

# AI SMART WATERING



**Ein Projekt von Viyona und Aarav Singh**

**Gymnasium Gröbenzell**

## Kurzfassung

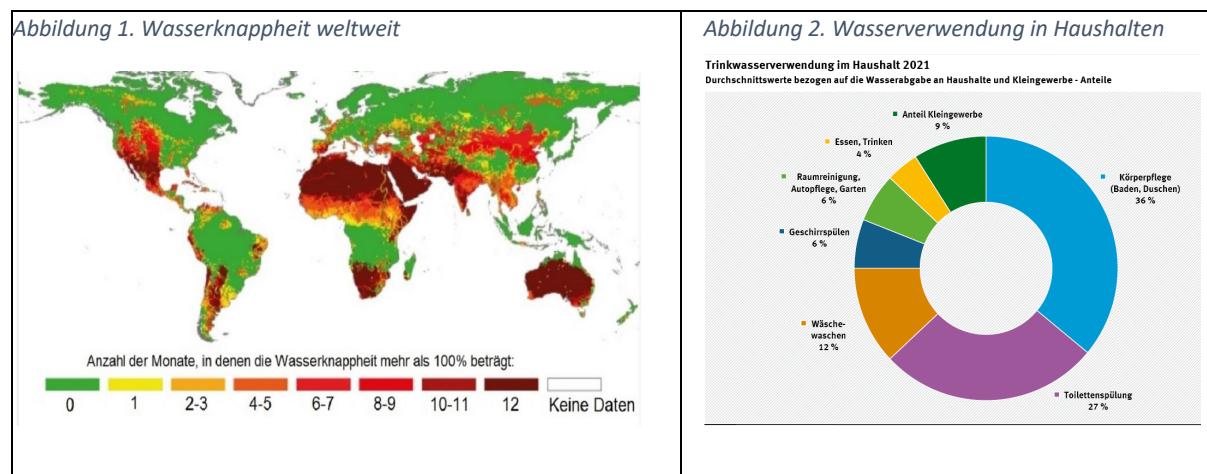
Aufgrund der zunehmenden Wasserknappheit in vielen Ländern und auch bei uns in Deutschland, kamen wir auf die Idee etwas dagegen zu unternehmen. Eine Ursache für den zunehmenden Wassermangel ist die verschwenderische Bewässerung von Gartenanlagen. Mithilfe von Wetterprognosen und eines Regression-Modells, welches die Bodenfeuchte einschätzt, kann so die optimale Gießmenge berechnet werden. Dadurch kann Gießwasser gespart werden und die Pflanzen verwelken nicht. Die erhobenen und berechneten Daten wurden in eine Android App integriert und zusätzlich mit wassersparenden Tipps versehen.

# Inhaltsverzeichnis

1	Einleitung .....	2
2	Problemformulierung.....	3
2.1	Goldene Regeln fürs Gießen.....	3
2.2	Bodenfeuchte.....	3
2.3	Das Problem in zwei Teilen .....	5
3	KI-Modelle .....	5
3.1	Was ist ein KI-Modell?.....	5
3.2	Arten von KI-Modellen .....	5
4	Die Lösung: AI Smart Watering .....	7
4.1	Teil 1. Bodenfeuchte schätzen .....	7
4.1.1	Messgerät und Programmierung .....	7
4.1.2	Daten sammeln .....	8
4.1.3	Model Entwicklung.....	8
4.2	Teil 2. Berechnung von optimaler Gießmenge.....	9
4.3	Wasserersparnis.....	10
5	AI Smart-Watering App .....	11
5.1	Einstellungsseite.....	11
5.2	Hauptseite .....	12
6	Zusammenfassung .....	13
6.1	Probleme .....	13
6.2	Weitere Schritte .....	13
7	Quellenangaben.....	14
8	Unterstützungsleistungen .....	15

# 1 Einleitung

Rund zwei Milliarden Menschen haben keinen regelmäßigen Zugang zu sauberem Wasser. Das entspricht etwa ein Viertel der Weltbevölkerung. Die Länder in Afrika, Asien und in Australien sind schwer mit Wassermangel betroffen. Abb. (1) stellt die Wasserknappheit weltweit dar.



Wassermängel können zu schweren Schäden von Gehirn und Niere und auch zum Tod führen. Täglich sterben weltweit mehr als 1.000 Kinder unter fünf Jahren an Krankheiten, die durch verschmutztes Wasser und mangelnde Hygiene verursacht werden. Der Mangel an sauberem Wasser und Hygiene gehören immer noch zu den häufigsten Todesursachen bei kleinen Kindern.

Weltweit werden rund 4000 km<sup>3</sup> Wasser pro Jahr entnommen [1]. Aber wie kommt es zu so einer großen Zahl? Abb. (2) zeigt die Verbrauchsmengen an Wasser bei täglichen Arbeiten in Deutschland. Die Bewässerung des Gartens spielt dabei keine unwichtige Rolle. Pro Quadratmeter Grünfläche werden laut einer verbreiteten Faustregel etwa 60 Liter Bewässerung pro Jahr benötigt. Wenn man geschickt gießen würde bräuchte man nur die Hälfte davon. Woher weiß man die genaue Menge an Wasser, die benötigt wird? Der Gießbetrag hängt vom Wetter, Pflanzenart, Bodenart, Wettervorhersage etc. ab. Doch wer macht sich die Mühe, beachtet alle Eigenschaften und berechnet dann die genaue Gießmenge? Eine Möglichkeit wäre eine Sensorüberwachung, die misst, wie viel Bodenfeuchte im Boden ist. Diese Methode wäre jedoch sehr aufwendig und die Sensoren produzieren eine große Menge an Müll, da sie nicht nachhaltig sind.

