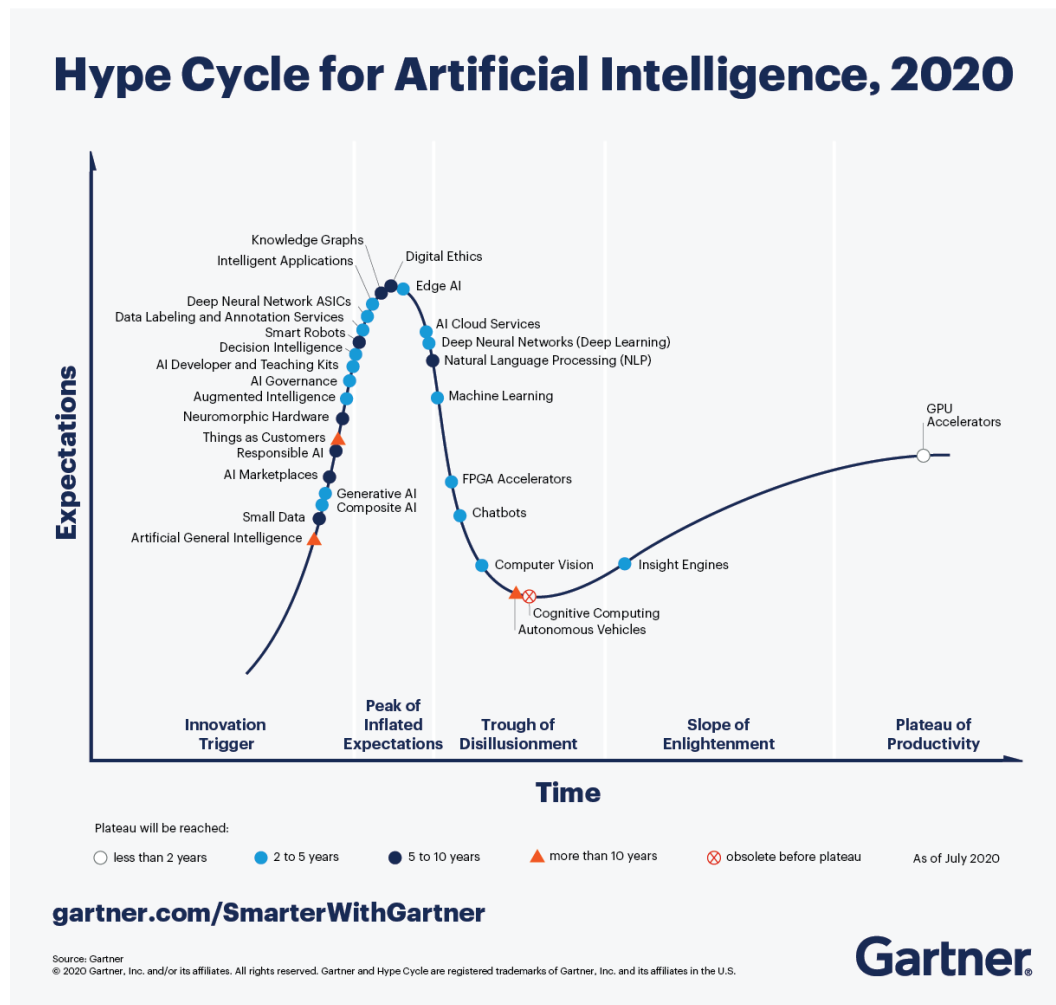


Abbildung 2.10: Gartner Hype Cycle für KI [22]

Im Gartner Hype Cycle für KI aus dem Jahr 2020, dargestellt in Abbildung 2.10, wurde das Thema Digitale Ethik auf dem Höhepunkt der Kurve platziert. Der Forschungsbereich rund um die digitale Ethik befindet sich jedoch erst im Anfangsstadium. Trotzdem sind die Erwartungen bereits sehr hoch, denn auch der gesellschaftliche Druck steigt an. Ursache dafür ist unter anderem der Skandal um den Chatbot Tay von Microsoft der innerhalb von nur einem einzigen Tag rassistische Äußerungen von den Nutzern lernte und daraufhin offline genommen werden musste.[20] Um einen einheitlichen Rahmen für die Ethik in der

KI zu schaffen wurde 2018 von der EU eine Kommission von hochrangigen KI Experten (kurz. HEG KI) gegründet, aber auch Forschungsinstitute setzten sich vermehrt mit der Thematik auseinander. 2021 waren es allein in Deutschland über 40 Projekte.[23]

Die Ethik wird dabei meist mit dem Konzept von vertrauenswürdiger KI in Verbindung gebracht. In diesem Konzept werden die folgenden drei Anforderungen an eine KI gestellt:

1. **Rechtmäßigkeit:**

Eine KI soll das geltenden Recht und die gesetzlichen Bestimmungen einhalten.

2. **Ethik:**

Eine KI soll ethische Grundsätze und Werte einhalten.

3. **Robustheit:**

Eine KI soll technisch Robust gegen Manipulation gestaltet sein.

Erfüllt eine KI alle drei Anforderungen gilt sie als eine vertrauenswürdige KI. Ziel des Konzepts ist es, die Vorteile von KI zu maximieren aber gleichzeitig die Risiken, egal ob rechtliche oder ethische, möglichst gering zu halten.[14] Wenn man sich nun mit dem Konzept einer vertrauenswürdigen KI genauer beschäftigt, müssen Fragestellungen, wie sie in Abbildung 2.11 zu sehen sind, gestellt werden.



Abbildung 2.11: Ethische und rechtliche Anforderungen an KI [20]

Hier kommt jetzt mehr Detaill [23] [24]

2.3 Bias im Zusammenhang mit KI

2.3.1 Bias

- ML ist immer auf Basis von Daten aus der Vergangenheit - Begriffserklärung: Data Bias vs Bias Verzerrung
 - Arten von Bias:
 - Bias durch Abwesenheit - Wenn eine Info fehlt, kann das zu Diskriminierung führen.
 - Diskriminierung durch Menschen.
- Arten von Bias: Cognitive, Social, Perceptual und Motivational Bias [12][25]

2.3.2 Diskriminierung durch Bias in Daten

- Bias in Daten - Wie funktioniert Bias in Daten - Was ist die Problematik - Welche Auswirkungen hat es - Realbeispiele <- Bewerbungsverfahren [26]

2.3.3 Gegenmaßnahmen

- Was kann man dagegen tun? - Gibt es möglichkeiten Trainingsdaten künstlich zu erzeugen und so eine neutrale betrachtung zu schaffen - Parameter entfernen als Lösung -> Die KI wird aber dadurch schlechter -

Wenn der Parameter mit dem Bias entfernt wird, wird das Ergebnis erstmal schlechter.

3 | Praktischer Teil

In diesem Teil der Arbeit werden zuerst die beiden Szenarien erläutert und daraufhin die Konzeption und Umsetzung derer in Python beschrieben. Zusätzlich wird die Datenauswertung in Tableaux dargestellt und zum Schluss das Ergebnis der Umsetzung evaluiert.

3.1 Szenarien

Für das generieren von Daten wurden zwei möglichst reale Szenarien ausgewählt. Zum einen das Szenario eines Bewährungsantrages, für welches 5 verschiedene Attribute und eine endgültige Bewertung mit stattgegeben oder nicht generiert werden. Zum anderen das zweite Szenario des social creditpoint system, für welches pro Person 7 Attribute zu generieren sind und eine numerische Bewertung zwischen 600 und 1400 creditpoints erstellt wird. Diese beiden Szenarien werden im folgenden genauer erläutert.

3.1.1 Szenario 1

In Szenario 1 soll ein Bewährungsantrag einer Person Bewertet werden. Ein Antrag besteht dabei aus dem Namen der Person, dessen Geschlecht, Hautfarbe und den entscheidenden Attributen der laufenden Strafe in Jahren und der Härte des Vergehens. Basierend auf diesen Attributen soll ein Bewerter beurteilen, ob der Antrag genehmigt oder abgelehnt wird. Das Geschlecht wird in „Männlich“ und „Weiblich“ angegeben. Da zur Vereinfachung sich auf das biologische Geschlecht begrenzt wurde und aus diesem Grund die Genderdiversität für den Datengenerator außenvor gelassen wurde. Die Hautfarbe der Person wird als „Schwarz“ oder „Weiß“ festgehalten. Die noch laufende Strafe des Gefangenen wird als Ganzzahl in Jahren von 1-5 angegeben. Da in diesem Fall ein Bewährungsantrag erst ab maximal 5 Jahren noch offene Strafe gestellt werden. Die Härte des Vergehens wird einfachheitshalber in den Gruppen „Leicht“, „Mittel“ oder „Hart“ festgehalten.

Für die Beurteilung des Antrags von der bewertenden Person werden folgende Regeln definiert:

Das Geschlecht und die Hautfarbe werden hierbei nicht direkt aufgelistet, da diese in der Regel keine Auswirkung auf die Bewertung haben sollten. Diese können jedoch durch einen konkreten Bias Aussagekraft bekommen. Damit soll in den generierten Daten die gewünschte Verzerrung auf einen gewissen Wert gelegt werden können. In diesem Szenario

Attribut	Positive Auswirkung	Negative Auswirkung
Laufende Strafe	1-3	4-5
Härte des Vergehens	Leicht, Mittel	Hart

Tabelle 3.1: Tabelle für die Auswirkung der Attributen von Szenario 1

sind die möglichen Werte, welche durch eine Verzerrung und damit einem menschlichem Vorurteil einer bewertenden Person beeinflusst werden können, das Geschlecht und die Hautfarbe. Die anderen beiden Attribute, welche in der Tabelle 3.1 aufgeführt sind, wirken sich durch ihre Ausprägungen positiv oder negativ auf die Bewertung des Antrages aus. So wirkt z.B. eine Härte des Vergehens vom Niveau Leicht sich eher für eine positive Bewertung des Antrages aus, als eine mittlere Härte. Dasselbe gilt auch für die Laufende Strafe. So kann eine bewertende Person dann anhand dieser beiden Werte eine Tendenz erhalten und über die Gestattung des Antrages entscheiden.

3.1.2 Szenario 2

Im zweiten Szenario wird das durch China populär gewordene sozial creditpoint system in einer lagenunabhängigen Version nachgebaut. Dafür werden Einträge zu Personen erstellt, nach welchen die Punktzahl der einzelnen Person zwischen 600 und 1400 Punkten bestimmt wird. Ein Eintrag zu einer Person beinhaltet die sieben in der folgenden Tabelle dargestellten Attribute mit den unterschiedlichen Ausprägungen. Die in Tabelle 3.2

Attribut	Ausprägungen
Name	Beliebig
Alter	20-79
Politische Orientierung	Links, Mitte, Rechts
Bildungsabschluss	Ausbildung, Fachschulabschluss, Bachelor, Master, Diplom, Promotion, ohne
Soziales	0-3
Wohnlage	Großstadt, Kleinstadt, Vorort, Ländlich
CO2-Fußabdruck	4-12

Tabelle 3.2: Tabelle der Attribute und Auswirkungen von Szenario 2

aufgeführten Ausprägungen haben ähnlich wie zu Szenario 1 unterschiedlich starke Auswirkungen auf den am Ende bestimmten social Score. Einzig allein der Name und das Alter sollen keine direkte Auswirkung auf den social Score haben. Die anderen Attribute wirken sich je nach Auswirkung positiv durch eine Erhöhung des Scores oder negativ durch eine Verringerung des Scores aus. Insgesamt werden so in diesem Szenario viele

Einträge von Personen erstellt, welche alle unterschiedlichste Verteilungen der Ausprägungen besitzen und dadurch in der Bewertung einen individuellen social Score erzielen. Um nun eine gewünschte Verzerrung in die Daten zu bekommen, können alle Attribute bis auf den Namen, welcher rein als Füllwert dient, durch eine Verzerrung beeinflusst werden. So können z.B. Personen zwischen 20-30 Jahre negativ verzerrt werden, da ein oder zwei Bewertende etwas gegen junge Leute haben und diesen aus ihrer Überzeugung einen schlechteren Score geben. In diesem Szenario ist somit eine hohe Variabilität geboten inwieweit eine Verzerrung in die Daten gebracht wird. Zudem kann auch eine Verzerrung über mehrere Attribute eingebracht werden, da eine bewertende Person z.B. auch etwas gegen eine Rechte Politische Orientierung und ein schlechtes Soziales Engagement von 0 haben kann.

