

## Projektbericht

# Gruppe 'Neugedacht' - Matching Spender und Spendenorganisationen

Projektseminar „Informatik und Optimierung für Nachhaltigkeit“

Marvin Henn

Christian Koch

Nicolas Steigerwald

Niklas Vogt

Semester: Sommersemester 2023

Abgabedatum: 30. Juli 2023

Betreuer: Prof. Dr. Marie Schmidt



Julius-Maximilians-Universität Würzburg  
Lehrstuhl für Informatik I  
Algorithmen und Komplexität

# Inhaltsverzeichnis

<b>1. Abstract.....</b>	<b>2</b>
<b>2. Einleitung.....</b>	<b>3</b>
2.1 Projektgeber.....	3
2.2 Projektziele.....	3
2.2.1 Fragebogen.....	4
2.2.2 Filter-Algorithmus.....	4
2.2.3 Code.....	4
<b>3. Projektarbeit.....</b>	<b>5</b>
3.1 Koordination in der Gruppe.....	5
3.2 Kontakt und Interaktion mit den Projektstellern.....	5
3.3 Zeitlicher Ablauf.....	6
3.3.1 Beginn bis zur ersten Zwischenpräsentation.....	6
3.3.2 Bis zur zweiten Zwischenpräsentation.....	7
3.3.3 Bis zur Abschlusspräsentation.....	7
3.4 Probleme im Verlauf des Projektes und Lösungsansätze.....	8
3.4.1 Fragenfindung.....	8
3.4.2 Ausarbeiten der Kategorien.....	9
3.4.3 Scoring-Algorithmus und Varianten.....	9
3.4.3.1 Verworfenen Ideen.....	10
3.4.4 Lokalität.....	11
3.4.4.1 Postleitzahl-Vergleich.....	11
3.4.4.2 Distanz in Kilometer.....	13
3.4.4.3 Weitere Probleme.....	14
3.5 Beiträge der Gruppenmitglieder.....	14
<b>4. Projektergebnisse und Diskussion.....</b>	<b>16</b>
4.1 Einordnung.....	16
4.2 Vorgehen.....	16
4.2.1 Ideen.....	16
4.2.2 Testumgebung.....	16
4.2.3 Entwicklung des Skripts.....	17
4.3 Resultate.....	17
4.4 Diskussion.....	18
4.5 Zukunft.....	19
<b>5. Anhang.....</b>	<b>21</b>

# 1. Abstract

Dieser Projektbericht dient der Darstellung des Projekts mit "neugedacht" als Teil des Projektseminars Informatik und Optimierung für Nachhaltigkeit.

Im Rahmen dieses Seminars kamen an einem Informationstag vier Firmen an die Universität und stellten den anwesenden Studenten ihre Probleme vor, zum gegebenen Thema, auf welches sich im Anschluss beworben werden konnte.

In unserem Projekt wurde für die neugedacht gGmbH ein Pythonskript entwickelt, welches für Antworten auf einen von uns festgelegten Katalog an Fragen möglichst passende Organisationen aus ihrer Datenbank findet und diese an die Hauptanwendung zurückgibt.

Im Folgenden soll dargestellt werden, wie wir zu unserem Ergebnis gekommen sind.

## 2. Einleitung

### 2.1 Projektgeber

Projektgebend war das junge Unternehmen "neugedacht". Vor kurzem hat die Firma die Gemeinnützigkeit erreicht und versucht durch ihre Lösung den Prozess der Unternehmensspende zu demokratisieren.

Zunächst werden mögliche gemeinnützige Organisationen recherchiert und, falls sie die Transparenz- und Ethik-Anforderungen erfüllen, in das Portfolio aufgenommen. Dabei sind die Gründer auch auf Probleme wie z.B. einen Missionierungsauftrag oder etwaige Skandale eingegangen, um für ihre Kunden nur sichere Spendenziele anzubieten.

Weiter wurden nun Kunden gesucht, die mithilfe der zum Zeitpunkt des Projekts noch nicht fertiggestellten Webanwendung ihren Mitarbeitenden die Möglichkeit bieten wollen, am Prozess der Unternehmensspende teilzuhaben. Dafür werden die Unternehmen eine Übersicht bekommen, um das Ergebnis der Spendenverteilung und den Impact dieser Messen darstellen zu können.

Die Mitarbeitenden bekommen Zugang zu einer Seite, auf welcher sie ihr präferiertes Spendenziel auswählen können, um ihren Anteil an der Gesamtspende zu verteilen. Außerdem können die Mitarbeitenden ebenfalls ihre bisher verteilte Spendensumme, sowie den Impact dieser einsehen.

Die Neugedacht gGmbH bekommt hierbei einen Teil von den über ihre Plattform verteilten Spenden und hilft gleichzeitig dabei, dass sich die Mitarbeitenden in einen weiteren Prozess des Unternehmens eingebunden fühlen.

### 2.2 Projektziele

Es ist aufgefallen, dass Mitarbeitende, die nach einem geeigneten Spendenziel auf der Plattform suchen, leicht überfordert werden, von der Menge an möglichen Organisationen. Vor allem wenn die Vision der Gründer aufgeht und es möglich ist, an hunderte verschiedene Organisationen weltweit zu spenden, wirkt die Menge an Möglichkeiten auf unentschlossene Mitarbeitende schnell erschlagend.

Dies soll durch einen Fragebogen an den Mitarbeitenden gelöst werden, in dem allgemeine Fragen zu Präferenzen an eine gemeinnützige Organisation gestellt werden. Im Anschluss soll auf Basis dieser Fragen eine deutlich kleinere Menge an passenden Optionen zur Auswahl gestellt werden.

Unser Projekt umfasst die Erstellung des Fragebogens, die Definition des Prozesses wie aus den Antworten des Mitarbeitenden die vorgeschlagenen Optionen gefiltert werden, sowie die Erstellung einer Software für das Backend, welche diesen Prozess abbildet.

### 2.2.1 Fragebogen

Der Fragebogen obliegt einem offensichtlichen Zielkonflikt. So würden möglichst viele Detailfragen voraussichtlich zu einem besseren Ergebnis führen, gleichzeitig aber auch den Prozess deutlich verlängern und Frustration auslösen.

Daraus leitet sich als Ziel für den formulierten Fragebogen mit möglichst wenigen, dafür gut durchdachten Fragen ein qualitativ hochwertiges Ergebnis zu erzielen.

### 2.2.2 Filter-Algorithmus

Das reine "harte" Filtern nach Eigenschaften ist zu restriktiv, da so potentielle Organisationen, die nah am gewünschten Ergebnis sind, aussortiert werden oder möglicherweise gar kein Ergebnis zustande kommt. Daher muss eine Lösung gefunden werden, mit welcher wir Antworten aus dem Fragebogen verwerten und ohne zu restriktiv zu handeln Ergebnisse finden, die möglichst nah an dem aus den Antworten ersichtlichen Ideal sind.