Damit keine Sensoren mehr gekauft werden müssen, wurde ein KI-Modell trainiert, welches die Bodenfeuchte mithilfe von Wetterprognosen eigenständig einschätzt. Durch die Bodenfeuchte kann der optimale Gießbetrag berechnet werden. Die Lösung AI Smart Watering wurde in eine benutzerfreundliche Android App eingebaut, in der der Benutzer den optimalen Gießbetrag finden kann.

# 2 Problemformulierung

## 2.1 Goldene Regeln fürs Gießen

Es gibt goldene Regeln, die empfehlen, wie und wann man am besten gießen soll. Die erste Regel bezieht sich auf die Gießzeit: In Hitzeperioden wird am besten frühmorgens gegossen. Damit sind die Pflanzen gut versorgt und die Mittagstemperaturen können gut überstanden werden. Die Blätter sollen trocken gehalten werden, damit die Gefahr von Pilzerkrankungen gering bleibt. Die zweite besagt wie viel man gießen soll: Man soll selten, aber durchdringend wässern. Dadurch gelangt das Wasser auch in die tieferen Bodenschichten. Auf diesem Link kann man sich mehr über die Regeln informieren [2].

Wir untersuchen die Regel „Selten gießen, aber durchdringend gießen“. Bei Fällen, wo es in den nächsten Tagen lange nicht regnet, ist diese Regel sinnvoll. Sonst besteht die Gefahr, dass man umsonst Wasser vergießt, obwohl es am nächsten Tag schon heftig regnet. Nach der goldenen Regel „Selten gießen, aber durchdringend gießen“ würde man in diesem Beispiel unglaublich viel Wasser verschwenden. Außerdem ist es riskant, da die Wurzel der Pflanzen aufgrund Staunässe keine Zellatmung betreiben können und erstickten. Somit stellt sich die Frage: Wie viel soll man genau gießen? Um diese Frage zu beantworten, muss sich mit der Bodenfeuchte auseinandersetzen werden.

## 2.2 Bodenfeuchte

Die Bodenfeuchte ist der Wassergehalt der oberen Bodenschicht (0 bis 2 m). Sie bezeichnet das Wasser, welches durch hygrokopische und kapillare Kräfte entgegen der Schwerkraft in den oberen Bodenschichten festgehalten wird. Neben der Bodenfeuchte gibt es auch die Feldkapazität (FK) und den Welkepunkt (WP). Unter Feldkapazität versteht man, wie viel Wasser diejenige Bodenart 2 bis 3 Tage lang halten kann. Der Welkepunkt ist die Menge an Wasser, die Pflanzen nicht entnehmen können (Punkt, an dem Pflanzen anfangen zu welken). Die nutzbare Feldkapazität ( $nFK\ %$ ) beschreibt den Füllstand des pflanzenverfügbaren Bodenwasserspeichers, wobei die bodenbezogenen Standorteigenschaften (Feldkapazität und Welkepunkt) berücksichtigt werden [3].

Insofern wird die Differenz der Feldkapazität und des Welkepunkts als nutzbare Feldkapazität ( $nFK = FK - WP$ ) bezeichnet. Somit ist die  $nFK$  ein entscheidender Faktor für das gute Pflanzenwachstum (siehe Abb. (3)).

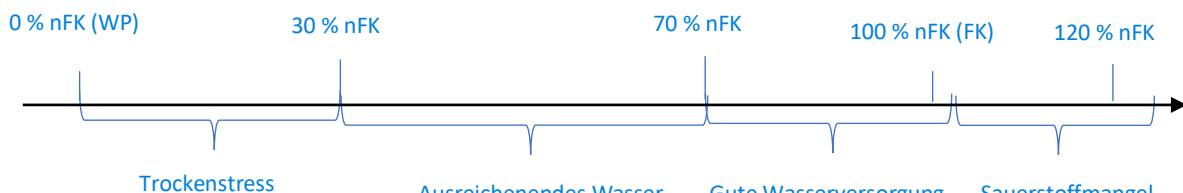


Abbildung 3. Bedeutung von der  $nFK$  an Wasserversorgung

Bei intensiven Niederschlagsereignissen kann die Bodenfeuchte auch kurzzeitig auf Werte über 100 %  $nFK$  ansteigen. Die Bodenfeuchte kann auch unter 0 %  $nFK$  absinken, wenn es sehr trocken ist. Diese hängt also von vielen Faktoren ab:

**Wetter:** Bei höheren Temperaturen verliert der Boden seine Feuchte schneller als bei niedrigeren. Beim Niederschlag steigt die Bodenfeuchte, beim trockenen Wetter fällt sie.

**Bodenart:** Jede Bodenart hat ihre eigene Feldkapazität und Welkepunkt. Abb. (4) zeigt verschiedene Bodenarten und deren FK und WP.