3.2 Konzeption

In diesem Kapitel wird die erarbeitete Konzeption für die Umsetzung der beiden im Kapitel 3.1 aufgeführten Szenarien erläutert. Dabei wird in ein Grobkonzept zur allgemeinen Generierung der Daten und daraufhin ein Feinkonzept für jedes Szenario unterteilt.

3.2.1 Grobkonzept

Das Grobkonzept beinhaltet die Überlegungen, wie die Programme/Notebooks für die beiden Szenarien generell aufgebaut sein sollen. Der Ablauf der Programme von der Eingabe der Parameter bis hin zu den fertig generierten Daten wird in fünf Schritten durchgeführt. Der Ablauf der Schritte ist in folgendem Programmablaufplan dargestellt.

Die in der Abbildung 3.1 dargestellten Hauptschritte des Programmablaufs lauten: Parametereingabe, Generieren der Daten, Regeln aufstellen, Bewerten und Speichern der Daten. Im ersten Schritt der Parametereingabe, wird den Benutzenden die Möglichkeit gegeben die Parameter für die Generierung der Daten einzugeben, wie z.B. die Anzahl der Daten oder Bewertende welche generiert werden sollen. Im folge Schritt wird die passende Anzahl an Daten für das jeweilige Szenario generiert. Dabei sollen die Daten möglichst an Verhältnissen aus der Realität angepasst und auf dieser Grundlage generiert werden. Es soll jedoch eine gewisse Zufälligkeit in der Generierung vorhanden sein, sodass bei mehrfach Generierung unterschiedliche Datensätze auf Basis der definierten Verteilungen entstehen. Nach Abschluss der Generierung wird der fertige Datensatz zwischengespeichert, um diesen später bewerten zu können. Für die Bewertung des Datensatzes muss

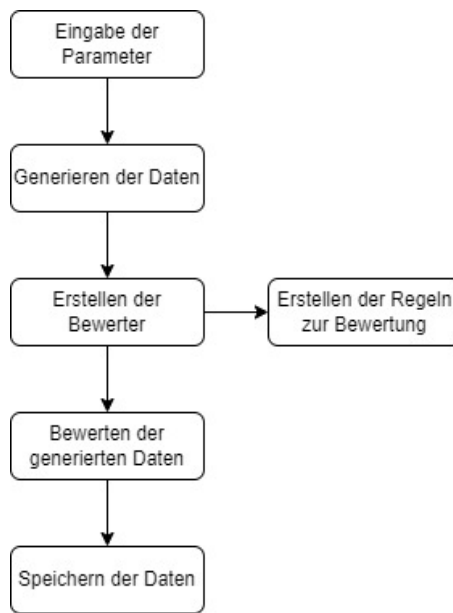


Abbildung 3.1: Programmablaufplan der fünf Hauptschritte zur Generierung der Daten

im folgenden 3. Schritt die Regeln nach welchen bewertet wird aufgestellt werden. Dafür soll zuerst die in den Parametern gewünschte Anzahl an Bewertenden erstellt werden, da durch diese die Regeln erstellt werden. Unter den erstellten Bewertenden müssen zudem noch die angegebene Anzahl an diskriminierenden Bewertenden in solche umgewandelt werden. Daraufhin können dann die Regeln für die Bewertenden erzeugt werden. Somit ist der 3. Schritt abgeschlossen und alle Vorbereitungen getroffen für die Bewertung. In der Bewertung bekommen die Bewertenden alle Anträge des Datensatzes vorgelegt, welche Sie basierend auf den Regeln bewerten. Die Bewertung des jeweiligen Antrages wird diesem in den Daten hinzugefügt. Damit kann zum letzten Prozess übergegangen werden. In diesem wird der Ursprungsdatensatz und der bewertete Datensatz abgespeichert und damit auch außerhalb von dem Programm zugänglich gemacht.

Somit ist das Grobkonzept der Szenarien abgeschlossen und ein Grundgerüst konnte entworfen werden. Im weiteren kann nun auf die detaillierte Feinkonzeption der einzelnen Szenarien eingegangen werden.

3.2.2 Feinkonzept

Die im Grobkonzept beschriebenen fünf Hauptschritte der Programme sind für beide Szenarien gleich. Jedoch unterscheiden sich die Schritte im Detail bei beiden Szenarien. Daher wird für jedes Szenario ein eigenes Feinkonzept zur Füllung des selben vorhandenen Grundgerüst entwickelt.

Parametereingabe

Im ersten Schritt der Parametereingabe unterscheiden sich die Szenarien nicht, da beide einen Parameter für die gewünschte Diskriminierung, die Anzahl der Daten, die Anzahl der Bewertenden, die Anzahl der diskriminierenden Bewertenden und der Stärke der Auswirkung von der Diskriminierung benötigen. Diese Parameter können von den Benutzenden in beiden Fällen in einer finalen Zelle editiert werden.

Daten generieren

In diesem Prozess unterscheiden sich beide Szenarien stark, da zum einen für das erste Szenario nur fünf anstelle von sieben Attributen bei Szenario 2 generiert werden müssen und zum anderen existieren deutlich weniger Verbindungen zwischen den Attributen in Szenario 1. Bei dem ersten Szenario basiert nur die Härte der Strafe und die Hautfarbe auf dem Geschlecht, dies bedeutet die Wahrscheinlichkeiten für diese Attribute sollen dem Geschlecht entsprechend angepasst werden. So haben zum Beispiel weibliche Personen eher seltener eine harte Strafe als männliche Personen. Alle anderen Attribute in diesem Szenario haben eine feste Wahrscheinlichkeitsverteilung. Damit sind die Zusammenhänge in diesem Szenario sehr klein gehalten und überschaubar.

In Szenario 2 müssen sieben Attribute generiert werden und es sollen deutlich mehr Verbindungen zwischen diesen existieren. Um diese verständlich darzustellen wurde ein Diagramm für das Feinkonzept entworfen, welches nachfolgend dargestellt ist.

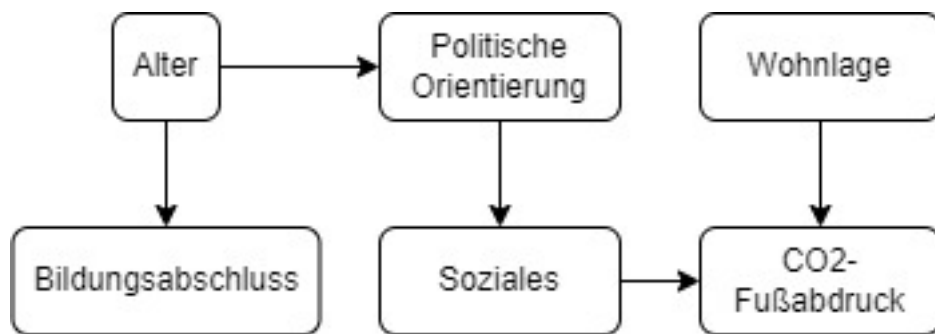


Abbildung 3.2: Verbindungen zwischen den Attributen des zweiten Szenario

In der Abbildung 3.2 sind alle Attribute, bis auf das Attribut Name, des zweiten Szenario als abgerundete Rechtecke und die Verbindungen zwischen diesen mit Pfeilen dargestellt. Insgesamt existieren fünf Verbindungen zwischen den Attributen. Diese sollen so aufgebaut werden, um möglichst Realitätsnahe Daten generieren zu können. Die Verbindung zwischen dem Alter und dem Bildungsabschluss existiert, da zum Beispiel die Wahrscheinlichkeit, dass eine Person mit 20 schon eine Promotion besitzt nicht so hoch ist wie bei einer Person im Alter von 50. Zudem hat das Alter einen Einfluss auf die Politische Orientierung einer Person, da junge Leute sicherlich andere Orientierungen haben als Personen im Alter von 50 zum Beispiel, siehe Aktionen wie „FridaysforFuture“. Durch die Politische Orientierung einer Person wird in diesem Fall auch mit einer Beeinflussung auf das soziale

Verhalten gerechnet und es besteht daher hier auch eine Verbindung. Im Falle des CO₂-Fußabdruck wird in dieser Arbeit mit einer Auswirkung der Wohnlage und des Sozialen gerechnet. Durch das Soziale und die Vorverbindungen wird die Politische Orientierung und das Alter ebenfalls indirekt darauf mit einbezogen. Dies ist hier der Fall, da die Berechnung des CO₂-Fußabdruck sich aus vielen Faktoren zusammensetzt und daher auch dieser in diesem Szenario durch viel beeinflusst werden soll. Durch diese Zusammenhänge sollen dann möglichst realitätsnahe Daten entstehen.

Insgesamt müssen für beide Szenarien später bei der Umsetzung die Wahrscheinlichkeiten der Ausprägungen der Attribute, wo es möglich ist, durch Statistiken bestimmt und die Verbindungen dadurch ebenfalls bestätigt werden. Dies wird im Kapitel 3.3 im Detail erläutert.

Regeln aufstellen

Im diesem Schritt werden zuerst bei beiden Szenarien die gewünschte Anzahl an Bewertenden erstellt, von welchen danach die Anzahl an diskriminierenden ausgewählt wird. In der Folge können die Regeln für die zuvor als Objekte erstellte Bewertenden erstellt werden. Im Fall des ersten Szenarios werden zwei Listen mit Hilfe der unter Kapitel 3.1.1 gezeigten Tabelle erstellt. Eine Liste beinhaltet die in der Tabelle dargestellten positiven Auswirkungen und die andere Liste beinhaltet die negativen Auswirkungen. So haben die Objekte der Bewertenden ihre eigenen Listen an Regeln. Daher kann für die diskriminierenden Bewertenden in der Liste der negativen Auswirkungen das zu diskriminierende Attribut hinzugefügt werden. So diskriminieren diese Bewertenden automatisch, da sie diese Regeln zur Überprüfung der Bewertung haben. Für das zweite Szenario sieht das Konzept hier etwas anders aus, da es hier nicht um eine Bewertung in genehmigt oder nicht geht, sondern um Punkte. Somit müssen die Regeln so erstellt werden, dass die Bewertenden eine Liste an allen Ausprägungen der Attributen haben und dazu eine passende Zuordnung mit wie vielen Punkten sich welche Ausprägung auf den Score auswirkt. Die Verzerrung wird in diesem Fall erst im nächsten Schritt der Bewertung betrachtet.