### 2.2.3 Code

Die Live-Anwendung von neugedacht befand sich zu Beginn des Projekts noch kurz vor dem Entwicklungsstart. So können wir leider nicht damit rechnen, das Front-End für den in unserem Projekt betrachteten Prozess zu bauen.

Daher muss unsere Softwarelösung unabhängig von etwaigen Designentscheidungen der Hauptlösung möglichst gut in diese integrierbar und leicht anpassbar sein.

## 3. Projektarbeit

### 3.1 Koordination in der Gruppe

Während des ersten festgelegten Seminartermins gemeinsam mit Moritz von neugedacht umrissen wir nicht nur grob den Plan für den Rest des Projekts, sondern machten bereits einen festen wöchentlichen Termin aus.

So trafen wir uns, bis auf wenige Ausnahmen, jeden Freitag um 8:15 Uhr auf Zoom, um den Fortschritt der letzten Woche zu besprechen, Aufgaben für die nächste Woche zu planen bzw. zu verteilen und wichtige Grundsatzentscheidungen gemeinsam zu treffen. Bei diesen Treffen war auch Moritz beinahe jedes Mal dabei. Außerdem schloss sich Frau Schmidt diesem einmal an, um unseren Workflow zu beobachten, sowie Input zu gewissen Themen zu geben. Durch die regelmäßigen festen Termine konnten wir sicherstellen, dass das Projekt kontinuierlich voranschreitet und gemeinsam die Ergebnisse der letzten Wochen diskutiert werden. Dieser feste Termin hatte den Charakter eines Huddle Meetings und stellte sich als sehr effektiv für die Verbesserung der Zusammenarbeit heraus.

Weiter standen wir in regem Austausch in einer WhatsApp-Gruppe, in welcher auch Moritz dabei war. So konnten wir, auch wenn wir gerade nicht im Meeting waren, schnell Fragen klären, sowie gemeinsam Richtungsentscheidungen "on the fly" treffen.

Dadurch, dass wir einen großen Teil der Aufgaben auf einzelne Mitglieder oder Zweierteams aufgeteilt haben, konnten die einzelnen Teammitglieder ihre Zeiten deutlich freier einteilen. Wenn an einzelnen Aufgaben mehrere Teammitglieder beteiligt waren, wurden diese meist in einzelnen kleinen Zoom-Meetings organisiert.

Auch ist zu erwähnen, dass alle Entscheidungen als Team gemeinsam getroffen wurden und wir uns aktiv dagegen entschieden haben, einen festen Team Lead auszumachen. Diese Mentalität und teaminterne Demokratie hat uns jedoch selten ausgebremst und eher das Engagement und die selbstständige Arbeitsweise gefördert.

### 3.2 Kontakt und Interaktion mit den Projektstellern

Uns war von vornherein wichtig, dass wir möglichst nah mit den Projektgebern zusammenarbeiten, damit wir sicherstellen können, nicht am Ziel von Neugedacht vorbei zu arbeiten und das bestmögliche Ergebnis liefern zu können.

So war einer der Gründer, wie bereits dargelegt, jede Woche im Meeting dabei und in unserer gemeinsamen WhatsApp-Gruppe. Der zweite Mitgründer von Neugedacht kam ebenfalls bei einem der Meetings dazu, um seinen Input zu liefern.

Von diesem engen direkten Austausch konnten wir sehr profitieren.

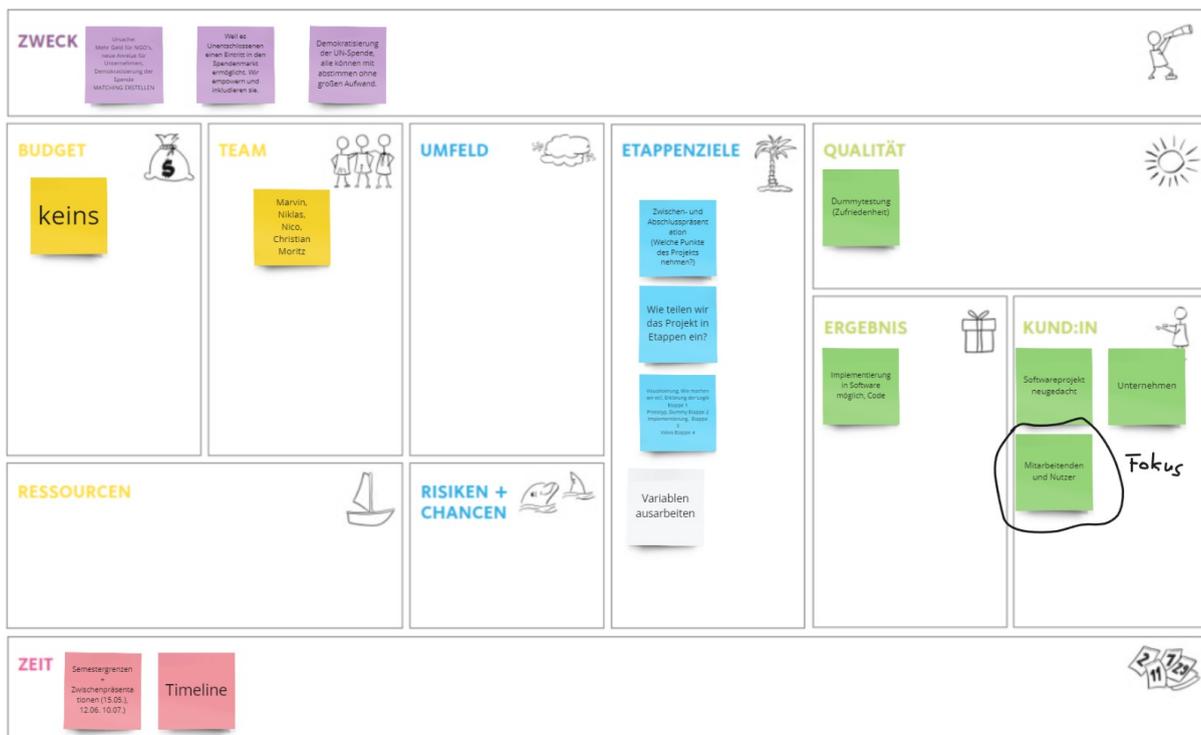
### 3.3 Zeitlicher Ablauf

Im Folgenden soll der zeitliche Verlauf des Projekts zwischen den Präsentationen als Meilensteine dargestellt werden.

#### 3.3.1 Beginn bis zur ersten Zwischenpräsentation

Vom Beginn des Semesters bis zur ersten Zwischenpräsentation am 15.05. machten wir uns viele Gedanken darüber, wie die Fragen auszusehen hatten, wie wir alles in Code umsetzen konnten, was wir auf jeden Fall erreichen wollten und was wir, falls noch Zeit da wäre, ergänzen könnten.

Am 24.04. nutzten wir unser erstes Treffen für ein Kennenlernen der Projektgruppe mit Moritz dem Projektsteller. Wir füllten den Projekt-Canvas aus und umrissen so den Scope unseres Projekts.