Abbildung 4. Tabelle mit Beispielen von Bodenarten und ihren Feldkapazitäten und Welkepunkten. Da Sand im Gegenteil zu

Bodenart	Feldkapazität	Welkepunkt
Sand	20	3
Schwach sandiger Lehm	35	17
Schluff	39	11
Ton	41	29

Ton größere Lücken hat, ist die Feldkapazität niedriger (Wasser sickert leichter durch die Lücken).

Abb. (5) stellt eine Bodenübersichtskarte Deutschlands dar. Mithilfe der Karte kann man die Bodenart einer Region bestimmen, und damit die FK und WP rechnen.

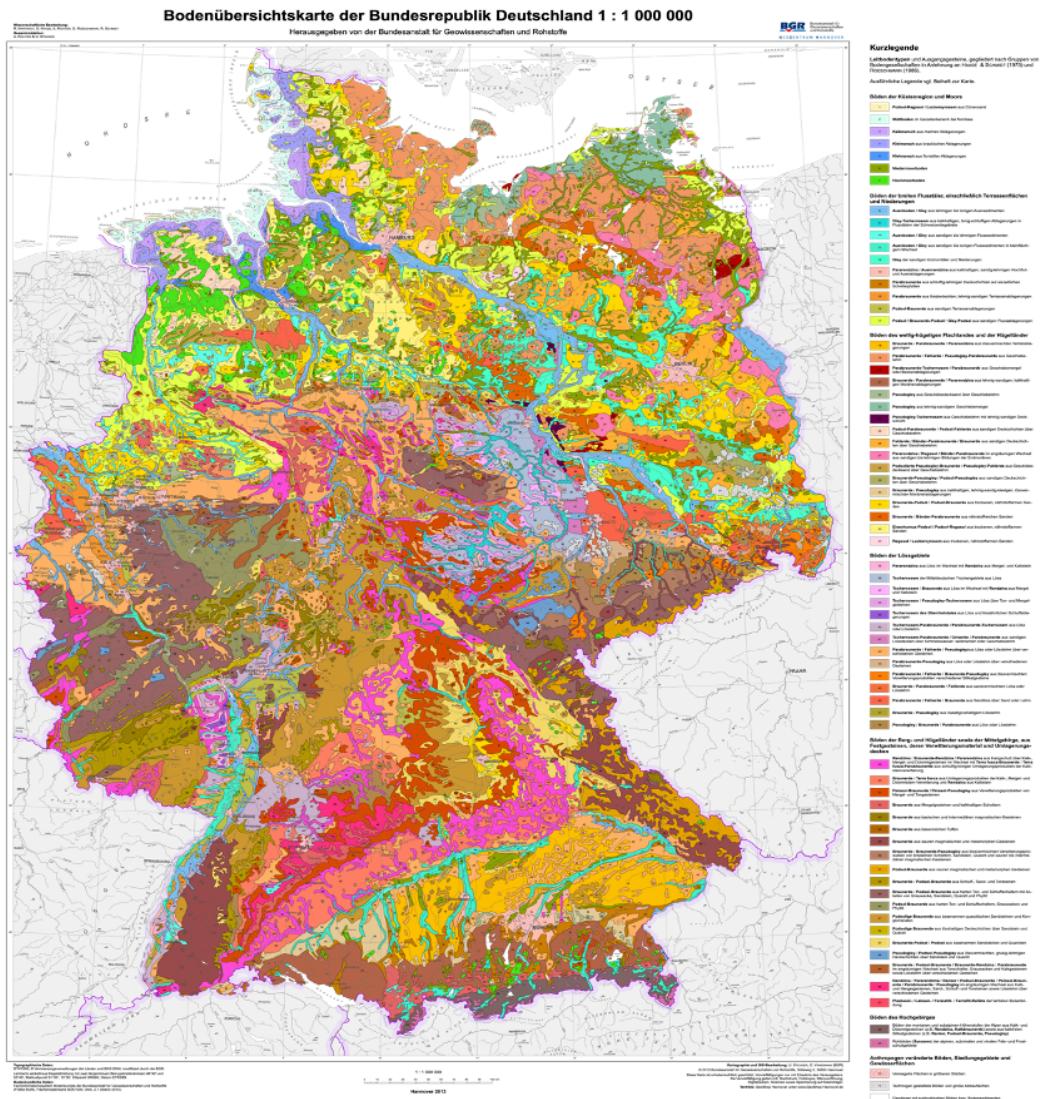


Abbildung 5. Bodenübersichtskarte für Deutschland

## 2.3 Das Problem in zwei Teilen

**Teil 1. Bodenfeuchte schätzen.** Es gibt verschiedene Sensoren, um die Bodenfeuchte (BF) zu messen. Leider korrodieren die meisten Sensoren nach einiger Zeit und landen in den Müll, deshalb müssten sie regelmäßig ersetzt werden. Deshalb soll die tägliche Änderung der Bodenfeuchte mithilfe von Wetterdaten und der Bodenart geschätzen werden. KI erleichtert die Schätzung und kann auch selbstständig ein ziemlich genaues Modell durch gegebene Daten entwickeln (siehe Abb. (6)).