Bewertung

Folgend auf die erstellten Regeln können die Bewertenden nun die vorgelegten Einträge der Daten bewerten. Im ersten Szenario wird das ganze durch eine Wahrscheinlichkeitsverteilung durchgeführt. Zu Beginn jeder Bewertung steht es 50:50 für genehmigt oder nicht. Durch die erstellten Regel-Listen können dann die Bewertenden die im Antrag aufgeführten Attribute abgleichen, ob diese sich positiv (also für eine Genehmigung) oder negativ auswirken. Nach dieser Bestimmung wird dann die Wahrscheinlichkeitsverteilung verschoben in positive oder negative Richtung. So kann am Ende wenn der Bewertende alle Attribute durch hat, mit Hilfe der übrig gebliebenen Wahrscheinlichkeiten für positiv und negativ eine Entscheidung getroffen werden. Falls ein bewertendes Objekt diskriminierend sein sollte, hat dieses wie oben erläutert in seinen negativen Regeln die gewünschte Aus-

prägung enthalten, sodass diese sich dann auf die Entscheidung auswirkt. Für das zweite Szenario wird nicht mit einer Wahrscheinlichkeitsverteilung gearbeitet, sondern mit dem mittleren Wert des Scores als Startwert(1000). So können die Bewertenden von Attribut zu Attribut aus dem zu bewertenden Eintrag durchlaufen und entsprechend nach der Ausprägung den in Ihren eigenen Regeln definierte Wert dem Startwert hinzu addieren. Damit entsteht dann letztendlich der finale Score für den Eintrag. Für die Verzerrung wird das Attribut und die Ausprägung dessen welche verzerrt werden soll in jedem Eintrag gesucht. Falls die gewünschte Ausprägung vorhanden ist, wird der Score dieses Eintrages um die in den Parametern eingegebene negative Auswirkung für die Verzerrung addiert. Insgesamt werden bei beiden Szenarien die Einträge aus dem generierten Datensatz zufällig einem Bewertenden zur Bewertung zugeordnet. So ist eine zusätzliche Variabilität in der Verteilungen der Verzerrung gegeben.

Speichern der Daten

Zum Abschluss werden bei beiden Szenarien gleich, die beiden Datensätze als CSV Datei gespeichert. Zum einen den ursprünglich generierten Datensatz und zum anderen auch der Datensatz mit der jeweiligen Bewertung enthalten. Durch eine CSV Datei können die Daten dann beliebig in anderen Programmen weiterverwendet werden.

Insgesamt ist damit die Konzeption abgeschlossen. Im Grobkonzept wurde ein Grundgerüst für die beiden Programme der Szenarien entworfen, welches auch für noch weitere Szenarien der Art verwendet werden kann. Im Feinkonzept wurde dann das Grundgerüst durch Inhalt der jeweiligen Szenarien gefüllt und das geplante im Detail beschrieben. So kann nun zur Umsetzung der beiden Programme als Notebooks übergegangen werden.

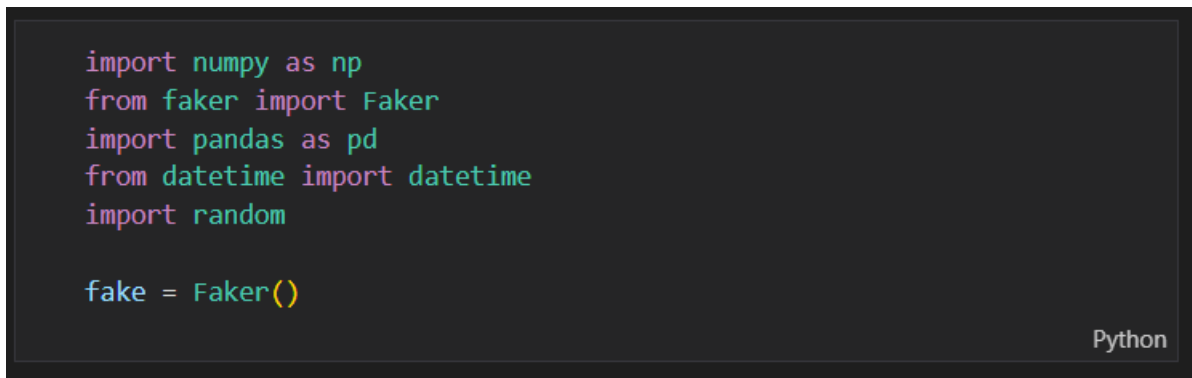
3.3 Umsetzung

In diesem Kapitel wird die konkrete Umsetzung der beiden Szenarien in einzelnen Unterkapiteln beschrieben.

3.3.1 Umsetzung des Szenario 1

Da die Programme als Notebooks in Python umgesetzt sind, können die einzelnen, in der Konzeption dargestellten, Prozessschritte als Zellen verwirklicht werden. Nun wird die Umsetzung des ersten Szenarios beschrieben.

Im Programm für das erste Szenario, welches „Szenario1.ipynb“ heißt, müssen in der ersten Zelle die benötigten Bibliotheken geladen werden.



```
import numpy as np
from faker import Faker
import pandas as pd
from datetime import datetime
import random

fake = Faker()
```

Python

Abbildung 3.3: Erste Zelle Code des Szenario1

In der Abbildung 3.3 ist die erste Zelle mit den Importen der Bibliotheken zu sehen. Die Bibliothek „numpy“ wird für Zufallsauswahlen unter bestimmten Wahrscheinlichkeiten benötigt. „faker“ ist eine Bibliothek für generierte Daten, so wird diese hier für das bestimmen zufälliger Namen verwendet. „pandas“ bietet sogenannte Dataframes in welchen die Daten gespeichert werden und durch „pandas“ auch in eine CSV Datei geschrieben werden können. Die letzte Bibliothek „random“ ist ebenfalls wie „numpy“ für das generieren von Zufallswerten zuständig. Zum Schluss wird in der Zelle noch eine Instanz der Faker Klasse erstellt, welches zur Verwendung der Bibliothek benötigt wird.

In der nächsten Zelle ist die Methode „create_fake_data“ zur Generierung der Daten umgesetzt. Diese Methode bekommt die beiden Parameter „num“ und „seed“ übergeben. Der Parameter „seed“ wird zu Beginn verwendet um den Startwert der Faker Instanz und von „numpy“ zu setzen. Dadurch wird es ermöglicht, mit unterschiedlichen Startwerten, eine nahezu „echte“ Zufallszahl zu generieren. Bei gleichbleibendem Startwert und gleicher Methode würde der Code immer die gleiche Zufallszahlen bestimmen. Zum Beispiel wenn drei Zahlen von 0-10 generiert werden sollen, werden bei gleichem Startwert immer die drei selben Zahlen generiert. Variiert der Startwert jedoch, werden jedes Mal unterschiedliche Zahlen generiert und eine ausreichende Variabilität erreicht. Als nächstes wird mit einer Schleife über die im Parameter „num“ angegebene Zahl iteriert. In jedem Schleifendurchlauf wird ein Eintrag für den Datensatz generiert. Somit ist „num“ die Größe des gewünschten Datensatzes. In diesem Szenario müssen somit in jedem Durchlauf ein Wert für die Attribute Name, Geschlecht, Härte der Strafe, Hautfarbe und Laufende Strafe ermittelt werden. Der Name in jedem Eintrag wird durch die Faker Instanz generiert. Dafür wird je nach Geschlecht die Methode zum generieren eines weiblichen oder männlichen Namen aufgerufen. Die Attribute Geschlecht, Härte der Strafe und Hautfarbe müssen nach bestimmten Wahrscheinlichkeiten berechnet werden. Dies wird für jedes dieser Attribute wie folgend ausgeführt:

Geschlecht

Formel zur Berechnung der Wahrscheinlichkeiten für Männlich (M) und Weiblich (W):

Gegeben: Gesamt = Gesamt Zahl der Gefangenen

$$W/M\% = \frac{W/M}{Gesamt} \quad (3.1)$$

Ausprägungen	Berechnung	Wahrscheinlichkeit
Weiblich	$\frac{105.000}{1.439.800}$	7,3%
Männlich	$\frac{1.334.800}{1.439.800}$	92,7%

Tabelle 3.3: Tabelle zur Bestimmung der Wahrscheinlichkeiten für das Geschlecht

Das Geschlecht für eine Person wird nach den in der Tabelle 3.3 berechneten Wahrscheinlichkeiten bestimmt. Die hier berechneten Wahrscheinlichkeiten ergeben sich aus den Gefängniszahlen von 2017 der USA, welche in einem Beitrag des U.S. Department of Justice im April 2019 veröffentlicht wurden. Die Zahlen wurden hierbei aus der „Table 8“ des Papers entnommen und zur Berechnung nach der oben aufgeführten Formel verwendet.[27, S. 17]

Die daraus entstehenden Wahrscheinlichkeiten werden wie in der folgenden Abbildung 3.4 zur Bestimmung des Geschlechts mit Hilfe der „numpy“ Bibliothek verwendet.

```
#Random choice of sex with specified probability
sex = np.random.choice(["M", "W"], p=[0.927, 0.073])
```

Abbildung 3.4: Codezeile zur Bestimmung des Geschlechts einer Person

In der Zeile Code werden zuerst die möglichen Ausprägungen als Strings in einem Array angegeben und dann die dazugehörigen Wahrscheinlichkeiten nach welchen eine Ausprägung bestimmt werden soll.

Härte der Strafe

Für die Bestimmung einer Ausprägung der Härte der Strafe, ist die Berechnung der Wahrscheinlichkeiten und die letztendliche Auswahl etwas komplexer. Da wie in der Konzeption geplant die Härte der Strafe vom Geschlecht einer Person abhängig ist. Daher ist es nicht möglich eine einfache Formel zur Berechnung aufzustellen, sondern es muss aus der Datenquelle erörtert werden wie sich die Wahrscheinlichkeiten verteilen. Die Verteilung der Wahrscheinlichkeiten ist in der folgenden Tabelle dargestellt.

Die in Tabelle 3.4 gezeigten Wahrscheinlichkeiten ergeben sich aus dem Beitrag des U.S. Department of Justice vom April 2019. In diesem ist in „Table 12“ eine prozentuale Verteilung von den Strafgruppen Gewalttätig, Eigentum, Drogen, Öffentliche Ordnung und

Ausprägungen	Weiblich	Männlich
Leicht	36,1%	26,6%
Mittel	26,4%	16,9%
Hart	37,5%	56,5%

Tabelle 3.4: Tabelle der Wahrscheinlichkeiten für die Härte der Strafe nach Geschlecht

Sonstige über die Geschlechter W und M aus dem Dezember 2016 aufgelistet. Dabei sind die Strafen nach den schwersten Delikten absteigend aufgeführt. Somit wird diese Verteilung auf die in der Konzeption definierten Ausprägungen Leicht, Mittel und Hart verteilt. So wird der Prozentsatz der gewalttätigen Strafen Hart zugeordnet, die Eigentumsstrafen Mittel zugeordnet und die Drogen, Öffentliche Ordnung und Sonstigen Strafen Leicht zugeordnet. Dadurch ergeben sich auf Grundlage der Zahlen aus den USA die Wahrscheinlichkeiten in der Tabelle 3.4.[27, S. 21]

Um die Bestimmung der Härte der Strafe nun durchzuführen, wird der selbe Code wie in Abbildung 3.4 gezeigt auf dieses Attribut angepasst und im Verbund mit einer if Abfrage zur Überprüfung des Geschlechts umgesetzt. So wird entsprechend dem Geschlecht einer Person nach den dazu passenden Wahrscheinlichkeiten die Härte der Strafe zufällig ausgewählt.

