



# **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 versichern 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 . . . . .	VI
Tabellenverzeichnis . . . . .	VII
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Zielsetzung . . . . .	3
1.3 Aufbau der Arbeit . . . . .	4
<b>2 Stand der Technik</b>	<b>5</b>
2.1 Daten als wertschöpfende Ressource . . . . .	5
2.1.1 Daten . . . . .	5
2.1.2 Datenqualität . . . . .	6
2.1.3 Bias . . . . .	6
2.2 Künstliche Intelligenz . . . . .	7
2.2.1 Künstliche Intelligenz Allgemein . . . . .	7
2.2.2 Teilgebiet maschinelles Lernen . . . . .	7
2.2.3 Ethik in der künstlichen Intelligenz . . . . .	7
2.3 Bias im Zusammenhang mit Künstliche Intelligenz (KI) . . . . .	8
2.3.1 Diskriminierung durch Bias in Daten . . . . .	8
2.3.2 Gegenmaßnahmen . . . . .	8
<b>3 Praktischer Teil</b>	<b>9</b>
3.1 Szenarien . . . . .	9
3.1.1 Szenario 1 . . . . .	9

3.1.2	Szenario 2 . . . . .	10
3.2	Konzeption . . . . .	11
3.2.1	Grobkonzept . . . . .	11
3.2.2	Feinkonzept . . . . .	12
3.3	Umsetzung . . . . .	15
3.4	Datenauswertung . . . . .	26
3.5	Evaluation der Ergebnisse . . . . .	26
<b>4</b>	<b>Schluss</b>	<b>27</b>
4.1	Zusammenfassung . . . . .	27
4.2	Diskussion . . . . .	27
4.3	Ausblick . . . . .	27

# Abkürzungsverzeichnis

<b>KI</b>	Künstliche Intelligenz
<b>ML</b>	Machine Learning
<b>M</b>	Männlich
<b>W</b>	Weiblich

# Abbildungsverzeichnis

2.1	Weltweit jährlich anfallende Datenmenge [2] . . . . .	5
3.1	Programmablaufplan der fünf Hauptschritte zur Generierung der Daten . .	12
3.2	Verbindungen zwischen den Attributen des zweiten Szenario . . . . .	13
3.3	Erste Zelle Code des Szenario1 . . . . .	16
3.4	Codezeile zur Bestimmung des Geschlechts einer Person . . . . .	17
3.5	Codezeilen zum Erstellen eines Dictionary mit den zur Bewertung relevanten Attributen . . . . .	19
3.6	Programmablaufplan zur Generierung der Regeln von Szenario 1 . . . . .	20
3.7	Methode zur Initialisierung eines Bewertenden . . . . .	21
3.8	Methode eines Bewertenden zum Bewerten von Anträgen. . . . .	22
3.9	Dictionaries gefüllt mit Standard Werten des Szenario 1 . . . . .	24
3.10	Letzte Zelle des Szenario 1 für die Benutzenden Interaktion . . . . .	24

# Tabellenverzeichnis

3.1	Tabelle für die Auswirkung der Attributen von Szenario 1 . . . . .	9
3.2	Tabelle der Attribute und Auswirkungen von Szenario 2 . . . . .	10
3.3	Tabelle zur Bestimmung der Wahrscheinlichkeiten für das Geschlecht . . .	17
3.4	Tabelle der Wahrscheinlichkeiten für die Härte der Strafe nach Geschlecht .	17
3.5	Tabelle zur Bestimmung der Wahrscheinlichkeiten für die Hautfarbe unter Berücksichtigung des Geschlechts . . . . .	18



# 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 gesetzt und der Einfluss eines Bias auf eine KI betrachtet und 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“. Hiermit ist gemeint, dass Daten in ihre Rohform nicht sonderlich wertvoll sind, jedoch sobald man beginnt sie zu verarbeiten gewinnen sie an Wert. Lange Zeit waren Daten nur ein Nebenprodukt der Digitalisierung. Daten wurde gesammelt und gespeichert, aber nicht weiter verwendet. Insbesondere durch das rasante Aufkommen des Internet of Things, steigt die Menge der gesammelten Daten exponentiell an.[1]