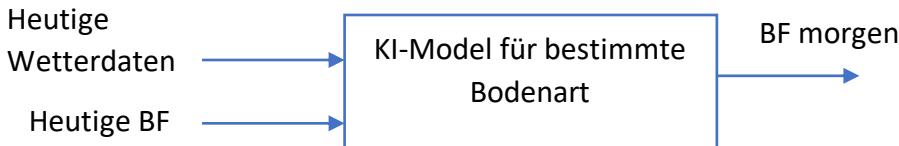


Abbildung 6. KI-Modell die Bodenfeuchte zu schätzen

**Teil 2. Optimale Gießmenge berechnen.** Mithilfe der geschätzten Bodenfeuchte und der Wetterprognose für die kommenden Tage kann man einen Algorithmus erstellen, der die optimale Gießmenge berechnet (gezeigt in Abb. (7)). Die optimale Gießmenge beträgt die Wassermenge, die den Boden für drei Tage über 30% nFK halten kann.

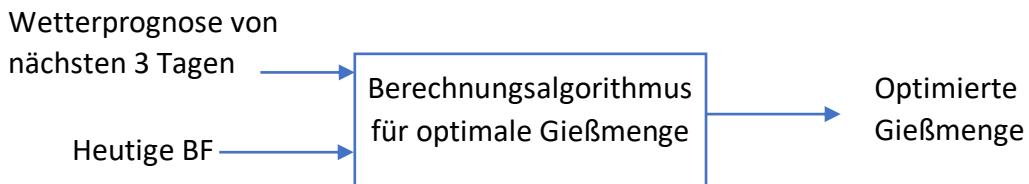


Abbildung 7. Algorithmus die optimale Gießmenge zu berechnen

## 3 KI-Modelle

### 3.1 Was ist ein KI-Modell?

Man kann die KI in zwei Teile unterteilen: Supervised Learning (im Deutschen überwachtes Lernen) und Unsupervised Learning (im Deutschen unüberwachtes Lernen). Um ein Modell zu trainieren, braucht man Datensatz (Trainingsdaten), also Daten, bei denen auch die Zielvariable gegeben ist. Beim überwachten Lernen muss man dem Computer genug Beispiele geben damit er ein Modell entwickeln kann. Dagegen braucht der Computer keine Trainingsdaten beim unüberwachten Lernen und kann gleich mit den Eingabedaten beginnen. Wir haben uns mit dem Zweig Supervised Learning beschäftigt. Supervised Learning ist in zwei Hauptteile verfasst: Classification und Regression. Bei beiden werden die Modelle trainiert, d.h. sie bekommen Daten, mit denen sie lernen können. Danach entwickelt der Computer ein Modell und testet es mit Daten, die er noch nicht gesehen hat. Falls Fehler auftauchen, versucht er das Modell zu verbessern.

### 3.2 Arten von KI-Modellen

**Classification (Klassifizierung):** In einem Klassifikationsmodell teilt das Modell Objekte in Klassen so auf, wie man sie aufgrund ihrer geteilten Eigenschaften oder Charakteristika wahrnimmt.

Beispiel: Man kann alle Lebensmittel mit einem Klassifikationsmodell in drei Kategorien aufteilen: Obst, Gemüse und Sonstiges. Diese kann man mit Farbe, Gewicht, Größe und Form beschreiben. Zuerst muss man dem Modell viele Beispiele vorzeigen:

Obst: Apfel (rot; 0,2 kg; 8cm breit; rund); Gemüse: Aubergine (lila; 0,5 kg; 15cm breit; krumm); ...

Dann wird ein anderes Lebensmittel, welches nicht unbedingt in den Beispielen vorkam, beschrieben und das Modell versucht herauszufinden, zu welcher Kategorie es gehört. Wenn die Kategorie nicht klar ist, wird das Objekt zu Sonstiges zugeteilt.

**Regression:** In einem Regressionsmodell wird ein Zusammenhang zwischen zwei oder mehreren Variablen angegeben. Eine Variable verändert sich in Abhängigkeit von einer oder mehreren unabhängigen Variablen. Diese Variable wird auch Reaktionsvariable genannt, da sie auf Veränderungen von den unabhängigen Variablen reagiert.

Beispiel: In der Abb. (8) kann man eine lineare Regression sehen. Der Batterieverbrauch ist die unabhängige Variable und die Bildschirmzeit stellt die Reaktionsvariable dar. Die roten Punkte sind Beispiele, mit welchen das Modell trainiert wurde.

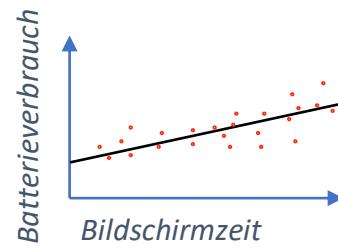


Abbildung 8. Ein Beispiel von einer Linearen Regression

Jetzt wird die genaue Mathematik, die hinter der linearen Regression steckt, beschrieben.

Beispiel: Angenommen es gibt einen Planeten, bei dem das Modell die Größe eines Neugeborenen mithilfe der Größen der Eltern bestimmen soll. Folgendes sind die Trainings-Daten:

Muttergröße	Vatergröße	Kindesgröße
[ 8 , 10 ]		[ 8,5 ]
[ 40 , 20 ]		[ 35 ]
[ 240 , 400 ]		[ 280 ]

Als Training-Daten eignen sich besonders gut eine Matrix, eine Art „2D Array“. In jede Spalte kommen Daten von verschiedenen Messungen. Wenn es nur eine Zahl als Endergebnis gibt, benutzt man einen Vektor.