Wir einigten uns darauf, dass wir den Fragenkatalog, einen Filteralgorithmus und ein Software-Backend als Mindestziel festhalten. Weiter wurde der Vorschlag eines Scoring-Algorithmus gemacht, dieser erklärt und diskutiert. Bis zur nächsten Woche sollte dieser besser ausgearbeitet werden. Weiter sollten mögliche Fragen überlegt und vorgeschlagen werden. Außerdem hielten wir den Freitag 8:15 Uhr als festen Termin für unser wöchentliches Meeting fest.

Beim zweiten Treffen am 05.05. wurde der nun weiter verfeinerte Scoring-Algorithmus vorgestellt. Dabei fiel auf, dass wir keine Möglichkeit haben, diesen sinnvoll zu testen. Also sollte bis zur nächsten Woche eine Testumgebung gebaut werden, mit welcher wir in der Lage sind, die Sinnhaftigkeit unserer Lösung zu testen. .

Die Vorgeschlagenen Fragen sahen eine Klassifizierung nach Kategorien vor. Diese sollten auch anders behandelt werden als die anderen Fragen und tatsächlich "hart" gefiltert werden. Weiter kam es zu einer längeren Diskussion darüber, welche Kategorien sinnvoll wären und so sollten diese bis nächste Woche definiert werden. Weiterhin waren die Fragen nicht final und eine weitere Verfeinerung sollte bis zur nächsten Woche geschehen.

Beim 3. Meeting am 12.05. war der Algorithmus fertig definiert und wurde angenommen. Die Testumgebung sah sehr vielversprechend aus und sollte bis zur Zwischenpräsentation mit dem gemeinsam gesammelten Feedback vorgestellt werden. Ebenfalls war der Fragenkatalog vorerst fertig und eine Kategorisierung, die alle NGOs abdeckt, angenommen.

Diese Ergebnisse stellten wir bei der Zwischenpräsentation am 15.05. vor.

### 3.3.2 Bis zur zweiten Zwischenpräsentation

Zwischen der ersten Präsentation und unserem vierten Treffen am 26.05. gab es zusammen mit Moritz ein Meeting mit einer Psychologiestudentin namens Anne Mächel. Zunächst wurde ihr die Thematik ausführlich geschildert und der bereits ausgearbeitete Fragenkatalog vorgestellt. Sie fand den ausgewerteten Fragenkatalog gelungen und betonte, dass die Reihenfolge der Fragen wichtig sei. Weiter würde sie nichts ergänzen.

Beim nächsten Meeting einigten wir uns auf einen Aufbau der Back-End Lösung und ihren vorläufigen Umfang. .

Der Code, der bis dahin leider noch nicht ganz fertig war, sowie die Ergebnisse aus dem Meeting mit Frau Mächel wurden bei der Zwischenpräsentation am 12.06. vorgestellt.

### 3.3.3 Bis zur Abschlusspräsentation

Bis zur Abschlusspräsentation am 10.07. war unser Ziel, den Code voll und ganz fertigzustellen und ein Konzept für das Video, welches ebenfalls gedreht, geschnitten und bis zum 06.08. abgegeben werden musste, zu erstellen.

Beim sechsten Treffen am 16.06. besprachen wir das weitere Vorgehen. Der Code muss noch aufgeräumt und optimiert werden. Wir müssen definieren, wann das Projekt als abgeschlossen gilt. Außerdem müssen wir uns überlegen, ob die Fragen wirklich so final abgegeben werden können. Außerdem diskutierten wir einige Vorschläge für das Video und begannen mit den Planungen für den Projektbericht..

Am 23.06 beim letzten Meeting vor der Abschlusspräsentation schaltete sich Frau Schmidt zu unserem Meeting dazu. Diese Möglichkeit nutzten wir, um wichtige Fragen bezüglich der Finalisierung unseres Projekts sowie der Abschlusspräsentation zu klären. Außerdem stellten wir ihr unseren Entwurf für das Projektvideo vor. Den von ihr gesammelten Input nutzten wir, um den Code und die Präsentation fertigzustellen.

Bei der Abschlusspräsentation am 10.7, stellten wir den fertigen Code vor. Weiterhin hatten wir bis dahin auch das Problem der präzisen Distanzbestimmung mit Hilfe von Postleitzahlen gelöst und konnten dies ebenfalls präsentieren. Damit konnte unser Code alle von uns zu Beginn aufgestellten Anforderungen lösen.

## 3.4 Probleme im Verlauf des Projektes und Lösungsansätze

Im Folgenden sollen die einzelnen größeren Probleme, auf welche wir gestoßen sind, atomar betrachtet werden, sowie dargelegt werden, wie wir zu den Lösungen dieser gekommen sind.

### 3.4.1 Fragenfindung

Das Wichtigste an einem Fragenkatalog sind in erster Linie selbstverständlich die Fragen, die man abfragt. Aber welche Fragen sind sinnvoll, was sollte man eher nicht fragen?

Was ist erst-, was zweitrangig? Wie viele Fragen sind zu viel? Wie sind diese in den Code implementierbar?

Anfangs überlegten wir uns, was Eigenschaften von NGOs sind, die alle Organisationen besitzen und einfach zu prüfen sind, um eine Sortierung anhand eines Algorithmus einfach handhaben zu können.

Als Beispiel wäre das die Größe einer NGO, woran diese festgelegt wird, haben wir variabel festgelegt.

Ein Negativbeispiel für unseren Zweck ist der Wahl-O-Mat, da er viel zu viele Fragen beinhaltet und schnell eintönig wird, deswegen haben wir uns von Anfang an eine Obergrenze von 12 Fragen gesetzt.

Die Definition sinnvoller Fragen forderte mehrere Wochen und viel Diskussion, was auch durch die ungenaue Vorstellung vom Endergebnis hervorgerufen wurde.

Letzten Endes kamen wir auf folgende Fragen in dieser Reihenfolge:

Wie groß soll die NGO sein?

Bei dieser wird mit klein, mittel oder groß geantwortet.

In welcher Region soll die NGO aktiv sein?

Bei dieser Frage hat man regional, national und international zur Auswahl, wobei man bei regional eine Postleitzahl angeben muss und bei international ein entsprechendes Land angeben kann, sofern man das möchte.

Soll die NGO einen Religionsbezug haben?

Hier ist eine Liste von den gängigsten Religionen gegeben, Buddhismus, Christentum, Hinduismus, Islam, Judentum, bei denen man einzeln mit ja, nein oder egal antworten kann.

Welche Verwaltungsrat soll die NGO haben?

Soll die NGO die Spenden selbst nutzen oder nur weitergeben, oder ist das egal?

Welche Kategorien interessieren dich?

Hier fragen wir ebenfalls von uns erstellte Kategorien ab, um die NGOs final zu filtern und anschließend ausgeben zu können.

Diese Fragen besprachen wir in einem Gespräch mit einer Psychologiestudentin, die keine

weiteren Ergänzungen zu machen hatte und die Fragen als sehr präzise und gut gewählt empfand.