Hautfarbe

Formel zur Berechnung der Hautfarbe basierend auf dem Vorwissen des Geschlechtes:

Gegeben: Gesamt W/M = Anzahl an Weißen und Schwarzen Gefangenen pro (G)Geschlecht
 $= S(\text{Schwarz}) + Wi(\text{Weiß})$

$$S/Wi\% = \frac{S/Wi}{\text{GesamtW/M}} \quad (3.2)$$

Ausprägungen	Berechnung	Wahrscheinlichkeit
Schwarz	$S\% = \begin{cases} \frac{456.300}{843.700} & \text{ Geschlecht: M} \\ \frac{19.600}{68.700} & \text{ Geschlecht: W} \end{cases}$	$S\% = \begin{cases} 54,1\% & \text{ Geschlecht: M} \\ 28,5\% & \text{ Geschlecht: W} \end{cases}$
Weiß	$Wi\% = \begin{cases} \frac{387.400}{843.700} & \text{ Geschlecht: M} \\ \frac{49.100}{68.700} & \text{ Geschlecht: W} \end{cases}$	$Wi\% = \begin{cases} 45,9\% & \text{ Geschlecht: M} \\ 71,5\% & \text{ Geschlecht: W} \end{cases}$

Tabelle 3.5: Tabelle zur Bestimmung der Wahrscheinlichkeiten für die Hautfarbe unter Berücksichtigung des Geschlechts

Die in der Tabelle 3.5 dargestellten Berechnungen und daraus resultierenden Wahrscheinlichkeiten beruhen erneut auf der Veröffentlichung vom U.S. Department of Justice. In dieser sind in „Table 8“ die Gefangenen nach Geschlecht und ethnischen Gruppen aufge-

teilt. Zur Vereinfachung wurden für die Hautfarbe jedoch nur zwischen Schwarz und Weiß unterschieden. Dabei werden Ethnien bewusst nicht gesondert berücksichtigt. Somit sind lediglich die Zahlen für die Anzahl an der Gruppe Schwarz und Weiß nach Geschlecht Männlich Weiblich von Interesse. Um daraus Prozentwerte zu bilden, nach welchen dann eine Person entweder die Hautfarbe Schwarz oder Weiß bekommt, wurde die oben gezeigte Formel aufgestellt. Für die Formel wird zuerst zum einen die Gesamtheit an Schwarzen sowie Weißen männlichen Personen und zum anderen die Gesamtheit an Schwarzen sowie Weißen weiblichen Personen gebildet. Daraufhin können basierend auf diesen Gesamtwerten die Wahrscheinlichkeitsverteilungen für Männlich und Schwarz, Männlich und Weiß, Weiblich und Schwarz, Weiblich und Weiß anhand der Formel berechnet werden.

Damit diese Wahrscheinlichkeiten bei der Bestimmung angewendet werden können, wird wie schon bei der Härte der Strafe, der Code aus Abbildung 3.4 durch eine if Abfrage erweitert und an dieses Attribut angepasst.

Als letztes Attribut wird noch eine Ausprägung für die Länge der Strafe bestimmt. Hierfür wird mit der „numpy“ Bibliothek eine zufällige Ganzzahl zwischen eins und fünf ausgewählt.

Um einen Schleifendurchlauf abzuschließen werden die durch Wahrscheinlichkeiten bestimmten Werte zusammen als ein Dictionary als neuen Eintrag in ein Array mit der Zuordnung(Attribut:Ausprägung) hinzugefügt. Damit ist ein Schleifendurchlauf abgeschlossen und der nächste kann beginnen. Wenn die Schleife fertig ist, wird das volle Array mit den gespeicherten Einträgen aus der Methode, zum Datengenerieren, zurückgegeben und diese ist damit auch vollends durchgeführt.

Als nächsten, aus der Konzeption definierten Prozessschritt, nach dem generieren der Daten, wird das aufstellen der Regeln umgesetzt. Hierfür wird die Methode „create_Rules“ mit den Parametern „request_values, request_bias, bias“ implementiert. Der Parameter „request_values“ ist ein Dictionary mit den Keys an den für die Bewertung relevanten Attributen und den dazugehörigen Ausprägungen in einem Array als Value. In diesem Fall sind es wie in der Konzeption definiert die Attribute Härte der Strafe und Laufende Strafe, welche wie in der folgenden Abbildung 3.5 im Dictionary angegeben werden.

```
request_values = {
    "Laufende_Strafe": [1,2,3,4,5],
    "Haerte_des_Vergehens": ["Leicht", "Mittel", "Hart"]
}
```

Abbildung 3.5: Codezeilen zum Erstellen eines Dictionary mit den zur Bewertung relevanten Attributen

Der Parameter „request_bias“ ist genau gleich aufgebaut, beinhaltet jedoch die Attri-

bute Geschlecht und Hautfarbe und deren Ausprägung. Dieser gibt die Liste der durch Verzerrung beeinflussbaren Attribute an. Der letzte Parameter „bias“ gibt die gewünschte Verzerrung ebenfalls wie die anderen Parameter an. In der Methode werden vier Rückgabewerte als Dictionaries generiert. Ein Dictionary für die Regeln der positiven Auswirkung, eines für die negative Auswirkung und nochmals die selben zwei Listen ergänzt durch die gewünschte Verzerrung.

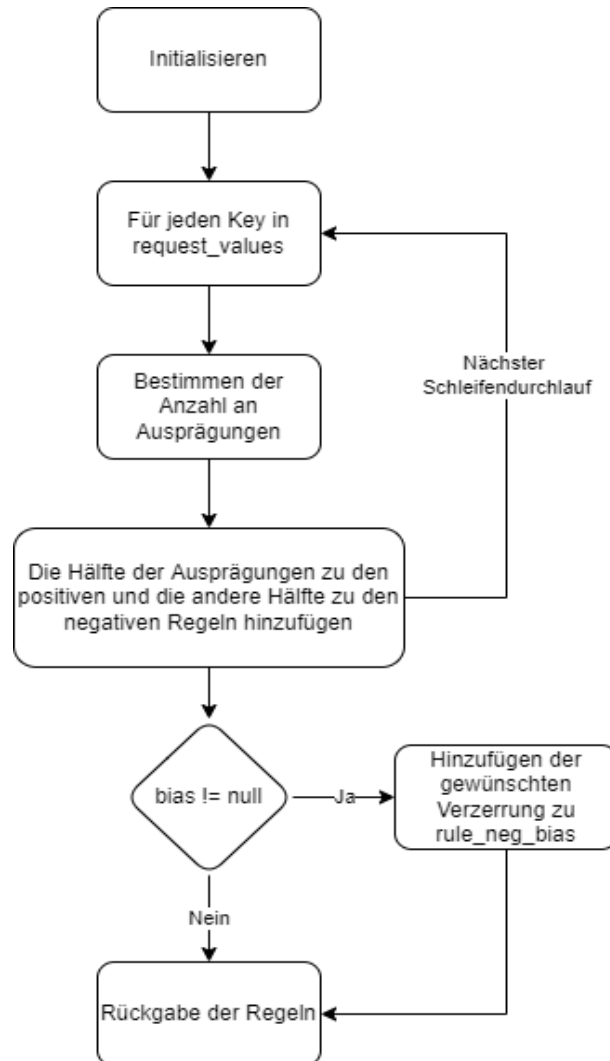


Abbildung 3.6: Programmablaufplan zur Generierung der Regeln von Szenario 1

In Abbildung 3.6 ist der Ablauf der Methode als Programmablaufplan skizziert. Zu Beginn werden die vier Dictionaries, in welchen die Regeln gespeichert werden, initialisiert. Daraufhin wird eine Schleife durch alle Keys des „request_values“ Dictionary durchlaufen. Darin werden als erstes die Anzahl an Ausprägungen des in diesem Durchlauf ausgewählten Attributes gezählt. Nach dem die Anzahl an Ausprägungen klar ist, kann die Mitte bestimmt werden. Anhand der Mitte werden die Ausprägungen unterhalb und gleich der Mitte dem Dictionary der negativen Regeln hinzugefügt und die restlichen oberhalb der

Mitte dem Dictionary der positiven Regeln. Wenn dies vollbracht ist, ist der erste Durchlauf beendet und es kann mit dem nächsten weiter gemacht werden. Solange bis alle „request_values“ den Regeln zugeordnet sind. Danach werden die erstellten Dictionaries für die positiven und negativen Regeln in die Dictionaries für die Bias Regeln kopiert. Im nächsten Punkt wird überprüft, ob ein Bias angegeben wurde. Wenn kein Bias angegeben wurde, werden die zuvor erstellten und kopierten Regeln als die vier Dictionaries zurückgegeben. Falls ein Bias angegeben wurde, wird für jeden Key in „request_bias“ überprüft, ob dieser dem Bias entspricht. Sobald der Key und damit das Attribut, welches verzerrt werden soll, gefunden ist werden die Ausprägungen, welche im Bias angegeben wurden, den negativen Bias Regeln hinzugefügt und die anderen übrig gebliebenen den positiven Bias Regeln hinzugefügt. So wird zum Beispiel bei Angabe: „bias: Hautfarbe:Weiß“ die Hautfarbe:Weiß den negativen Bias Regeln und die Hautfarbe:Schwarz den positiven Bias Regeln hinzugefügt. Daraufhin sind alle vier Dictionaries für die Regeln gefüllt und können zurückgegeben werden. Damit ist die Methode zum erstellen der Regeln vollständig umgesetzt. In der nächsten Zelle wurde eine Methode zum generieren eines Seeds/Startwerts umgesetzt. Dieser wird wie zu Beginn der Umsetzung beschrieben benötigt, um zu jedem Ausführungszeitpunkt unterschiedliche Zufallswerte zu erhalten. Daher sollte der Startwert ebenfalls immer variieren. Um dies zu schaffen werden momentane Zeitwerte genommen und miteinander verrechnet, sodass am Ende ein zeitabhängiger Wert resultiert. Zur Absicherung, falls der Wert Mal negativ oder Null sein sollte, wird ein zweiter alternativ Wert aus dem aktuellen Tag multipliziert mit den aktuellen Minuten plus Eins berechnet, da bei dieser Rechnung immer ein Wert größer Null resultiert. Der am Ende berechnete Startwert wird dann zurückgegeben und kann somit verwendet werden. Im nächsten Prozessschritt wird wie in der Konzeption beschrieben, das Erstellen der Bewertenden und die Methode zum Bewerten umgesetzt. Dafür wird eine Klasse namens „Evaluator“ erstellt, welche die Methoden „init“ und „rate“ implementiert. Der Aufbau der Methode „init“ ist in der folgenden Abbildung 3.7 dargestellt.

```
#Creating a evaluator with its own rules and a bias or not
def __init__(self, rule_pos, rule_neg, bias, percentage=0.2):
    self.rule_pos = rule_pos
    self.rule_neg = rule_neg
    self.bias = bias
    self.bias_percentage = percentage
```

Abbildung 3.7: Methode zur Initialisierung eines Bewertenden

Die Methode ist für das Erstellen eines Objektes mit den angegebenen Parametern zuständig. Dabei wird jedem „Evaluator“ Objekt ein Dictionary mit positiven Regeln und eines mit negativen Regeln sowie ein Boolean ob der Bewertende diskriminierend ist und

ein Prozentsatz, welcher die Stärke der Diskriminierung angibt, übergeben. Diese vier Angaben speichert jedes Objekt für sich und kann intern für sich selbst darauf zugreifen. Damit können dann die gewünschte Anzahl an Bewertenden und Bewertende welche diskriminieren erstellt werden. Durch die Angaben der eigenen Regeln, haben dann die diskriminierenden Bewertenden die Bias regeln und die neutralen Bewertenden die normalen Regeln. Zudem kann durch das Flag als Boolean ein diskriminierender von einem neutralen Bewerter unterschieden werden. Die zweite Methode der Klasse, welche „Evaluator“ implementiert und von jedem Objekt der Klasse verwendet werden kann, ist in der folgenden Abbildung 3.8 dargestellt.

```
#Function to evaluate a submitted application with or without bias
def rate(self, request, bias):
    #First 50/50 distribution
    pos = 50
    #Calculate the proportion according to which the decision is influenced positively or negatively.
    prop = 45/self.rule_pos.__len__()
    #Depending on how the rules match the request, the weight of the positive evaluation is shifted.
    for key in self.rule_pos.keys():
        if(self.rule_pos[key].__contains__(request[key])):
            pos += prop
        else:
            pos -= prop
    try:
        #If a bias is present, this is additionally taken into account with the Parameter in %
        if(self.bias):
            for b in bias:
                if(bias[b].__contains__(request[b])):
                    pos = pos*self.bias_percentage
            #Normalise positive value
            pos = pos/100
            #Determine negative value
            neg = 1-pos
            #Rating by chance with indication of pos and neg rating and adding the rating to the request.
            request["Bewertung"] = np.random.choice(["positiv", "negativ"], p=[pos, neg])
    except:
        print("Failure")
    return request
```

Abbildung 3.8: Methode eines Bewertenden zum Bewerten von Anträgen.