Das Modell muss durch die Ergebnisse entziffert werden. Wir bezeichnen das Modell als [a, b, c]. Zum Anwenden des Modells multipliziert man die Daten mit ihren zugehörigen Anteilen und kommt so zum Ergebnis (in diesem Fall zur Kindesgröße).

Durch die Gleichungen muss das Modell bzw. [a, b, c] bestimmt werden:  $8 * a + 10 * b + c = 8,5$ ,  $40 * a + 20 * b + c = 35$ , usw.

Das Linear Regression Modell kann somit herausfinden, dass  $a = 0,75$  und  $b = 0,25$  und  $c = 0$ .

Dieses Modell ist zu 100 % richtig aber bei realen Beispielen mit viel mehr Features (Daten) und unterschiedlichen Anteilen kann das zu Ungenauigkeit führen.

## 4 Die Lösung: AI Smart Watering

Nun wird unsere Lösung beschrieben. Zuerst wird der Ansatz, wie die Bodenfeuchte geschätzt wurde, präsentiert. Im Folgenden wird erklärt, wie die optimale Gießmenge berechnet wird.

### 4.1 Teil 1. Bodenfeuchte schätzen

#### 4.1.1 Messgerät und Programmierung

Die Bodenfeuchte wurde mit der SenseBox [4] gemessen (siehe Abb. (9)). Diese beinhaltet viele Sensoren wie auch ein Bodenfeuchtesensor, ein Steckbrett und ein Display. Die Sensoren wurden mithilfe von Blockly programmiert. Das ist eine grafische App, welche man mit Zusammensetzungen von Blöcken programmiert. Mit dieser wurde ein Programm geschrieben, welche die vom Sensor gemessene Bodenfeuchte auf dem Display anzeigt (siehe Abb. (10)). Im Internet kann man sich über die Kabelverbindungen informieren (siehe Abb. (11)). Auf der SenseBox Seite [4] findet man mehr Informationen über den Bodenfeuchtesensor für die SenseBox.

Abbildung 9. SenseBox



Abbildung 10. Blockly Programm

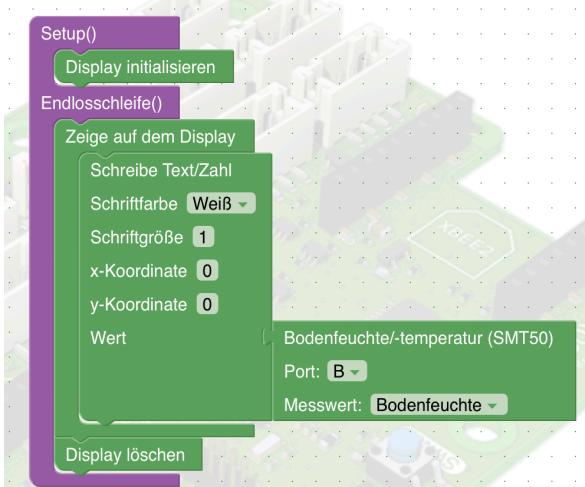


Abbildung 11. Messgerätverknüpfungen

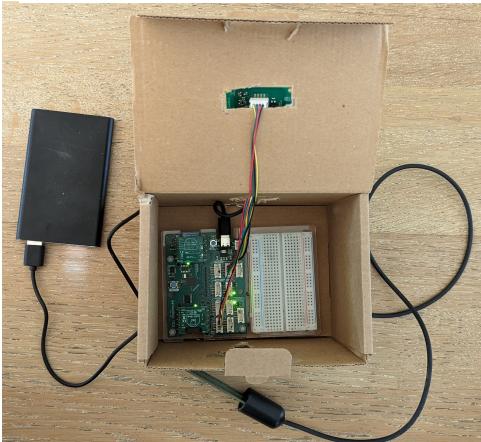


Abbildung 12. Messstelle im Garten



#### 4.1.2 Daten sammeln

Für das KI-Modell benötigt man Trainings-Daten, d.h. man muss dem Computer zuerst Beispiele vorgeben. Somit weiß es, wie die Bodenfeuchte in diesem Fall mit den Features (min. Temperatur; max. Temperatur; Regenmenge; ...) abhängt. Mit dem zusammengebauten Sensor wurde die Bodenfeuchte im Garten täglich gemessen und aufgeschrieben (siehe Abb. (12)). Zusätzlich wurden noch die Tagestemperaturen, der Bereich und die Regenmenge notiert. Man kann auch an mehreren Stellen mit verschiedenen Eigenschaften (z.B. schattig, sonnig, ...) messen und mehrere Modelle programmieren. Hier wurde abends im Sommer gemessen. Die Daten wurden mithilfe von einer Excel Tabelle aufgeschrieben. Für eine Rasenbewässerung misst man die Bodenfeuchte ungefähr 10cm bis 15cm tief. Doch die Tiefe, in der gemessen wird, ist bei allen Pflanzen unterschiedlich. Durch die Bodenübersichtskarte (Abb. (5)) haben wir abgelesen, dass wir Lehmboden in unserem Garten haben. Die Feldkapazität entspricht 35 und der Welkepunkt ist 11. Durch diese Werte wurde die Bodenfeuchte in nFK umgeschrieben.

Abb. (13) zeigt einen Ausschnitt der gesammelten Daten, die genutzt wurden, um das Modell zu trainieren. Diese wurden von Juni bis August 2023 gemessen.