### 3.4.2 Ausarbeiten der Kategorien

Alle NGOs in Kategorien einzuteilen, die nicht zu allgemein, aber auch nicht zu spezifisch sind, sind keine allzu leichte Aufgabe.

Bei zu wenigen Kategorien ist die Gefahr, dass sehr unterschiedliche NGOs in derselben Kategorie landen zu hoch, bei zu vielen ist es schwierig, diese vernünftig in Code umzusetzen und wenn einzelne NGOs Sonderkategorien haben, wird auch der Sinn der Kategorisierung genommen.

Niklas sah sich hunderte deutsche NGOs an und lieferte beim zweiten Treffen am 05.05. sechs große Kategorien ab, welche waren: Tierwohl, Soziales, Kinder und Ältere, Bildung und Wissenschaft, Umwelt und Religion.

Die Kategorie "Soziales" war noch zu allgemein gehalten, deshalb wurde diese in "Hilfe für Bedürftige" umbenannt, um "Umwelt" und "Tierwohl" stärker in sich abzugrenzen.

Natürlich hätte man diese noch um einiges weiter aufspalten können, dabei wäre das unseres Erachtens nach aber zu spezifisch geworden.

Außerdem stand die Frage im Raum, ob das nicht doch zu allgemein gehalten sei.

Bis zum vierten Treffen am 26.05. wurden die Kategorien nochmal überarbeitet.

Tierschutz, Hilfe für Bedürftige, Gesundheitswesen, Kinder- und Jugend, Senioren, Umwelt, Bildung, Wissenschaft und Religion waren die, mit denen alle zufrieden waren und laut Moritz obendrein alles abgedeckt wurde.

### 3.4.3 Scoring-Algorithmus und Varianten

Um dem Nutzer bei seiner Spenden-Entscheidung zu helfen, wollen wir ihm passende NGOs präsentieren. Dafür müssen wir dem Nutzer eine Auswahl bieten und die vorhandenen NGOs eingrenzen, also nicht passende ausschließen. Dafür haben wir uns für einen Scoring-Ansatz entschieden. Hierbei wird jeder NGO ein Wert zugewiesen, der, sofern die Antworten des Arbeitnehmers mit den Eigenschaften der NGO übereinstimmen, erhöht wird. Die Idee ist, dem Nutzer dann die NGOs mit möglichst hohem Score vorzuschlagen, da diese am besten zu ihm passen.

Um einzelne Eigenschaften der NGOs überprüfen zu können, legten wir für jede fünf Attribute in Datenbanken fest. Die Eigenschaften sind Größe, Postleitzahl, Land, Verwaltungsart und ein Verweis auf eine Datenbank mit den von uns festgelegten Religionsarten, sowie ein Verweis auf eine Datenbank, mit den von uns ausgedachten Kategorien und welche die entsprechende NGO beinhaltet.

Die Frage, die sich für das Scoring stellt, ist also, wie aus einem ausgefüllten Fragebogen und einer NGO aus der Datenbank ein Score berechnet wird. Diese Bewertung wird dann für alle NGOs durchgeführt, wodurch man die am höchsten bewerteten NGOs dem Nutzer als Vorschlag zurückgeben kann. Uns war wichtig, nicht nur "die beste" NGO mit dem höchsten Score anzuzeigen, sondern eben eine Auswahl. Dies ermöglicht dem Nutzer, selbst noch eine Entscheidung zu treffen, ohne durch unsere Auswahl zu stark beschränkt zu werden.

Hierbei haben wir eigene Erfahrungen, z.B. beim Online-Kauf von Handys oder Laptops, genutzt. Bei den meisten Vergleichsseiten kann man die gewünschten Geräte ebenfalls nach bestimmten Eigenschaften filtern, wie Preis, Speicherplatz oder Displaygröße. Es kommt aber oft vor, dass es kein Modell mit allen gewünschten Eigenschaften gibt, oder dass die Eigenschaften nicht einschränkend genug sind, und die Auswahl noch zu groß ist. Mit unserem Ansatz wollen wir sicherstellen, dass ein Nutzer immer eine Auswahl von uns erhält, egal wie er den Fragebogen beantwortet hat.

Unser Scoring ist additiv und modular, das heißt, es berechnet zu jeder einzelnen Eigenschaft unabhängig voneinander einen Score und rechnet ihn auf einen gesamten NGO-Score zusammen.

Im Prinzip wird verglichen, wie gut eine Antwort aus dem Fragenkatalog des Nutzers zu einer NGO passt. Als Beispiel betrachten wir die NGO-Größe: Will der Nutzer eine "große" NGO, und hat die zu bewertende NGO auch die Eigenschaft "groß", dann ergibt sich hieraus ein Score von 1. Derselbe Score entsteht bei der Paarung "mittel-mittel" oder "klein-klein". Das bedeutet, wenn die Größe einer NGO mit der gewünschten Größe des Nutzers übereinstimmt, geht das ins gesamte Scoring mit +1 ein. Wenn die beiden Größen nicht übereinstimmen, wird berechnet, wie weit sie voneinander entfernt liegen. So führt die Nutzer Antwort "klein" bei einer "mittleren" NGO zur Bewertung  $+1/2$ . Liegen die Antworten noch weiter auseinander, wie bei der Paarung "klein-groß", so wird dieser Fall mit 0 bewertet. Dies steht dafür, dass die NGO im Bezug auf das Größe-Attribut nicht zu den Benutzereingaben passt.

Im Code ist jedes Scoring für jeden Vergleichswert einzeln implementiert, sodass Änderungen, z.B. nur am "Größe-Scoring" leicht vorgenommen werden können. Jedes Scoring liefert dabei Werte von 0 bis 1. Will man eine Eigenschaft höher gewichten, so müsste man das im Code ändern. Wenn man die Lokalität stärker gewichten will, müsste man das Scoring dafür ändern, sodass nahe NGOs nicht mehr nur mit +1, sondern mit +2 oder höher bewertet werden. Wir haben uns dafür entschieden, jedes einzelne Scoring von 0 bis 1 zu bewerten und keine zusätzliche Gewichtung vorzunehmen.

#### 3.4.3.1 Verworfenen Ideen

Es gab anfangs die Idee, die Kategorien einer NGO zusätzlich mit einer Bewertung abzuspeichern. So hätte eine NGO nicht mehr nur die Kategorie "Umweltschutz", sondern diese wäre zusätzlich mit einem Score von 0 bis 1 abgespeichert. Die Idee dahinter war zu bewerten, wie "stark" sich eine NGO in einem bestimmten Bereich engagiert, wobei 1 für "engagiert sich stark" und 0 für "engagiert sich gar nicht" steht. Ergänzend hätte man den Nutzer gefragt, wie wichtig ihm die Kategorie "Umweltschutz" auf einer Skala von 0 bis 1 ist.

Aus diesen Werten hätte folglich für jede Kategorie einer NGO ein Scoring berechnet werden können.