Die hier gezeigte Methode „rate“ wird von den Objekten der Klasse „Evaluator“ zum Bewerten eines Antrages verwendet. Als Parameter werden zum einen mit „self“ das Objekt welches die Methode aufruft, mit „request“ der zu bewertende Antrag und mit „bias“ die Verzerrung welche ausgewirkt werden soll übergeben. Als Rückgabe wird der erhaltene

Antrag mit Ergänzung der Bewertung zurückgegeben. Im Nachfolgenden wird erläutert wie die Bewertung in der Methode abläuft. Zu Beginn wird die Wahrscheinlichkeitsverteilung zwischen der positiven und negativen Bewertung auf 50 zu 50 Prozent gesetzt. Somit ist die Entscheidung noch offen. Danach wird eine Proportion bestimmt, durch welche sich eine im Antrag positive oder negative Ausprägung auf die Wahrscheinlichkeitsverteilung der Entscheidung auswirkt. Diese wird wie folgt errechnet: 45 geteilt durch die Anzahl an Attributen in den Regeln. Es wird mit 45 gerechnet, da es immer noch eine rest Wahrscheinlichkeit geben soll, falls alle Ausprägungen im Antrag positiv oder negativ ausfallen. Als nächstes werden die Ausprägungen der Attribute im Antrag durch eine Schleife mit den Regeln verglichen, um aus den Regeln zu entscheiden, ob die Wahrscheinlichkeitsverteilung der Entscheidung die Proportion hinzu oder abgezogen wird. So tendiert die Entscheidung am Ende mehr zu einer positiven oder negativen Entscheidung abhängig von den im Antrag angegebenen Ausprägungen und den Regeln des Bewertenden. Um nach der Bewertung noch eine potentielle Diskriminierung einzubringen, wird überprüft ob das Flag des Bewertenden zur Diskriminierung gesetzt ist. Wenn die Bewertende Person eine diskriminierende und die gewünschte Bias Ausprägung in dem Antrag vorhanden ist, wird die Wahrscheinlichkeit für eine positive Auswirkung durch die Code Zeile: `pos = pos*self.bias_percentage` mit dem für den Bewertenden angegebenen Prozentsatz verringert. Somit ist dadurch die Möglichkeit auf eine positive Bewertung deutlich gesunken. Nachfolgend wird dann noch die Wahrscheinlichkeitsverteilung für die Entscheidung passend umgewandelt, um beruhend auf diesen Wahrscheinlichkeiten die Entscheidung zu treffen und das Ergebnis im Antrag also dem Dataframe anzuhängen. Zum Schluss wird der überarbeitete Antrag wieder zurückgegeben und das Bewerten ist abgeschlossen.

Nach Abschluss der Klasse werden nun noch zwei Methoden benötigt, welche den gesamt Ablauf durchführen und die anderen Methoden vereinen. Zum einen die Methode `„generate_data“`, welche die Anzahl an zu generierenden Datensätzen übermittelt bekommt. In der Methode wird zuerst ein Seed durch Aufruf der oben beschriebenen Methode zum Seed/Startwert generieren erzeugt. Danach kann die Methode für das Datengenerieren mit dem Seed und der Anzahl an Daten aufgerufen werden. Der Rückgabewert wird dann in ein neuen Pandas Dataframe geschrieben und zurückgegeben. Die andere Methode, welche implementiert wird, ist für den gesamt Ablauf der Datenbewertung zuständig. Diese trägt den Namen `„work“` und bekommt den Datensatz, die gewünschte Verzerrung, die Anzahl der Bewertenden, die Anzahl der diskriminierenden Bewertenden und die Prozentual Auswirkung der Verzerrung als Parameter übergeben. Als erstes werden in der Methode die beiden benötigten Dictionaries für `„request_values` und `request_bias“` angelegt und mit den Werten des Szenarios wie in der folgenden Abbildung 3.9 zu sehen gefüllt.

Danach werden daraus die Regeln bestimmt und gespeichert. Im Anschluss wird die ge-

```
#Values in the request which influence the evaluation
request_values = {
    "Laufende_Strafe": [1,2,3,4,5],
    "Haerte_des_Vergehens": ["Leicht", "Mittel", "Hart"]
}
#Values in the request which can have an effect on the evaluation as a bias
request_bias = {
    "Hautfarbe": ["Schwarz", "Weiß"],
    "Geschlecht": ["M", "W"]
}
```

Abbildung 3.9: Dictionaries gefüllt mit Standard Werten des Szenario 1

wünschte Anzahl an „Evaluator“ Objekte erzeugt und diesen die Regeln übergeben. Um die Verzerrung umzusetzen wird danach die gewünschte Anzahl der „Evaluator“ Objekte in diskriminierende Bewertende umgewandelt und dessen Regeln ausgetauscht durch die Bias Regeln. Damit sind alle Vorbereitungen abgeschlossen und das eigentliche Bewerten der Anträge kann beginnen. Dafür wird eine Schleife über den Dataframe der Anträge durchlaufen. Für jeden Antrag wird dann zufällig bestimmt, welches „Evaluator“ Objekt den Antrag bewertet. Nachdem der Antrag bewertet wurde wird dieser der Liste der fertigen Anträge hinzugefügt. Sobald alle Anträge bewertet sind und die Schleife daher durchlaufen ist, wird die Liste der fertigen Anträge zu einem Dataframe umgewandelt und aus der Methode zurückgegeben. Damit sind auch die letzten beiden Methoden umgesetzt und es muss letztendlich nur noch eine finale Zelle zur Ausführung des gesamten Programmes erstellt werden.

In der Abbildung 3.10 ist die letzte Zelle für die Interaktionen des Benutzenden dargestellt. Im oberen Teil der Zelle haben die Benutzenden die Möglichkeit, Anpassungen am Datengenerator zu machen. So kann hier die gewünschte Verzerrung, die Größe des Datensatzes, die Anzahl an Bewertenden, die Anzahl der diskriminierenden unter den Bewertenden und der Prozentuale Einfluss der Verzerrung angepasst werden. Im unteren Teil wird dann die Hauptmethode zum Datengenerieren (`generate_data`) aufgerufen und im Anschluss durch die Methode „work“ die zuvor generierten Anträge bewertet. Beide Datensätze werden separat als Variablen geführt, sodass am Ende der Methode die Ursprungsdaten als „Daten.csv“ und der bewertete Datensatz als „Daten_Bewertet.csv“ abgespeichert werden.

Insgesamt ist mit dieser letzten Zelle die gesamte Umsetzung des ersten Szenarios als „Szenario1.ipynb“ Datei abgeschlossen und kann so direkt verwendet werden, um Daten zu generieren.

```
#Here is the section for the possible parameters to enter
#This dictionary specifies the bias(es) on a possible attribute
bias = {
    "Hautfarbe": ["Schwarz"]
}
#The number of datasets that are to be generated
datasets=10000
#The number of evaluators who evaluate entries
evaluator_count=10
#The number of evaluators who evaluate with a bias
bias_evaluator=4
#This decides how strong the bias will be. The higher the stronger.
bias_percentage=0.2

#Dont touch this
data = generate_data(datasets)
finished = work(data,bias,evaluator_count,bias_evaluator,bias_percentage)
data.to_csv("Daten.csv", sep=';', encoding='utf-8', index=False)
finished.to_csv("Daten_Bewertet.csv", sep=';', encoding='utf-8', index=False)
```

Abbildung 3.10: Letzte Zelle des Szenario 1 für die Benutzenden Interaktion

3.3.2 Umsetzung des Szenario 2

Das zweite Szenario wird ebenfalls wie das erste in eine Python Notebook umgesetzt. Das File trägt den Namen „Szenario2.ipynb“ und beginnt sowie auch Szenario 1 mit dem laden der benötigten Bibliotheken. Auch in diesem werden die gleichen Bibliotheken wie in Abbildung 3.3 dargestellt geladen. Einzige Ausnahme hierbei liegt darin, dass das Faker Objekt nicht wie in der Abbildung gezeigt ist erstellt wird. In diesem Fall wird es wie folgt erstellt: „fake = Factory.create("de_DE“)“. Der Grund dafür, liegt darin, dass in diesem Szenario Deutsche Namen generiert werden sollen, da es sich um ein auf Deutschland angelegtes Szenario handelt. In der zweiten Zelle des Notebooks wird die Methode für das erstellen eines Seed/Startwert aus Szenario 1 implementiert. Diese verändert sich nicht und kann daher so verwendet werden.

Als nächstes ist die Methode „create_fake_data“ mit den Parameter num und seed umgesetzt. In dieser wird die durch num angegebene Anzahl an Daten generiert. Der Parameter seed wird zu beginn der Methode als Startwert für die Zufallsmethoden von numpy und Faker gesetzt. Als nächstes wird eine Schleife über num begonnen, in welcher in jedem Durchlauf einen Eintrag für den Datensatz generiert wird. In jedem Durchlauf müssen die Ausprägungen der für dieses Szenario bestimmte Attribute Name, Alter, Politische Orientierung, Bildungsabschluss, Soziales, Wohnlage und CO2-Fußabdruck, unter Betrachtung

der im Kapitel 3.2 angegebenen Abhängigkeiten, bestimmt werden. Der Name eines Eintrages wird durch die Faker Instanz wie folgt „Name": fake.first_name()“ generiert. Die anderen Ausprägungen der Attribute werden, um das Szenario so real wie möglich zu halten, beruhend auf Wahrscheinlichkeiten aus Statistiken bestimmt. Die Wahrscheinlichkeiten der verschiedenen Attributen ergeben sich wie folgend aufgeführt:

Alter

Ein Alter für eine Person wird zwischen 20 und 79 Jahren bestimmt. Zur Bestimmung wird aufgrund der verwendeten Statistik in drei Altersgruppen unterteilt, welche unterschiedliche Wahrscheinlichkeiten besitzen. Die Altersgruppen sind von 20 bis 39, 40-59 und 60 bis 79 Jahren.

Die Formel zur Berechnung der Wahrscheinlichkeiten:

Gegeben: Gesamt(Personen im Alter zwischen 20-79) = 20,36 + 23,38 + 18,15 = 61,89 Millionen

$$\text{Altersgruppe}\% = \frac{\text{Altersgruppe}}{\text{Gesamt}} \quad (3.3)$$

Altersgruppen	Berechnung	Wahrscheinlichkeit
20-39	$\frac{20,36}{61,89}$	33%
40-59	$\frac{23,38}{61,89}$	38%
60-79	$\frac{18,15}{61,89}$	29%

Tabelle 3.6: Tabelle zur Bestimmung der Wahrscheinlichkeiten für die Altersgruppen

Die in der Tabelle 3.6 nach der Formel 3.3 errechneten Wahrscheinlichkeiten beruhen auf der Statistik des Statistischen Bundesamt vom 31. Dezember 2020.[28] Die Zahlen wurden passend zu den Altersgruppen aus der Statistik entnommen. Um nun aus den errechneten Wahrscheinlichkeiten auf ein Alter zu kommen, wird im Code zu erst die Altersgruppe nach den Wahrscheinlichkeiten bestimmt. Nachdem die Altersgruppe klar ist, wird eine Zufallszahl im Bereich der Altersgruppe bestimmt, welche dann das endgültige Alter ist. Dadurch wird eine gute Verteilung des Alters entsprechend der Statistik gewährleistet.