A	B	C	D	E	F	G
Datum	min_temp	max_temp	Regen	Gießmenge	Bodenfeuchte	nFK
29. Jun		12	28	1	0	14 12.5
30. Jun		18	24	18	0	40 120.83333333
01. Jul		18	24	2	0	25 58.33333333
02. Jul		15	23	3	0	30 79.16666667
03. Jul		14	24	0	0	25 58.33333333
04. Jul		13	25	0	0	18 29.16666667
05. Jul		14	23	2	0	24 54.16666667

Abbildung 13 Ausschnitt von den Trainings Daten in der Excel Datei

#### 4.1.3 Model Entwicklung

Durch die gesammelten Daten wurde ein Modell programmiert. Wenn man sich gut mit der Programmiersprache Python auskennt, lohnt es sich mit der Bibliothek SciKit-Learn [5] zu arbeiten. Diese Bibliothek enthält viele Regression Modelle. Linear Regression ist die simpelste von den Modellen. Aber dieses Modell hat nicht zu unseren Datenpunkten

gepasst. Das Problem war, dass die Ergebnisse zu oft in den negativen Bereich gelangt sind, da das Modell linear ist. Durch Ausprobieren von verschiedenen Modellen wurden die Wertungen verglichen und das gesuchte Modell gefunden. Unsere Daten haben am besten mit dem Gradient Boosten-Algorithmus funktioniert.

Der Gradient Boosten-Algorithmus benutzt sehr viele ungenaue Modelle und formt daraus ein besseres Modell. Hierbei werden die Fehler in Form von Gradienten bewertet und das nächste Modell wird genutzt, um den Gradienten in die entgegengesetzte Richtung zu "boosten" und damit auszugleichen. Abb. (14) zeigt das Programm, dass das KI-Modell trainiert. Zunächst wurden die Daten von der Datei gelesen. Danach wurde die Funktion fit(X, y) gerufen, welche ein Modell erstellt. Dann mit der score(X, y) Funktion wurde die Qualität des Modells ermittelt. Das erstellte Modell hat eine Wertung von 0,9985 – das heißt, dass das Training gut funktioniert hat. Am Ende wurde das Modell abgespeichert.

```
def ki_modell_trainieren():
    data = pd.read_csv('Bewaesserung.csv', index_col=False, header=0, delimiter=' ; ')
    print(data.columns)
    (min_temp, max_temp, bodenfeuchte, regen, giessmenge) = \
        data['min_temp'], data['max_temp'], data['bodenfeuchte'], data['regen'], data['giessmenge']
    print(min_temp, max_temp, bodenfeuchte, regen)
    y = bodenfeuchte[1:]
    X = np.array([min_temp[1:], max_temp[1:], regen[1:] + giessmenge[1:], bodenfeuchte[:-1]])
    X = np.transpose(X)
    model = Model().fit(X, y)
    print(model.score(X, y))
    onx = to_onnx(model, X)
    with open('modell.onnx', 'wb') as f:
        f.write(onx.SerializeToString())
```

Abbildung 14. Programm, dass das Modell trainiert

## 4.2 Teil 2. Berechnung von optimaler Gießmenge

Abb. (15) zeigt den Algorithmus, welcher die optimale Gießmenge berechnet.

```
def optimale_giessmenge(min_temp, max_temp, regen, heute_bodenfeuchte):
    if heute_bodenfeuchte > TROCKEN_NFK:
        return 0
    else:
        for i in range(0, 20):
            morgen_bodenfeuchte = predict(np.array([[min_temp[0], max_temp[0],
                                                      regen[0] + i, round(hoye_bodenfeuchte)]]))
            if morgen_bodenfeuchte < TROCKEN_NFK:
                continue
            ueermorgen_bodenfeuchte = predict(np.array([[min_temp[1], max_temp[1],
                                                          regen[1] + 0, round(morgen_bodenfeuchte)]]))
            if ueermorgen_bodenfeuchte < TROCKEN_NFK:
                continue
            tag3_bodenfeuchte = predict(
                np.array([[min_temp[2], max_temp[2], regen[2] + 0, round(ueermorgen_bodenfeuchte)]]))
            if tag3_bodenfeuchte >= TROCKEN_NFK:
                return i
        return 20
```

Abbildung 15. Algorithmus für die optimale Gießmenge

In diesem Programm wird auch die Wettervorhersage einbezogen. Dadurch wird nicht umsonst gewässert, wenn es in den nächsten Tagen schon massenweise regnet.

Hier ist das Programm nochmals in Worten erklärt: Falls die heutige Bodenfeuchte schon über einer bestimmten Bodenfeuchte (30 % nFK) entspricht, wird nicht gegossen. Sonst wird für 0l bis 20l pro qm geprüft, wie viel man heute gießen muss, damit der Boden für drei Tage genug feucht bleibt. Ob das Wasser reicht, berechnet das KI-Modell wobei die Wetterprognose berücksichtigt wird. Die Wetterdaten von den nächsten drei Tagen werden von einer Webseite abgelesen und sind aufgelistet [6].

## 4.3 Wasserersparnis

Um zu prüfen, dass das Modell auch wirklich Wasser spart, wurden die Wetterdaten für die Sommermonaten (Juli und August) der vergangenen Jahre hergenommen [7] und einmal mit dem Gießmodell und einmal ohne getestet. Es wurde in Prozent ausgerechnet, wie viel gespart oder verschwendet wurde.