Die Bewertungen "Nutzer will 0.5 Umweltschutz" und "NGO hat 0.7 Umweltschutz" könnte man dann zu einem Score von  $0.5 * 0.7 = 0.35$  multiplizieren und auf den gesamten Score

addieren. Eine Paarung "Nutzer will 1.0" und "NGO hat 1.0" würde also zu einem Teil-Score von 1 führen, die Paarung "0-0" zu einem Teil-Score von 0.

Das Verlockende an diesem Ansatz ist, dass man leicht Berechnungen zwischen Benutzereingaben und abgespeicherten NGOs durchführen kann.

Weil man damit mehr Eigenschaften einer NGO miteinbezieht, würden zwischen den Scores von unterschiedlichen NGOs größere Differenzen entstehen, was wiederum zu einer höheren Aussagekraft führt.

Dieser Ansatz hat aber entscheidende Nachteile, weswegen wir uns schlussendlich dagegen entschieden haben. Es stellt sich die Frage, woher die Bewertungen der Kategorien einer NGO kommen. Diese müssten festgelegt werden, um sie in der Datenbank zu speichern. Wenn eine NGO das selbst eintragen kann, besteht die Gefahr einer eigenen "Überbewertung", um bessere Scorings bei den Nutzern zu erzielen, denen entsprechende Kategorien besonders wichtig sind. Wenn neugedacht diese Bewertung selbst übernimmt, würde für jede neue NGO, die in das Portfolio aufgenommen wird, ein neuer Aufwand entstehen. Ebenso steht offen, ob es überhaupt objektive und effizient überprüfbare Kriterien gibt, eine NGO und ihre Kategorien mit einer Bewertung von 0 bis 1 zu versehen.

Auch ist fraglich, ob es grundlegend sinnvoll ist, eine Eigenschaft "engagiert sich in einen Bereich" in einem gegebenen Bereich zu messen.

Außerdem ist es für den Nutzer möglicherweise abschreckend, eine Frage nach der Gewichtung beantworten zu müssen, weil nicht klar ist, was es bedeuten soll, Umweltschutz auf einer Skala von 0 bis 1 zu bewerten.

Aus diesen Gründen haben wir uns entschieden, die Kategorien einer NGO nur als Filter zu betrachten, und nicht in das Scoring aufzunehmen.

### 3.4.4 Lokalität

Nach Angaben unseres Projektstellers ist die Lokalität ein wichtiger Faktor für einen Nutzer, um sich mit einer NGO zu identifizieren. Deswegen war es für uns wichtig, die Lokalität ins Scoring einzubauen, also eine Möglichkeit zu haben, um die Nähe zwischen einem Nutzer und einer NGO zu berechnen.

Als Grundlage, um diese Nähe zu berechnen, verwenden wir Postleitzahlen (im Folgenden: PLZ). Der Nutzer kann beim Beantworten der Fragen eine PLZ eingeben, die PLZ für die NGOs sind in Form einer Datenbank hinterlegt.

Das Ziel dabei lautet: Wir wollen bei Eingabe zweier PLZs wissen, wie weit diese voneinander entfernt sind, um dem Nutzer Ergebnisse vorschlagen zu können, die in seiner Nähe liegen. Dies kann als Grundlage für ein Scoring genutzt werden, bei dem weiter entfernte NGO niedriger bewertet werden als in der Nähe befindliche, oder auch als Filter, wenn der Nutzer nur nahe NGOs angezeigt bekommen möchte.

#### 3.4.4.1 Postleitzahl-Vergleich

Ein erster Ansatz für die Lokalität war, die Struktur von deutschen Postleitzahlen auszunutzen. In Deutschland gibt es insgesamt etwa 28.683 Postleitzahlen. Diese bestehen aus genau 5 Ziffern. Diese sind anhand ihrer ersten Ziffer in "Regionen" aufgeteilt, anhand

der zweiten Ziffern nochmal in "Zonen" oder "Postleitregionen". So gibt es 10 Regionen, die wiederum nach der zweiten Ziffer in Postleitregionen unterteilt werden.<sup>1</sup>

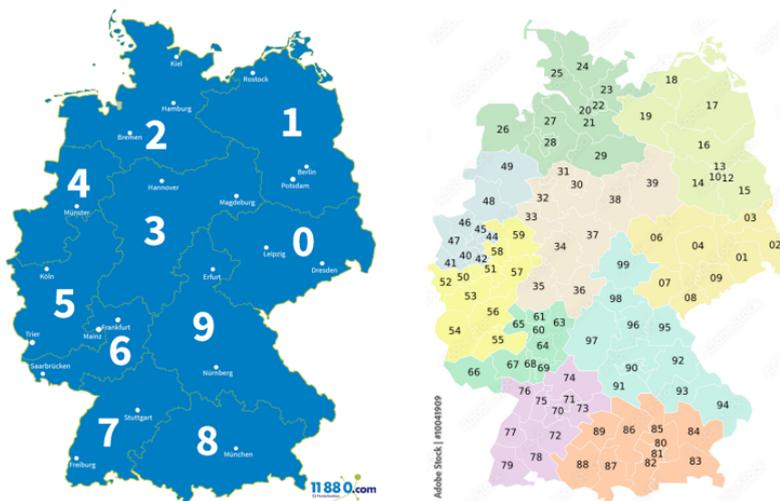
Zum Beispiel: Zu Würzburg gehören mehrere verschiedene PLZ: 97070, 97072, 97072,etc..., die aber alle mit "97" beginnen.

Diese Struktur ermöglicht es, die Distanz zwischen zwei Postleitzahlen auszudrücken als "liegen in derselben Region" oder "liegen in derselben Leitregionen". Da hierfür nur ein simpler String-Vergleich der Präfixe nötig ist, der also die ersten beiden Stellen einer PLZ vergleicht, war dies die erste Idee, um einfach eine Distanz berechnen zu können.

Dieser Vergleich reicht aus, um einen Nutzer aus Würzburg, andere NGO mit Standort in Würzburg anzuzeigen. Weiter könnte man einem Nutzer aus Würzburg alle NGOs anzeigen, die ansonsten mit einer 9 beginnen, da sich diese in derselben Zone mit Ziffer 9 befinden.

Ein Nachteil dieser Methode ist die Ungenauigkeit, vor allem wenn die Präfixe von nahen Orten nicht übereinstimmen. Dies sieht man besonders an Grenzen zwischen verschiedenen Postleitregionen, wie z.B. zwischen "97" und "63". Nach unserem Vergleich sind PLZs aus diesen Regionen entfernt voneinander, weil sie kein gemeinsames Präfix haben. Auf der Karte sieht man jedoch, dass sie benachbart sind. Es wäre also ein Fehler, einem Nutzer aus der Region "97" keine NGO aus der Region "63" als "nahe" anzuzeigen.

Ein weiterer Nachteil ist, dass die Entfernung zwischen unterschiedlichen PLZ damit nicht quantitativ gemessen werden kann, es kann also nicht angegeben werden, wie weit unterschiedliche PLZs voneinander entfernt sind. Die PLZ-Regionen mit Startziffer 6 und 9 sind unterschiedlich, aber liegen im Süden von Deutschland beieinander, während die Regionen 6 und 1 weiter entfernt liegen.