Politische Orientierung

Bei der Bestimmung der Politischen Orientierung in Links, Mitte und Rechts wird wie in der Abbildung 3.2 aus Kapitel 3.2.2 dargestellt, dass Alter als Grundlage mit verwendet. Daher ist die Berechnung und damit auch die Wahrscheinlichkeitsverteilung pro Altersgruppe für die Politische Orientierung unterschiedlich. Des weiteren wird, um die Daten so realitätsnahe wie möglich zu halten, eine statistische Auswertung des Wahlverhaltens

bei der Bundestagswahl 2017 nach Alter hinzugezogen. Da in der Statistik die Ergebnisse in Prozent pro Partei angegeben sind, wurde nach eigenem Ermessen entschieden, welche Partei zu welcher Ausprägung zugeordnet wird. Zur Ausprägung Mitte gehört dabei die CDU, SPD, CSU und FDP zu Links Die Linke und Grüne und zu Rechts wird die AFD zugeordnet. Daraus lassen sich dann die in der folgenden Tabelle 3.7 aufgelisteten Wahrscheinlichkeiten nach dem Alter berechnen.[29, S. 33]

Ausprägungen	20-24	25-34	35-44	45-59	60-69	70-79
Links	28,0%	24,4%	21,6%	20,7%	17,7%	10,7%
Mitte	63,1%	61,4%	61,8%	63,5%	68,7%	80,9%
Rechts	8,9%	14,2%	16,6%	15,8%	13,6%	8,4%

Tabelle 3.7: Tabelle der Wahrscheinlichkeiten für die Politische Orientierung nach Alter

Die in der Tabelle 3.7 dargestellten Wahrscheinlichkeiten wurden nach den dafür angegebenen Altersgruppen aus der Tabelle der Veröffentlichung von der Konrad-Adenauer-Stiftung e. V. berechnet. Dafür wurden die Prozente der oben zugeordneten Parteien aufsummiert. So ergeben sich die Prozentpunkte für Links, Mitte und Rechts. Jedoch fehlen zu den 100% pro Altersgruppe die Prozentpunkte der Sonstigen Parteien. Da diese nämlich keiner Richtung zugeordnet werden können müssen die Prozente der sonstigen Insgesamt verrechnet werden. Nach der Verrechnung der Sonstigen Parteien entstehen dann die in der Tabelle dargestellten Wahrscheinlichkeiten.[29, S. 33] Zur Umsetzung der Bestimmung der Politischen Orientierung wird eine Reihe an ifelse Abfragen über das zutreffende Alter durchlaufen, um dann mit den Altersentsprechenden Wahrscheinlichkeiten die Politische Orientierung zu bestimmen. Der Code für die Wahrscheinlichkeitsauswahl sieht dabei genau wie im Beispiel aus der Abbildung 3.4 dargestellt aus.

Bildungsabschluss

Der Bildungsabschluss einer Person basiert ebenfalls wie die Politische Orientierung auf dem Alter. Zusätzlich wird wie auch schon zuvor eine Statistik aus einem Datenreport des Jahres 2021 vom Statistischen Bundesamt verwendet. [30] Dadurch können die Daten so realitätsnahe wie möglich gehalten werden.

Die in der Tabelle 3.8 dargestellten Wahrscheinlichkeiten für die dort aufgeführten Altersgruppen ergeben sich aus der „Tab 9“ des Datenreports vom Statistischen Bundesamt. Die Daten der „Tab 9“ stammen aus dem Jahr 2018 der deutschen Bevölkerung. Für die Altersgruppe von 20 bis 29 Jahre existiert keine passende Angabe in der Quelle, da in dieser erst ab dem Alter von 25 Jahren Werte aufgelistet sind. Einfachheitshalber wird daher für die Personen kleiner gleich 29 Jahre die Werte der Personen zwischen 25 und 29 Jahren genommen. Dasselbe gilt auch für die Personen älter als 60, hier werden in

Ausprägungen	<=29	30-39	40-49	50-59	>=60
Ausbildung	42,5%	45,2%	51,6%	55,9%	54,4%
Fachschulabschluss	8,1%	9,1%	9,7%	11,3%	9,6%
Bachelor	11,7%	5,9%	1,4%	0,5%	0,2%
Master	7,1%	5,2%	1,1%	0,3%	0,1%
Diplom	5,9%	15,7%	17,9%	15,9%	13,2%
Promotion	0,3%	1,6%	1,7%	1,4%	1,2%
Ohne	24,4%	17,3%	16,6%	14,7%	21,3%

Tabelle 3.8: Tabelle der Wahrscheinlichkeiten für den Bildungsabschluss nach Alter

der Quelle nur eine Wertereihe für Personen mit 60 und älter angegeben. Das Szenario bezieht sich jedoch nur bis 79 Jahre und passt somit hier auch nicht vollkommen. Zur Vereinfachung wird hierbei jedoch diese Wertereihe für alle zwischen 60 und 79 Jahren verwendet.[30]

Soziales

Das soziale Engagement wird von 0 bis 3 angegeben und basiert wie in der Konzeption beschrieben, um die Daten interessanter zu machen, auf der Politischen Orientierung. Durch solche Abhängigkeiten gestalten sich die Daten nämlich realitätsnaher und bringen eine größere Variabilität mit. Da jedoch für die Werte keine Referenzen aus der realen Welt zur Verfügung stehen, wird mit fiktiven Werten gearbeitet.

Ausprägungen	Links	Mitte	Rechts
0	40%	60%	75%
1	25%	20%	15%
2	25%	15%	9%
3	10%	5%	1%

Tabelle 3.9: Tabelle der Wahrscheinlichkeiten für das Soziale Engagement nach Politischer Orientierung

Die in der Tabelle 3.9 aufgeführten fiktiven Wahrscheinlichkeiten sind ebenso wie schon bei der Bestimmung der anderen Attribute, durch den Code aus Abbildung 3.4 umgesetzt. Jedoch wird durch eine if else Abfrage die unterschiedlichen Politischen Orientierungen unterschieden und dementsprechend die passenden Wahrscheinlichkeiten in der Codezeile verwendet.

Wohnlage

Die Wohnlage wird durch die Ausprägungen Ländlich, Vorort, Kleinstadt und Großstadt angegebenen und wird durch kein anderes Attribut beeinflusst. Die Wahrscheinlichkeiten für die einzelnen Ausprägungen sind in nachfolgender Tabelle aufgezeigt.

Zur Bestimmung der Wahrscheinlichkeiten für die unterschiedlichen Ausprägungen der

Ausprägungen	Einwohnergrenze	Wahrscheinlichkeit
Ländlich	≤ 2000	5,38%
Vorort	2.000 - 5000	8,5%
Kleinstadt	5.000 - 20.000	26,6%
Großstadt	>20.000	59,52%

Tabelle 3.10: Tabelle der Wahrscheinlichkeiten für die Wohnlage

Wohnlage müssen zuerst die in der Tabelle 3.10 angegebenen Einwohnergrenzen pro Ausprägung festgelegt werden. Für die Festlegung wurde nach eigenem Ermessen entschieden, welche Ausprägung zu welcher Einwohnergrenze passt. Aus einer Publikation des Statistischen Bundesamt können dann nach dieser Eingliederung die Wahrscheinlichkeiten aufsummiert werden. In der Publikation „Gemeinden nach Bundesländern und Einwohnergrößenklassen am 31.12.2020“ sind der prozentuale Anteil an deutschen Einwohner in gewissen Einwohnergrenzen angegeben. Diese werden dann nach den hier für die Ausprägungen definierten Grenzen aufsummiert und in der Tabelle angegebenen. Damit kann eine möglichst Realitätsnahe Datenerzeugung gewährleistet werden. Für die Umsetzung der Auswahl der Ausprägung kann wie auch schon in den vorigen Fällen der selbe Code aus Abbildung 3.4 auf den Fall hier angepasst und verwendet werden. So sieht die finale Codezeile wie in der folgenden Abbildung dargestellt aus.

```
location = np.random.choice(["Großstadt", "Kleinstadt", "Vorort", "Ländlich"], p=[0.5952, 0.266, 0.085, 0.0538])
```

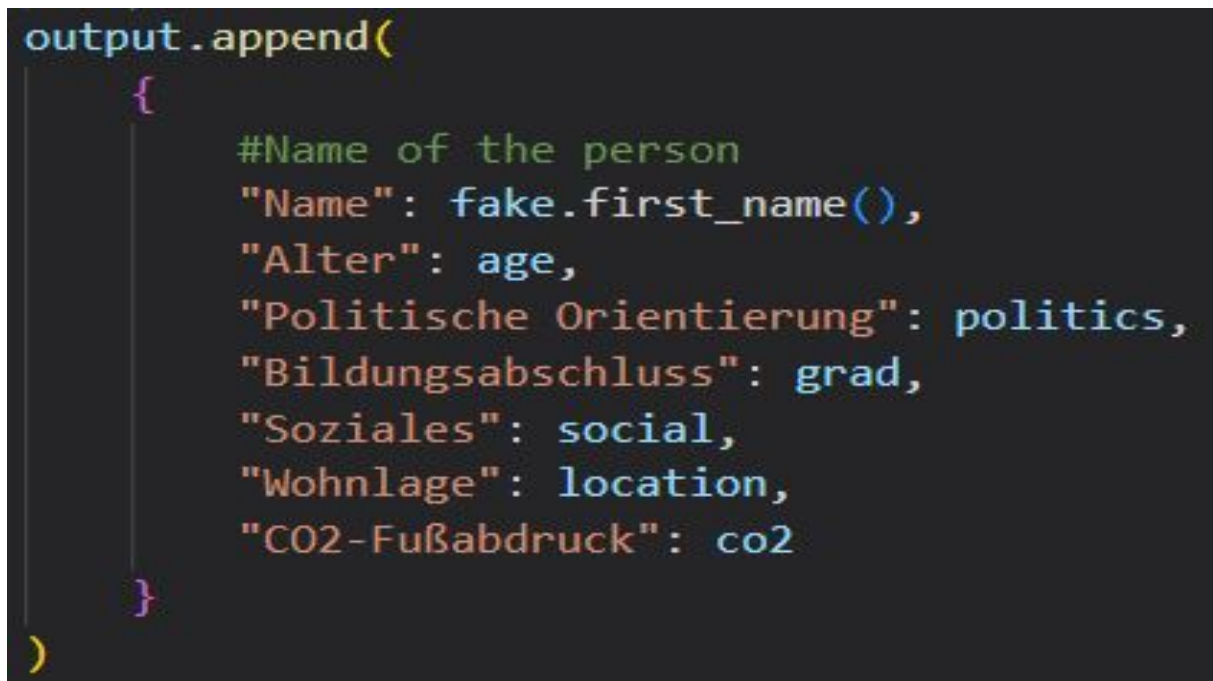
Abbildung 3.11: Codezeile zur Auswahl der Ausprägung der Wohnlage basierend auf angegebenen Wahrscheinlichkeiten

CO2-Fußabdruck

Der CO2-Fußabdruck ist das komplexeste Attribut des Szenarios, da dieser auf zwei Attributen basiert. Wie in der Abbildung 3.2 dargestellt hat das Soziale Engagement und die Wohnlage eine Auswirkung auf den CO2-Fußabdruck. Dadurch, dass das Soziale Engagement ebenfalls noch von der Politischen Orientierung und diese vom Alter abhängig ist, haben auch noch zwei weitere Attribute eine indirekte Auswirkung auf den CO2-Fußabdruck. Dies lässt sich auch dadurch begründen, dass der CO2-Fußabdruck in der Realität sich aus vielen unterschiedlichen Teilen zusammensetzt. Da es für den CO2-Fußabdruck unter Berücksichtigung dieser Auswirkungen keine Referenzen aus der realen Welt gibt, wurden fiktive Werte für eine Gaußkurve verwendet. Als Referenzen für die Angabe der Ausprägung des CO2-Fußabdruck in Tonnen pro Kopf konnte jedoch der Mittelwert von 7,91 Tonnen herangezogen werden.[31] Um daraus ein Intervall um diesen Wert als Mittelwert festzulegen. So ergab sich das Intervall von 4 bis 12 Tonnen in 0,5er Schritten. Für die daraus resultierenden 17 Ausprägungen wurde eine Gaußkurve erzeugt. Um die Auswirkung von Sozialem Engagement und Wohnlage durchzusetzen wird die

Gaußkurve, je nach Grundlage der Attribute verschoben. Für die Umsetzung der jeweiligen Verschiebung durch die beiden Attribute Soziales Engagement und Wohnlage muss eine große ifelse Abfragestruktur mit 16 unterschiedlichen Zuständen erstellt werden, da beide Attribute 4 verschiedene Ausprägungen haben ($4 \times 4 = 16$). In den 16 unterschiedlichen Zuständen wird dann die Codezeile aus der Abbildung 3.11, um die 17 Ausprägungen des CO₂-Fußabdruck angepasst und die Wahrscheinlichkeiten der Gaußkurve entsprechend dem Zustand eingetragen. Dadurch entsteht ein recht komplexes Codekonstrukt, welches jedoch sehr anschauliche abhängige Daten liefert und damit das Szenario in der Auswertung interessanter gestaltet.