Jahr	2019	2020	2021	2022
Optimierte Gießmenge	70 l/qm	101 l/qm	27 l/qm	238 l/qm
Verwendete Gießmenge	140 l/qm	180 l/qm	80 l/qm	340 l/qm
Gespartes Wasser (in %)	50 %	43 %	66 %	30 %

Abbildung 16. Wassersparpotenzial der letzten 4 Jahre

In der Tabelle in Abb. (16) kann man erkennen wie viel Wasser im jeweiligen Jahr pro Quadratmeter gespart wurde. Um die Auswertungen besser zu analysieren, wurden sie in Diagramme dargestellt (siehe Abb. (17)). 2019 vom 23 bis zum 57 Tag hat das optimierende Modell aufgehört zu gießen, da es berechnet hat, dass es in den nächsten Tagen regnet. Das nicht optimierende Modell hat immer noch weiter gegossen, wobei das auch zu Sauerstoffmangel der Pflanzen führen kann.

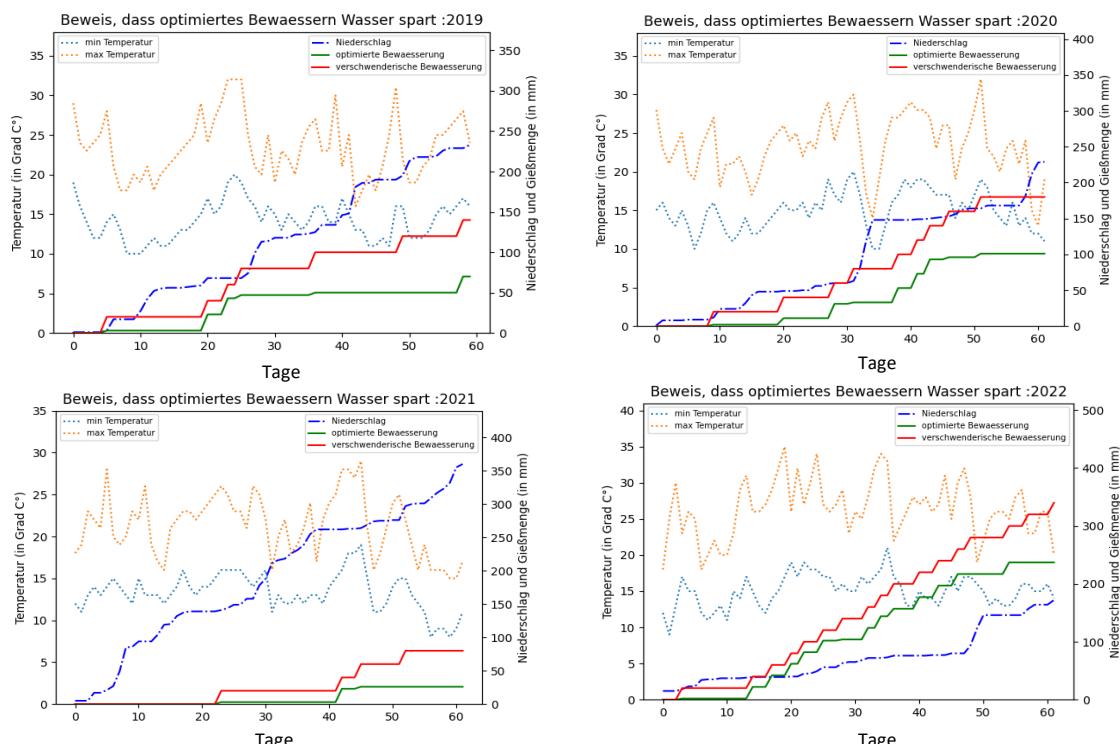


Abbildung 17. Auswertungen von den Experimenten in Diagramme gefasst

# 5 AI Smart-Watering App

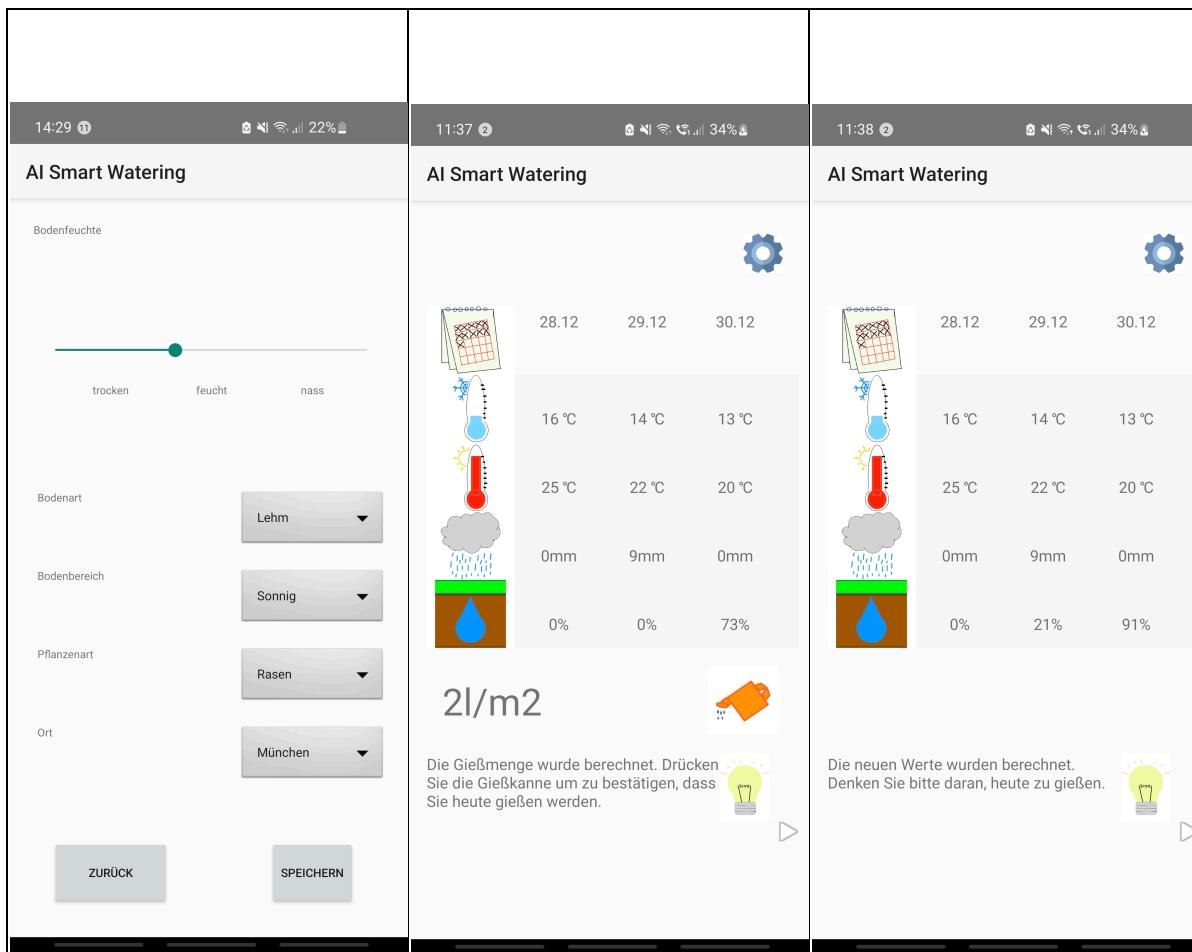


Abbildung 18. Screenshots von der entwickelten App. Der linke Screenshot zeigt die Einstellung beim ersten Start. Der mittlere zeigt die berechnete, optimale Gießmenge. Der rechte zeigt die neue gerechnete Bodenfeuchte, nachdem der Benutzer die Gießmenge bestimmt hat.

Damit auch andere Leute optimal gießen können, haben wir unsere Lösung in eine benutzerfreundliche Android App eingebaut. Dafür haben wir recherchiert, wie man Apps entwickelt und eine einfache App programmiert. Als nächstes wird die Entwicklung der App beschrieben.

## 5.1 Einstellungsseite

Beim ersten Öffnen der App wird eine Seite mit Einstellungen gezeigt, damit der Benutzer seinen Standort, die Bodenart in seinem Garten, die Sonnenlage, die Pflanzenart und die aktuelle Bodenfeuchte angeben kann (siehe Abb. (18)). Die App beinhaltet momentan nur das Modell einer bestimmten Bodenart, aber man kann andere Modelle entwickeln und hinzufügen.