<sup>2</sup>Links: Postregionen

<sup>3</sup>Rechts: Postleitregionen

<sup>1</sup> [https://de.wikipedia.org/wiki/Postleitzahl\\_\(Deutschland\)#Systematik](https://de.wikipedia.org/wiki/Postleitzahl_(Deutschland)#Systematik)

<sup>2</sup>Quelle: [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-5ZaejbWxGNwFeciSgK4i2tGMgGI5ze9al--2bn3c7XdXUhlJuKc\\_WguWos\\_cTHGyYfE&usqp=CAU](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-5ZaejbWxGNwFeciSgK4i2tGMgGI5ze9al--2bn3c7XdXUhlJuKc_WguWos_cTHGyYfE&usqp=CAU)

<sup>3</sup> Quelle:

[https://as2.ftcdn.net/v2/jpg/00/10/04/19/1000\\_F\\_10041909\\_vyS1HCYDMYUBVKnzNbKiLaMbEnUfaHm.jpg](https://as2.ftcdn.net/v2/jpg/00/10/04/19/1000_F_10041909_vyS1HCYDMYUBVKnzNbKiLaMbEnUfaHm.jpg)

### 3.4.4.2 Distanz in Kilometer

Wegen dieser Probleme brauchten wir eine neue Variante, die Distanz zwischen PLZ zu berechnen, die eben nicht mit abgeschlossenen Regionen arbeitet. Die mathematisch exakte Vorgehensweise dafür ist, die Distanz zwischen zwei Koordinaten auf der Erdoberfläche zu berechnen.

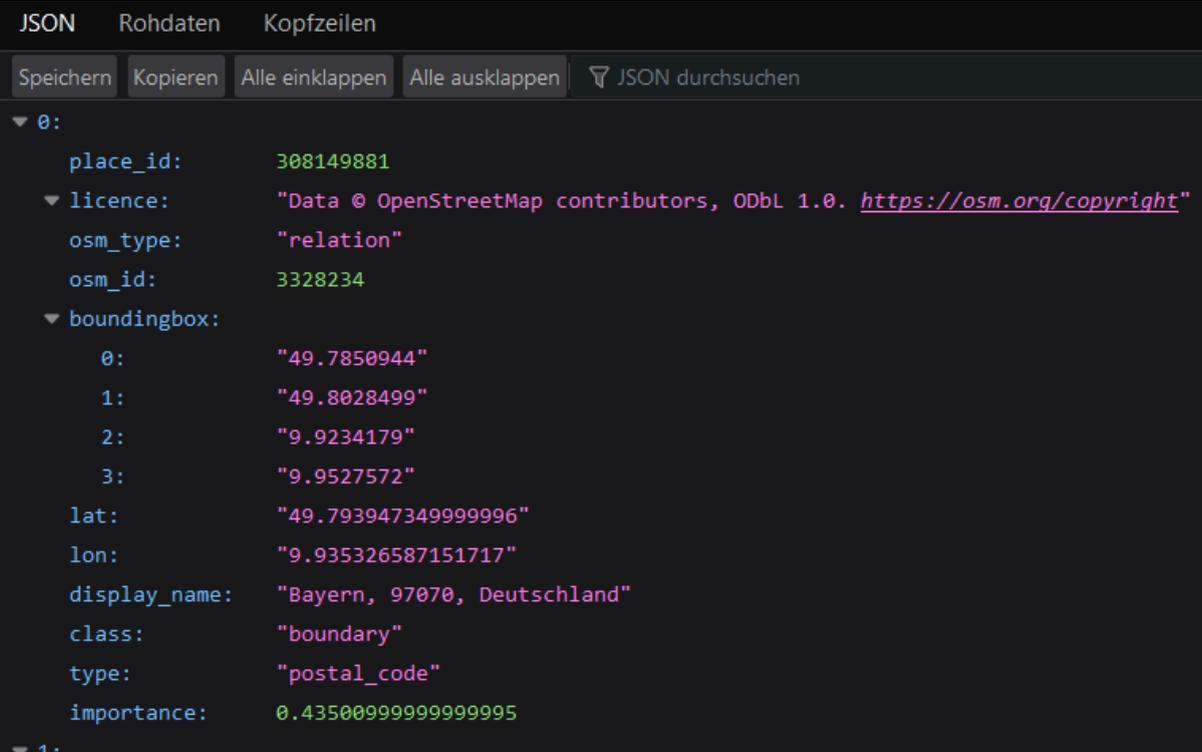
Nach kurzer Online-Recherche sind wir auf die Haversine-Formel gestoßen, mit der man die Distanz zweier Punkte auf der Erdoberfläche berechnen kann, sofern man die Längen- und Breitengrade der Punkte kennt.<sup>4</sup>

Diese Berechnung kann in einem kurzen Stück Python-Code implementiert werden, sofern die Koordinaten als Längen- und Breitengrad vorliegen. Das Problem besteht also nicht in der Berechnung, sondern in der Beschaffung der Koordinaten für eine PLZ.

Nach weiterer Online-Recherche sind wir auf eine Website gestoßen, an der API-Anfragen formuliert werden können. Bei Eingabe einer PLZ und zugehörigen Länderkürzel bekommt man die geographischen Koordinaten gesendet im JSON-Format gesendet. Die folgende URL entspricht dem Aufruf für die Koordinaten von Würzburg. Als Parameter werden dem Aufruf das Länderkürzel "country=de" und die PLZ "postalcode=97070" mitgegeben:

<https://nominatim.openstreetmap.org/search?country=de&postalcode=97070&format=json>

Das Ergebnis kann auch in einem Browser angezeigt werden, in Python lässt sich der Aufruf über das "requests"-Modul einlesen.



```
JSON Rohdaten Kopfzeilen
Speichern Kopieren Alle einklappen Alle ausklappen 🔍 JSON durchsuchen
0:
  place_id: 308149881
  licence: "Data © OpenStreetMap contributors, ODbL 1.0. https://osm.org/copyright"
  osm_type: "relation"
  osm_id: 3328234
  boundingbox:
    0: "49.7850944"
    1: "49.8028499"
    2: "9.9234179"
    3: "9.9527572"
  lat: "49.793947349999996"
  lon: "9.935326587151717"
  display_name: "Bayern, 97070, Deutschland"
  class: "boundary"
  type: "postal_code"
  importance: 0.4350099999999995
1:
```

<sup>4</sup> [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)

Da bei diesem Ansatz, aber für jede PLZ ein API-Call getätigt werden muss, ist die Laufzeit deutlich zu lang, um sie, für einen Nutzer, für jede NGO in der Datenbank durchführen zu können. Der Vorteil ist aber, dass man auch von beliebigen Ländern die Koordinaten abfragen kann, sofern man die PLZ im jeweiligen Land kennt.

Um diesen Aufwand zumindest für Standorte in Deutschland zu vermeiden, haben wir nach einem Datensatz gesucht, der bereits alle Koordinaten zu deutschen PLZs enthält.