Mit der abschließenden Bestimmung des CO₂-Fußabdruck sind alle Ausprägungen der Attribute bestimmt und die Werte können nun als ein Dictionary dem Array des gesamt Datensatzes hinzugefügt werden.

The image shows a code editor with a dark background. The code is written in Python and is a snippet for appending a dictionary to a list. The code is as follows:

```
output.append(  
    {  
        #Name of the person  
        "Name": fake.first_name(),  
        "Alter": age,  
        "Politische Orientierung": politics,  
        "Bildungsabschluss": grad,  
        "Soziales": social,  
        "Wohnlage": location,  
        "CO2-Fußabdruck": co2  
    }  
)
```

Abbildung 3.12: Codeausschnitt zum Hinzufügen eines Dateneintrags

In der Abbildung 3.12 ist der Codeausschnitt zu sehen, wie das Dictionary mit der Zuordnung Attribut als Key und die Ausprägung als Value erstellt und dem Array „output“ angehängt wird. Die Ausprägung des Namens wird in diesem Schritt erst bestimmt und die anderen werden durch die Variablen übertragen. Im Anschluss ist ein Schleifendurchlauf beendet, somit ein Eintrag dem Datensatz hinzugefügt und es wird falls noch offen der nächste Durchlauf begonnen. Sobald die Schleife beendet ist, ist der gesamte gewünschte Datensatz generiert und wird letztendlich nur noch als Array aus der Methode zurückgegeben.

Als nächste Zelle wird wie auch schon im Szenario 1 eine Methode, welche das Datener-

stellen zusammenfasst umgesetzt. Diese ist genau gleich wie im Kapitel ?? beschrieben umgesetzt. Zuerst wird ein Seed und danach mit diesem die gewünschte Anzahl an Daten generiert, als Dataframe gespeichert und zurückgegeben.

In der darauffolgenden Zelle ist die Methode „create_Rules()“ für das erstellen der Regeln implementiert. Diese Methode ist deutlich einfacher implementiert als sie im Szenario 1 implementiert werden musste. In diesem Szenario werden die Regeln fix als Dictionary angegeben erstellt.

```
def create_Rules():
    influential = {
        "Politische Orientierung": [-50, 50, -50],
        "Bildungsabschluss": [20, 40, 60, 80, 100, 120, -120],
        "Soziales": [-70, 0, 40, 100],
        "Wohnlage": [-30, 20, 20, -30],
        "CO2-Fußabdruck": [110, 70, 40, 10, 0, -20, -40, -70, -110]
    }
    return influential
```

Abbildung 3.13: Codeausschnitt zum Erstellen des Regel Dictionary

Das Dictionary der Regeln, welches in der Methode erstellt und zurückgegeben wird, ist in der Abbildung 3.13 dargestellt. Für jedes Attribut, welches in der Konzeption unter 3.1.2 als beeinflussendes Attribut eingestuft wird, muss ein Key Value Paar dem Dictionary hinzugefügt werden. Als Keys werden die Attribute und in den Values nach der in der zur Generierung angegebenen Reihenfolge eine Punktzahl für jede Ausprägung angegeben. Die angegebenen Punkte pro Ausprägung beeinflussen dann in diesem Maße den Social Score einer Person. So kann der Bewertende für jedes Attribut nach der Ausprägung und der zugehörigen Punktzahl in den Regeln schauen. So setzt sich dann Stück für Stück der Social Score einer Person zusammen. Die Punkte für die jeweiligen Ausprägungen wurden fiktiv bestimmt. Dabei wurde darauf geachtet, dass es in jedem Attribut sowohl positive als auch negative Auswirkungen von Ausprägungen vorhanden sind. Zudem darf die maximal mögliche Punktzahl nicht 1400 übersteigen und die minimale nicht unter 600 fallen. Zum Schluss wird das erstellte Dictionary aus der Methode zurückgegeben und kann dann als Regelwerk zur Bewertung verwendet werden.

Im nächsten Schritt ist wie auch schon im ersten Szenario die Klasse „Evaluator“ mit der Methode „__init__“ zum erstellen eines Objektes und „rate“ für das Bewerten von Personen. Für die Initialisierung eines „Evaluator“ Objektes wird zum einen ein Regel Dictionary, ein Boolean ob die Bewertende diskriminierend ist und eine Punktzahl für die negative Auswirkung einer Diskriminierung benötigt. Diese Parameter speichert dann

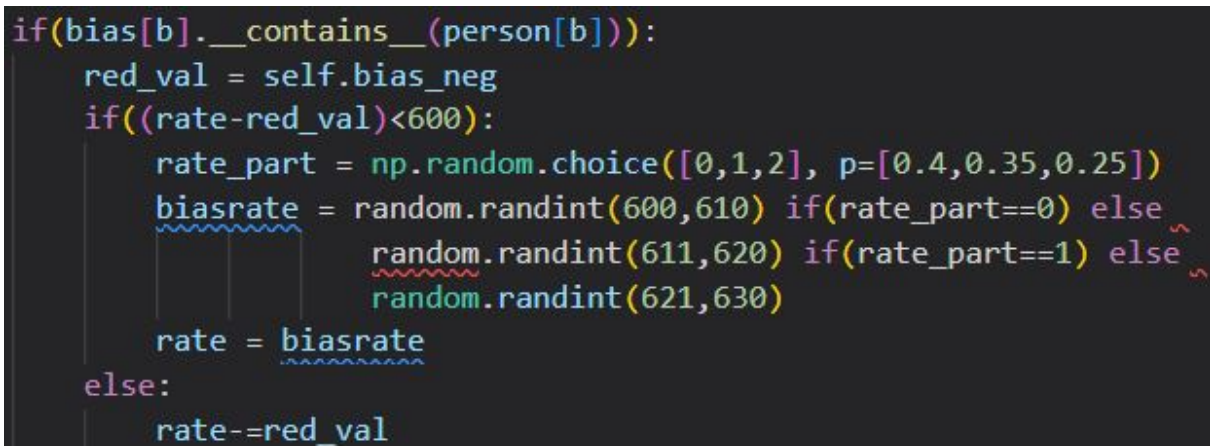
jedes erzeugte Objekt einzeln für sich. Die zweite Methode für das Bewerten der Personen bekommt die drei Parameter „influential“, „person“ und „bias“ übergeben. In „influential“ sind als Dictionary alle Attribute des Szenario bis auf das Alter und der Name mit deren Ausprägungen gegeben. In dem Parameter „person“ ist der Datenpunkt der zu bewertenden Person enthalten und in „bias“ ist die eine oder mehrere gewünschte Verzerrungen als Dictionary angegeben. Zu Beginn der Methode wird der momentan Bewertete Score auf 1000 Punkte gesetzt.

```
for key in self.rules.keys():
    if(key == "CO2-Fußabdruck"):
        value_of_key = int(person[key])
        index = influential[key].index(value_of_key)
        rate += self.rules[key][index]
        procent = self.rules[key][index] * 0.15
        procent = round(procent,0)
        if(procent<0):
            rand = random.randint(0,(procent*-1))
            rate -= int(rand)
        else:
            rand = random.randint(0,procent)
            rate += int(rand)
```

Abbildung 3.14: Codeausschnitt der Funktion zum Bewerten von Personen

Daraufhin folgt die in der Abbildung 3.14 zu teilen dargestellte Schleife über alle Keys, somit Attributen, in den Regeln des Bewerter. In der Schleife wird für jedes Attribut dann die Auswirkung auf den Score bestimmt. Dafür muss zuerst überprüft werden, ob es sich um das Attribut CO2-Fußabdruck handelt oder nicht, da bei diesem nicht für jeden Komma fünf Wert eine Auswirkung in den Regeln definiert ist. Somit hat 4 und 4,5 die gleiche Auswirkung. Diese Besonderheit des CO2 Fußabdruck muss daher in der Codezeile „value_of_key = int(person[key])“ durch Parsen des Values zu einem Integer besonders behandelt werden. Die darauf folgenden Zeilen Code sind jedoch für jedes Attribut dieselben. In der nächsten Zeile wird für die erhaltene Ausprägung des momentan in der Schleife bearbeitenden Attributes der Index in der Liste von allen Ausprägungen dieses Attributes bestimmt. Mit dem Index kann daraufhin nämlich die Auswirkung dieser Ausprägung aus den Regeln entnommen und auf den momentanen Score addiert werden. Damit ist die Auswirkung schon mal in den Score mit ein berechnet. Um die Zahlen je-

doch interessanter zu halten, wird danach noch ein zusätzlicher prozentualer Wert der Auswirkung bestimmt, welcher ebenfalls auf den Score berechnet wird. Dafür werden 15 Prozent der Auswirkung genommen und dann ein Zufallswert zwischen 0 und diesen 15 Prozent bestimmt und dem Score hinzugefügt beziehungsweise (bzw.) abgezogen. Danach ist der Durchlauf für dieses eine Attribut beendet und der Score wurde entsprechend der Ausprägung der zu bewertenden Person im Bezug auf die Regeln des Bewertenden angepasst. Sobald die Schleife fertig ist, wird im nächsten Schritt überprüft ob die Bewertende Person diskriminierend ist. Wenn dies der Fall ist, beginnt eine Schleife über alle im Parameter „bias“ angegebenen gewünschten Verzerrungen. Wie auch schon zuvor gibt es hier einen Sonderfall für das Attribut Alter, da die gewünschte Verzerrung für ein Alter in einem Intervall angegeben wird. Daher muss hier überprüft werden, ob das Alter der zu bewertende Person im Intervall liegt oder nicht. Der nachfolgende Code ist dann der selbe für alle anderen Attribute bis auf den Namen.



```

if(bias[b].__contains__(person[b])):
    red_val = self.bias_neg
    if((rate-red_val)<600):
        rate_part = np.random.choice([0,1,2], p=[0.4,0.35,0.25])
        biasrate = random.randint(600,610) if(rate_part==0) else
                    random.randint(611,620) if(rate_part==1) else
                    random.randint(621,630)
        rate = biasrate
    else:
        rate-=red_val

```

Abbildung 3.15: Codeausschnitt für das Hinzufügen eines Bias beim Bewerten der Personen

In der ersten Zeile des Ausschnittes von Abbildung 3.15 wird dann überprüft, ob die Person die zu diskriminierende Ausprägung des Attributes besitzt. Wenn dies der Fall ist, muss überprüft werden, dass wenn die Auswirkung des Bias der Person vom Score abgezogen wird, der Score trotzdem noch über 600 liegt. Falls der Score dann unter 600 liegt, wird ein Score zu 40% zwischen 600-610 oder zu 35% zwischen 611-620 oder zu 25% zwischen 621-630 zufällig für die Person bestimmt. Dadurch wird gewährleistet, dass nicht sehr viele Personen am Ende einen Score von genau 600 haben. Wenn der Score nicht unter 600 fallen würde, wird die Auswirkung des Bias vom Score der Person abgezogen. Zum Abschluss der Funktion wird nochmals überprüft ob die Grenzen des Scores überschritten wurden und falls zutreffend der Score auf die jeweilige Grenze zurück gesetzt. Danach wird dem Datenpunkt des Dataframes die Bewertung als Integer hinzugefügt und der Datenpunkt kann zurückgegeben werden.