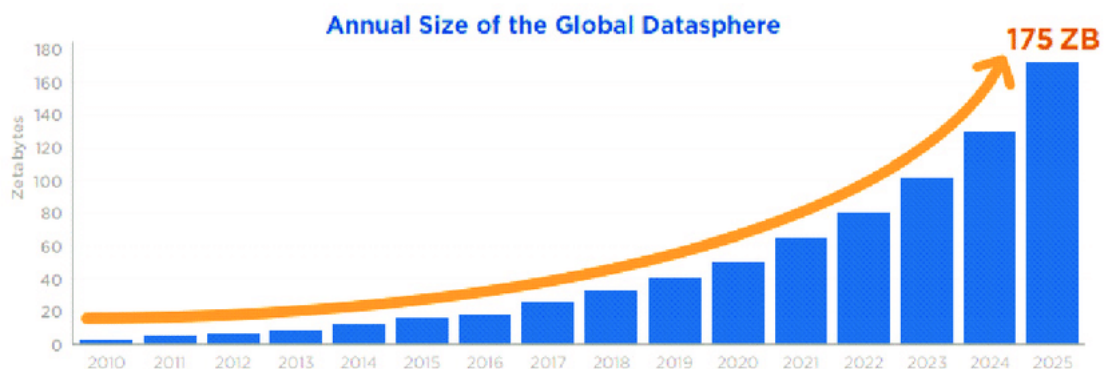


Abbildung 2.1: Weltweit jährlich anfallende Datenmenge [2]

Abbildung 2.1 stammt aus dem Jahr 2018 und verdeutlicht, dass bereits damals erwartet wurde, dass bis im Jahr rund 175 Zetabyte an Daten jährlich generiert und gesammelt werden. Im Vergleich dazu waren es 2018 gerade einmal 33 Zetabyte an Daten, die weltweit generiert wurden. [2]

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“. [3] Daten gibt es in unterschiedlicher Form. Meist unterscheidet man zwischen strukturierten und unstrukturierten Daten.

- Daten haben sich von einem Nebenprodukt zu einer wertvollen Ressource entwickelt
- Daten sind das Öl der Zukunft <- Zitat raussuchen - Wir sammeln enorm viel Daten
- > Big Data -> zig tausende an Zetabyte <- Quelle suchen - Daten ermöglichen neue Geschäftsfelder -> Data Driven Diagnoses, Development, Analytics

### 2.1.2 Datenqualität

- Datenqualität spielt eine große Rolle seit wir Daten anfangen analysieren auszuwerten - 80 Prozent der Arbeit eines Data Scientist ist das Daten vorverarbeiten aufgrund schlechter Datenqualität - Data Quality Management - Data Quality Richtlinien und Anforderungen
- Metadaten -> Das Wissen über die Daten, was sollten die Daten repräsentieren und tun sie das auch

### 2.1.3 Bias

- Begriffserklärung: Data Bias vs Bias Verzerrung / Over-/Underfitting (zu viel/zu wenig lernen im ml)
- 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 [4]

## **2.2 Künstliche Intelligenz**

### **2.2.1 Künstliche Intelligenz Allgemein**

- Was bildet KI alles ab <- Grafik B. Otto? - KI als Geschäftsfeld - KI und seine potentiale Ausleuchten - Was ist alles KI - Wie definiert sich KI

### **2.2.2 Teilgebiet maschinelles Lernen**

- Wie funktioniert ML -> Daten die gesammelt werden - Bewertete Daten aus der Vergangenheit -> Daten bewerteter Jobs <- Quelle - Welche Risiken - over under fitting etc.  
- Superviced learning -> Data Bias  
- Unsuperviced learning -> nicht zwingend Data Bias

### **2.2.3 Ethik in der künstlichen Intelligenz**

- Ethik spielt eine große Rolle in der Gesellschaft - EU Setzt sich mit Richtlinien auseinander - Diskriminierung durch KI ist ein fatales Problem - Maschinen besitzen keine Moral und Ethisches werteverständnis

## **2.3 Bias im Zusammenhang mit KI**

### **2.3.1 Diskriminierung durch Bias in Daten**

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

### **2.3.2 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.

### 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 Jahre 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 außen vor gelassen wurde. Die Hautfarbe der Person wird als „Schwarz“ oder „Weiß“ festgehalten. Die noch laufende Strafe des Gefangenen wird in Jahren von 1-5 angegeben. Da hier definiert wird ein Bewährungsantrag kann 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 dem Bewerter werden folgende Regeln definiert:

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

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 sind die möglichen Werte, welche durch eine Verzerrung und damit einem menschlichem Vorurteil eines Bewerter 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 ein Bewerter dann anhand dieser beiden Werte eine Tendenz erhalten und dann 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 Ausprä-

gungen 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 Bewerter 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 ein Bewerter 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 darauf in 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 werden daraufhin 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 im folgenden 3. Schritt die Regeln nach welchen bewertet wird aufgestellt werden. Dafür soll zuerst die in den Parametern gewünschte Anzahl an Bewertenden er-