Durch Recherche haben wir im Internet eine Datei gefunden, die zu allen deutschen Postleitzahlen die Namen und die geographischen Koordinaten enthält, diese haben wir in ein JSON-Format umgewandelt.<sup>5</sup>

Damit haben wir jetzt eine Möglichkeit, für jede deutsche Postleitzahl die geographischen Koordinaten abzulesen, ohne den Umweg über eine Web-Anfrage zu gehen. Die Datei kann auch als Basis für eine Datenbank genutzt werden.

Für nicht-deutsche Postleitzahlen muss man die geographischen Koordinaten weiterhin aus einer anderen Quelle beschaffen, wie die API-Abfrage. Das wird ein Problem, sobald nach mehr nicht-deutschen Postleitzahlen gefragt wird, wenn also Nutzer einen nicht-deutschen Standort angeben, oder neugedacht mehr NGOs in ihr Portfolio aufnehmen, die ihren Sitz nicht in Deutschland haben.

Die Frage dabei ist auch, wie wichtig die Entfernung in Kilometern für das Scoring von nicht-deutschen NGOs ist. Man könnte auch die Festlegung treffen, dass aus Sicht einer deutschen PLZ, nicht-deutsche Standort als "entfernt" gelten, ungeachtet der Größe.

#### 3.4.4.3 Weitere Probleme

Ein bestehendes Problem der Lokalität ist das einer NGO, die an verschiedenen Orten tätig ist, aber nur eine PLZ hat, die z.B. den Hauptsitz angibt. Bisher speichern wir für eine NGO nur eine PLZ in der Tabelle, es ist damit also nicht möglich, für eine NGO mehrere Orte anzugeben, an denen sie tätig ist. Eine Möglichkeit zur Anpassung ist, zu einer NGO eben mehrere PLZ abzuspeichern, und bei Berechnung der Lokalität nur die PLZ zu bewerten, die den geringsten Abstand zum Nutzer hat.

Beispiel: Eine Organisation mit Hauptsitz in Berlin, aber mit "Wirkungsstätten" in Berlin, München und Frankfurt. Diese NGO hat nach jetzigem Stand in unserer Datenbank nur eine PLZ, den Hauptsitz in Berlin. So wird sie für einen Nutzer, der als Standort Frankfurt angibt, als weiter entfernte NGO angezeigt, obwohl sie ja eine "Wirkungsstätte" in Frankfurt, also in der Nähe des Nutzers, hat.

## 3.5 Beiträge der Gruppenmitglieder

Die vier Mitglieder der Gruppe sind Christian Koch, Marvin Henn, Nicolas Steigerwald und Niklas Vogt.

---

<sup>5</sup> <https://gist.github.com/iteufel/af379872bbc3bf5261e2fd09b681ff7e>

Christian nahm sich der Konzeptionierung des grundlegenden Algorithmus an, wobei er Rücksicht auf das Schreiben des Codes und die Konzeptionierung der Fragen nahm, bei der er ebenso großen Anteil an der Findung hatte und viel zur Diskussion darüber beitrug. Ebenso half er bei Problemen im Code und konzipierte das Video.

Marvin schuf anfangs auf Basis des Algorithmus die Testumgebung, die er im Laufe des Semesters immer erweiterte, bis ein allgemeiner, sehr gut strukturierter Code entstand, der leicht erweiterbar ist und den wir letzten Endes stolz übergeben konnten.

Nicolas half bei der Findung der Fragen, unterstützte stark beim Code schreiben und verfeinern und kümmerte sich um die Präsentationsfolien.

Niklas erstellte mehrmals Kategorisierungen, die alle NGOs abdecken und einige Konzepte zur Reihenfolge der Fragen, die er sich ebenfalls mit ausdachte. Er kümmerte sich ebenfalls um die Präsentationen und traf sich zusammen mit Moritz mit einer Psychologiestudentin, um die Fragen zu finalisieren, die dann im Code festgelegt wurden.

Der Projektbericht ist in mehreren Meetings zusammen geschrieben worden. Dies beinhaltet für alle das Schreiben einzelner Passagen zu den jeweiligen Schwerpunkten, das Korrekturlesen und Vorschläge zur Verbesserung.

## 4. Projektergebnisse und Diskussion

### 4.1 Einordnung

neugedacht ist im Aufbau einer Anwendung, die es Arbeitgebern und Arbeitnehmern ermöglicht, an im Portfolio stehende NGOs zu spenden.

Das bedeutet, dass unser Endprodukt in diese einsetzbar sein muss.

Zusätzliche Tools oder Module vom Projektgeber gab es nicht.

Der Code, den wir entworfen haben, dient als Schnittstelle zwischen einem Frontend, das die Antworten des Arbeitnehmers entgegennimmt und das wir nicht entwickeln und unserem Algorithmus, beziehungsweise unserer Datenbank, die zuvor noch nicht existiert hat.

Unser Python-Code startet, sobald es einen Kommandozeilenaufruf-Aufruf mit den Antworten des Nutzers erhält. Nach deren Verarbeitung wird eine Liste an NGOs zurückgegeben, die die entsprechend höchsten Scores erzielt haben.

### 4.2 Vorgehen

Wir haben uns für Python entschieden, da es eine Sprache ist, die wir gut beherrschen und da wir uns zu diesem Zeitpunkt schon sicher waren, dass wir auch mit Datenbanken arbeiten müssen, bot sich die Wahl umso mehr an.

#### 4.2.1 Ideen

Bei unserem Projektziel war von Anfang an klar, dass das Ziel "einem Nutzer gute NGOs vorschlagen" kein hartes/quantitatives Kriterium ist, was objektiv erfüllt werden kann oder nicht. Die Idee bestand also darin, Konzepte zu entwickeln und diese dann im Code zu testen.

Ein Ziel beim Code war die Variabilität, sodass neugedacht später einfach Änderungen durchführen kann.

Bei Entscheidungen haben wir versucht, dem Team von neugedacht nichts vorzuschreiben und leicht änderbaren Code zu erstellen. Somit ermöglichen wir neugedacht, unser Projekt an ihre Bedingungen anzupassen.

#### 4.2.2 Testumgebung

Weil unser Projektziel darin besteht, Code zu schreiben, der später in eine Webanwendung eingebunden werden soll, haben wir versucht, Teile der Datenstruktur in einer Testumgebung nachzubilden. Das ist wichtig für ein iteratives Vorgehen, gerade wenn wir unseren Code schrittweise erweitern wollen. Dadurch können wir direkt in der Umgebung testen, ob unsere theoretischen Annahmen für den Scoring-Algorithmus auch sinnvolle Ergebnisse liefern.

Im Anschluss konnten wir mithilfe der Testumgebung den ersten Versuch eines Scoring Algorithmus, welcher zunächst seine Antwortdaten aus der Kommandozeile zieht, implementieren und die Qualität der Ergebnisse testen.