Als nächste und letzte Methode wird die Methode für den gesamt Ablauf des Bewertens Namens „work“ umgesetzt. Diese ist sehr ähnlich wie auch schon bei Szenario 1 umgesetzt. Zu aller erst ist ein Dictionary mit allen Attributen, bis auf Name und Alter, und deren Ausprägungen implementiert. Danach wird durch Aufruf der Methode zum erstellen der Regeln, diese in eine Variable gespeichert. Im Anschluss kann die gewünschte Zahl an „Evaluator“ Objekten unter Angabe der zuvor erstellten Regeln erstellt werden. Von diesen Objekten wird dann noch die gewünschte Anzahl zu diskriminierenden umgewandelt und diesen die negative Bias Auswirkung übergeben. Zum Schluss der Methode wird über den erhaltenen Datensatz iteriert und jeder Datenpunkt von einem zufälligen „Evaluator“ Objekt bewertet. Nach der Bewertung werden die bewerteten Datenpunkte separat abgespeichert und als Dataframe aus der Methode zurückgegeben. Damit ist die letzte Methode vollends implementiert und es kann in einer letzten Zelle des Notebooks der gesamt Ablauf zusammengefasst werden.

```
#Here is the section for the possible parameters to enter
#This is an example of how a bias on age can be indicated
age_sample = {
    "Alter": "20-30"
}
#This dictionary specifies the bias(es) on a possible attribute
bias = {
    "Alter": "20-30",
    "Bildungsabschluss": ["Ausbildung", "ohne"]
}
#The number of datasets that are to be generated
datasets = 10000
#The number of evaluators who evaluate entries
evaluatorcount = 10
#The number of evaluators who evaluate with a bias
bias_evaluator = 4
#This decides how strong the bias will be. The higher the stronger.
negativ_bias_impact = 200

#Dont touch this
data = generate_data(datasets)
finished = work(df=data, bias=bias, evaluator_count=evaluatorcount, bias_evaluator=bias_evaluator, bias_neg=negativ_bias_impact)
data.to_csv("Daten_Szenario2.csv", sep=';', encoding='utf-8', index=False)
finished.to_csv("Daten_Bewertet_Szenario2.csv", sep=';', encoding='utf-8', index=False)
```

Abbildung 3.16: Codeausschnitt für das Hinzufügen eines Bias beim Bewerten der Personen

In der Abbildung 3.16 ist die letzte Zelle dargestellt. Im oberen Bereich kann die Benutzende Person Anpassungen vornehmen, wie den Bias festlegen, die Datenset Größe bestimmen, die Anzahl der Bewertenden und diskriminierenden festlegen sowie die Auswirkung des Bias bestimmen. Als Beispiel wurden hierbei zwei mögliche Versionen für einen Bias angegeben. Im unteren Teil ist der Ablauf des gesamt Programms abgebildet. Zuerst werden die Daten durch die „generate_data“ Methode in der gewünschten Anzahl generiert. Danach werden diese Daten mit Hilfe der „work“ Methode bewertet und als neuen Dataframe zurückgegeben. Zum Schluss werden dann noch die Rohdaten als „Daten_Szenario2.csv“ Datei und die bewerteten Daten als „Daten_Bewertet_Szenario2.csv“ Datei abgespeichert. Hiermit ist die gesamte Umsetzung des Szenario 1&2 abgeschlossen

und beide Szenarien können getrennt voneinander als Notebooks verwendet werden, um Daten zu generieren.

Im nächsten Kapitel wird die Auswertung der generierten Daten beschrieben.

3.4 Datenauswertung

3.5 Evaluation der Ergebnisse

4 | Schluss

4.1 Zusammenfassung

- Fazit ziehen!!!

4.2 Diskussion

Kritische Reflektieren der gesamten Arbeit.

4.3 Ausblick

Literatur

- [1] M. Kershner, “Data Isn’t The New Oil — Time Is,” 2021, Letzter Zugriff: 22.05.2022 [Online]. Verfügbar:<https://www.forbes.com/sites/theyec/2021/07/15/data-isnt-the-new-oil--time-is/?sh=45eb63a135bb>.
- [2] B. Otto, D. Lis, J. Jürjes u. a., *Data Ecosystems*, 2019.
- [3] D. Reinsel, J. Gantz und J. Rydning, *The Digitization of the World - From Edge to Core*, IDC White Paper – US44413318, 2018.
- [4] I. Taleb, M. A. Serhani und R. Dssouli, “Big Data Quality: A Survey,” in *2018 IEEE International Congress on Big Data (BigData Congress)*, 2018, S. 166–173. DOI: 10.1109/BigDataCongress.2018.00029.
- [5] I. International Organization for Standardization, *Information technology - Vocabulary*, ISO 2382:2015. 2015, Letzter Zugriff: 20.5.2022 [Online]. Verfügbar:<https://www.iso.org/standard/63598.html>.
- [6] F. Horn, *A Practitioner’s Guide to Machine Learning*, Version 1.3, 02.02.2022. 2022, Letzter Zugriff: 22.05.2022 [Online]. Verfügbar:https://franziskahorn.de/mlbook_all.html.
- [7] A. Z. Faroukhi, I. El Alaoui, Y. Gahi und A. Amine, “Big Data Value Chain: A Unified Approach for Integrated Data Quality and Security,” in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 2020, S. 1–8. DOI: 10.1109/ICECOCS50124.2020.9314391.
- [8] M. Yalaoui und S. Boukhedouma, “A survey on data quality: principles, taxonomies and comparison of approaches,” in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, 2021, S. 1–9. DOI: 10.1109/ICISAT54145.2021.9678209.
- [9] C. Gröger, “There is no AI without data,” 2021. DOI: 10.1145/3448247.
- [10] J. Byabazaire, G. O’Hare und D. Delaney, “Data Quality and Trust : A Perception from Shared Data in IoT,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, S. 1–6. DOI: 10.1109/ICCWorkshops49005.2020.9145071.
- [11] L. Dominik, N. Tagalidou, K. Lingelbach und M. Spiekermann, *Ökosysteme für Daten und künstliche Intelligenz*, Positionspapier, 2019. DOI: DOI10.24406/ISST-N-543753.
- [12] HEG-KI und Europäische-Kommission, *Eine Definition der KI: Wichtigste Fähigkeiten und Wissenschaftsgebiete*, Brüssel, 2019.
- [13] Dathenethikkommission, C. Wendehorst und C. Woopen, *Gutachten der Datenethikkommission*, 2019.
- [14] HEG-KI und Europäische-Kommission, *Ethik-Leitlinien für eine Vertrauenswürdige KI*, Brüssel, 2019.

- [15] I. Döbel, M. Leis, M. M. Vogelsang u. a., “Maschinelles Lernen–Kompetenzen, Anwendungen und Forschungsbedarf,” *Fraunhofer IAIS, Fraunhofer IMW, Fraunhofer Zentrale. Zugriff am*, Jg. 21, S. 2020, 2018.
 - [16] OpenAI, I. Akkaya, M. Andrychowicz u. a., “Solving Rubik’s Cube with a Robot Hand,” *CoRR*, Jg. abs/1910.07113, 2019. arXiv: 1910.07113. Adresse: <http://arxiv.org/abs/1910.07113>.
 - [17] A. Kharwal, *What is Supervised Learning in Machine Learning*, 2020, Letzter Zugriff: 28.05.2022 [Online]. Verfügbar:<https://thecleverprogrammer.com/2020/10/23/what-is-supervised-learning-in-machine-learning/>.
 - [18] A. Ng, *Machine Learning Yearning - Technical Strateg for AI Engineers, In the Era of Deep Learning*, 2018.
 - [19] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller und W. Samek, “Analyzing Classifiers: Fisher Vectors and Deep Neural Networks,” Juni 2016, S. 2912–2920. DOI: 10.1109/CVPR.2016.318.
 - [20] A. B. Cremers und A. Englander, *Vertrauenswürdiger Einsatz von künstlicher Intelligenz*, 2019, Letzter Zugriff: 15.05.2022 [Online]. Verfügbar:https://www.iais.fraunhofer.de/content/dam/iais/KINRW/Whitepaper_KI-Zertifizierung.pdf.
 - [21] G. James, D. Witten, T. Hastie und R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, Bd. 112.
 - [22] L. Goasduff, *2 Megatrends Dominate the Gartner Hype Cycle for Artificial Intelligence, 2020*, 202, Letzter Zugriff: 25.05.2022 [Online]. Verfügbar:<https://www.gartner.com/smarterwithgartner/2-megatrends-dominate-the-gartner-hype-cycle-for-artificial-intelligence-2020>.
 - [23] B. Beckert, “Trustworthy artificial intelligence: Selected practical projects and reasons for the implementation deficit,” *TATuP - Zeitschrift für Technikfolgenabschätzung in Theorie und Praxis*, Jg. 30, Nr. 3, S. 17–22, 2021. DOI: 10.14512/tatup.30.3.17. Adresse: <https://www.tatup.de/index.php/tatup/article/view/6926>.
 - [24] S. Hallensleben und C. Hustedt, *From Principles to Practice An interdisciplinary framework to operationalise AI ethics*, Imprint, 2020.
 - [25] P. A, A. Jawaid, S. Dev und V. M S, “The Patterns that Don’t Exist : Study on the effects of psychological human biases in data analysis and decision making,” in *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, 2018. DOI: 10.1109/CSITSS.2018.8768554.
 - [26] Anonymous, *Incident Number 16. in McGregor, S. (ed.) Artificial Intelligence Incident Database*. Responsible AI Collaborative. 2015, Letzter Zugriff: 23.05.2022 [Online]. Verfügbar:incidentdatabase.ai/cite/16.
 - [27] J. Bronson und E. A. Carson, *Prisoners in 2017*, 2019, Letzter Zugriff: 17.05.2022. [Online]. Verfügbar: <https://bjs.ojp.gov/content/pub/pdf/p17.pdf>.
 - [28] S. R. Department, *Bevölkerung - Zahl der Einwohner in Deutschland nach Altersgruppen am 31. Dezember 2020*, 2022, Letzter Zugriff: 23.05.2022. [Online]. Verfügbar: <https://de.statista.com/statistik/daten/studie/1112579/umfrage/bevoelkerung-in-deutschland-nach-altersgruppen/>.
-

- [29] K.-A.-S. e. V., *Wahlverhalten nach Alter und Geschlecht*, 2021, Letzter Zugriff: 23.05.2022. [Online]. Verfügbar: <https://www.kas.de/documents/291186/291235/Datensammlung+Alter+und+Geschlecht.pdf/a25fb063-b305-d567-2d98-e77b1104f713?version=1.1&t=1618912553715>.
- [30] S. Bundesamt, *Datenreport 2021 - Kapitel 3: Bildung*, 2021, Letzter Zugriff: 23.05.2022. [Online]. Verfügbar: <https://www.destatis.de/DE/Service/Statistik-Campus/Datenreport/Downloads/datenreport-2021-kap-3.html>.
- [31] D. W. R. I. Washington, *Climate Watch Historical GHG Emissions*, 2022, Letzter Zugriff: 23.05.2022. [Online]. Verfügbar: <https://www.climatewatchdata.org/ghg-emissions>.