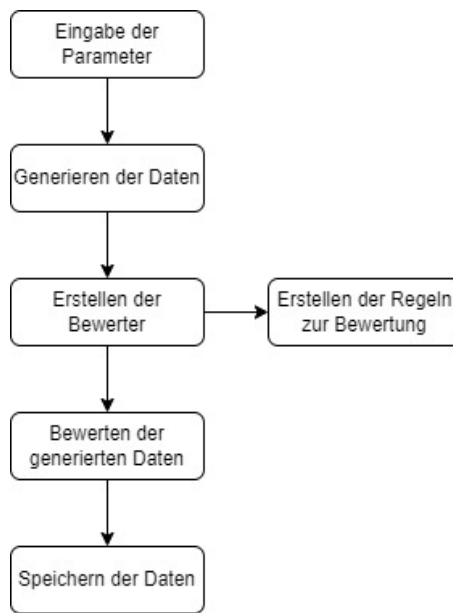


Abbildung 3.1: Programmablaufplan der fünf Hauptschritte zur Generierung der Daten

stellt 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 soll dem Geschlecht entsprechend angepasst werden. So haben zum Beispiel weibliche Personen eine 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 wie zu sehen 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

einer Auswirkung des Sozialen gerechnet. Durch das Soziale und die Vorverbindung wird die Politische Orientierung und das Alter ebenfalls indirekt darauf mit ein Bezogen. Dies ist hier der Fall, da die Berechnung CO<sub>2</sub>-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 für beide Szenarien müssen dann später bei der Umsetzung die Wahrscheinlichkeiten der Ausprägungen der Attribute, wo es möglich ist, durch Statistiken bestimmt werden 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 diskriminieren sollte, hat dieses wie oben erläutert in seinen negativen Regeln die gewünschte Ausprägung 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**

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 zu aller erst in der ersten Zelle die benötigten Bibliotheken geladen werden.

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

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

fake = Faker()
```




Abbildung 3.3: Erste Zelle Code des Szenario1

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

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

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 verwen-



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

det.[5, 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.

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

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 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.[5, 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 ethischen Gruppen aufgeteilt. 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 am 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 Attribute 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. Eine Dictionary für die Regeln der positiven Auswirkung, eines für die negative Auswirkung und nochmals die selben zwei Listen nochmals ergänzt durch die gewünschte Verzerrung.

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.

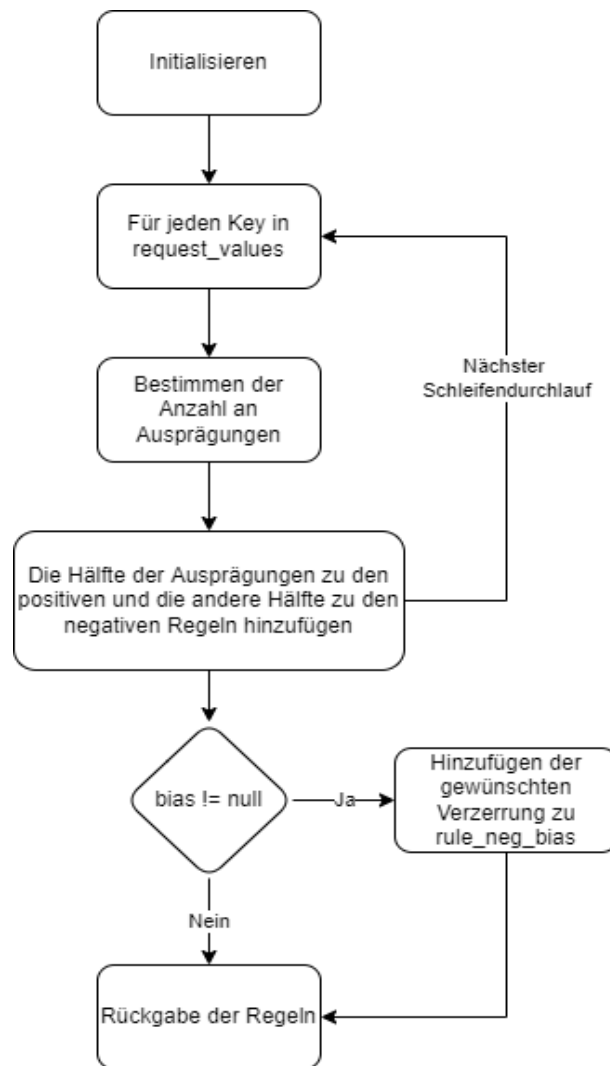


Abbildung 3.6: Programmablaufplan zur Generierung der Regeln von Szenario 1

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 nach den Regeln 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 dann beruhend auf diesen Wahrscheinlichkeiten die Entscheidung zu treffen und das Ergebnis im Antrag also dem Dataframe anzuhängen. Zum Schluss wird dann 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 gewünschte 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“ Objekte 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



```
#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

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.

```
#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

In der Abbildung 3.10 ist die letzte Zelle für die Benutzenden Interaktion 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.

### **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] B. Otto, D. Lis, J. Jürjes u. a., *Data Ecosystems*, 2019.
- [2] D. Reinsel, J. Gantz und J. Rydning, *The Digitization of the World - From Edge to Core*, IDC White Paper – US44413318, 2018.
- [3] I. International Organization for Standardization, *Information technology - Vocabulary*, ISO 2382:2015. 2015, Letzer Zugriff: 20.5.2022 [Online]. Verfügbar: <https://www.iso.org/standard/63598.html>.
- [4] 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.
- [5] 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>.