### 4.2.3 Entwicklung des Skripts

Nachdem die Testumgebung und der Fragenkatalog standen, konnten wir endlich an die Entwicklung des eigentlichen Skripts gehen. Nachdem wir mehrere Versionen des Scoring-Verfahrens, sowie der Verwendung der Kategorien getestet haben, entschieden wir uns für das additive und modulare Scoring. Außerdem filtern wir die Kategorien. Dies zeigt in unseren Testläufen durchweg die qualitativ besten Ergebnisse.

## 4.3 Resultate

Der finale Code beinhaltet den Scoring Algorithmus in Form einer ConfigFile, über die mehrere Modifizierungen vorgenommen werden können, Algorithmen, die der Regionalitätsfrage dienen, der eine berechnet die Distanz zwischen zwei gegebenen Koordinaten in km und der andere bildet alle deutschen Postleitzahlen auf Koordinaten ab.

Darüber hinaus übergeben wir mit dem Code eine Dokumentation dieses.

Diese enthält Anleitungen, wie die Testumgebung genutzt werden kann. Damit ist es möglich, Implementierungen von Scorings oder Fragen lokal zu testen, ohne eine Website nutzen zu müssen.

Ebenso ist die Testumgebung inbegriffen, mit der Datenbanken mit zufällig generierten NGOs erstellt werden können, der Fragenkatalog ausgefüllt und somit ein Durchlauf simuliert und ein Scoring Ergebnis wiedergegeben werden kann.

Das Ergebnis des Scorings wird im JSON-Format mit id-Werten zurückgegeben, welche sich auf eine geschlossene Datenbank beziehen.

Der Code muss, um einsatzfähig zu sein, mit der Config-File verbunden werden.

Er kann außerdem direkt ins Backend eingebunden werden.

Es können ebenfalls Fragen ergänzt und deaktiviert werden, neue Tabellen erstellt und vorhandene verändert werden.

```
python.exe .\main.py -a {"groesse_frage": "Klein",
  "region_frage": "0",
  "plz_frage": "97074",
  "distanz_frage": 500,
  "religion_gewollt_frage": "0",
  "verwalten_frage": "Beides",
  "kategorien_frage": ["Gesundheitswesen",
  "Hilfe fuer Beduerftige",
  "Kinder- und Jugend"]}
```

```
[
  {
    "NGO_id": 87,
    "score": 3.0
  },
  {
    "NGO_id": 121,
    "score": 3.0
  },
  {
    "NGO_id": 8,
    "score": 2.5
  },
  {
    "NGO_id": 14,
    "score": 2.5
  },
  {
    "NGO_id": 28,
    "score": 2.5
  },
  {
    "NGO_id": 35,
    "score": 2.5
  },
],
```

Auf dem ersten dieser Screenshots sieht man eine Anfrage an unser Skript, im zweiten eine Antwort auf diese.

## 4.4 Diskussion

Im Folgenden soll diskutiert werden, ob das Ergebnis den gesetzten Zielen entspricht, an welchen Punkten unsere Herangehensweise gut funktioniert hat und was wir hätten besser machen können.

Die Fragen wurden, wie bereits oben genauer ausgeführt, mit großer Sorgfalt ausgewählt und hinterfragt, bis jeder Beteiligte damit zufrieden war.

Der Algorithmus ist ein simpler, aber effektiver Prozess, um die NGOs zu sortieren und ist damit auch durchaus gut skalierbar.

Der Code ist einfach zu verstehen, sauber dokumentiert und funktioniert wie gewünscht.

Er ist durchaus nachhaltig, da er einfach abzuändern ist und somit kein weiteres Entwicklerteam damit beauftragt werden muss.

Beim Schreiben des Codes gab es ab und zu Kommunikationsschwierigkeiten, da lange unklar war, wie wir den Algorithmus auf die gegebenen Fragen anwenden wollen, was mit

der Dauer der Fragenfindung zusammenhang. Hier hätten wir viel früher eingreifen können und früher kontrollieren können, ob wir uns in die richtige Richtung bewegen.

Das genaue Formulieren der Fragen ist ohne den Rahmen dieses Seminars zu sprengen vermutlich nicht perfekt möglich. Außerdem wäre das eher ein Schnitt Thema der Soziologie und Informatik. Die Entscheidung, an einem Punkt die Fragen fix zu definieren und weiterzuarbeiten, hat sich im Zeitplan als richtig herausgestellt.

Wir hätten die anfangs erstellte Testumgebung deutlich intensiver nutzen können. So hat es sich im Arbeitsaufwand beinahe nicht gelohnt. Weiterhin hätten wir uns von Anfang an im Code auf Best-Practices einigen müssen, da durch die unterschiedlichen Stile später Schwierigkeiten bei der Zusammenarbeit entstanden sind. Dennoch sind wir gut und im Zeitplan durch die Programmierarbeit gekommen.

Der Algorithmus wurde im Laufe des Projektes angepasst und im Bezug auf die finalen Fragen finalisiert. Die fertige graphische Definition hat sehr geholfen, diesen leicht in den Code umzusetzen. Auch wenn der Algorithmus sicherlich noch weiter optimiert werden kann, können wir so sehr schnell in die weiteren Phasen einsteigen.

Wir hätten ebenso eine Testversion einer Benutzeroberfläche bauen können, haben uns aber dafür entschieden, es in der Kommandozeile von Python ausgeben zu lassen, um die Komplexität nicht unnötig zu erhöhen, da diese am Ende sowieso verworfen wird.

Zusammengefasst waren die Zeiten, die wir in die Teilprobleme des Projektes investiert haben, teilweise ein wenig zu lang, was dem erfolgreichen Verlauf jedoch nicht großartig im Weg stand.

Das Endergebnis in Form des fertigen Codes erfüllt die Problemstellung des Auftraggebers vollkommen und wird ein entscheidender Teil für die Zielgruppe des Fragenkatalogs in der Suche nach individuell passenden Organisationen sein, zu denen die Arbeitnehmer dann ihren Anteil an der Unternehmensspende spenden können.

## 4.5 Zukunft

Dieser Absatz soll einen Ausblick darüber geben, wie unser Projekt voraussichtlich eingesetzt werden wird.

Der fertige Code wird an neugedacht übergeben. Das Unternehmen ist aktuell dabei, mithilfe einer Softwarefirma ihre Anwendung entwickeln zu lassen. Dieser Firma wird unser Quellcode gemeinsam mit der Dokumentation übergeben. Sobald das Frontend fertig ist, kann unser Python-Skript verwendet werden, um die vorhandenen Spendenziele mithilfe der durch das Frontend gesammelten Antworten zu filtern.

Um diesen Prozess so einfach wie möglich für Softwareentwickler zu gestalten, kann unsere Dokumentation des Codes genutzt werden.

Weiter ermöglicht die Configfile die Fragen auf Wunsch zu deaktivieren, sowie weitere Anpassungen vorzunehmen.

Dank unseres ausgearbeiteten Algorithmus helfen wir so den Mitarbeitenden, ihr ideales Spendenziel zu finden, sowie den gemeinnützigen Organisationen ihren Anteil an den Unternehmensspenden zu erhalten.

## 5. Anhang

- Dokumentation
- Code