Durch diese Informationen kann die benötigte Gießmenge abgeschätzt werden. Danach muss der Benutzer auf „Speichern“ drücken, um die angegebenen Werte abzuspeichern.

Die Werte werden in eine App spezifische SharedPreferences-Objekt gespeichert. Ein SharedPreferences-Objekt verweist auf eine Datei mit Schlüssel-Wert-Paaren und stellt einfache Methoden zum Lesen und Schreiben dieser bereit.

## 5.2 Hauptseite

Das Modell wurde in Python programmiert (wie es in der „Modell Entwicklung“ beschrieben wurde) und danach in die Android App integriert. Das wurde mit einer onnx Datei gemacht. Diese hat ihre eigene Sprache und kann verschiedene Programmiersprachen übersetzen.

Die eingegebene Einstellungswerte und die aktuelle Bodenfeuchte werden von dem SharedPreferences-Objekt abgelesen. Darüber hinaus wird die Wettervorhersage von einem Dienstleister als JSON-Datei bezogen (siehe Abb. (19)) und von der App analysiert und gezeigt.

```
{"city_name":"Munich","country_code":"DE","data":[{"app_max_temp":-3.4,"app_min_temp":-10.3,"clouds":79,"clouds_hi":29,"clouds_low":7,"clouds_mid":62,"datetime":"2024-01-16","dewpt":-7.1,"high_temp":-0.5,"low_temp":-4.6,"max_dhi":null,"max_temp":-0.5,"min_temp":-4.3,"moon_phase":0.482194,"moon_phase_lunation":0.19,"moonrise_ts":1705398424,"moonset_ts":1705362268,"ozone":323.1,"pop":20,"precip":0.00048828125,"pres":946.5,"rh":69,"slp":1011.8,"snow":0,"snow_depth":63,"sunrise_ts":1705388297,"sunset_ts":1705420239,"temp":-2.2,"ts":1705359660,"uv":0,"valid_date":"2024-01-16","vis":24.128,"weather":{"icon":"c04d","description":"Overcast clouds","code":804}, "wind_cdir": "SSE", "wind_cdir_full": "south-southeast", "wind_dir": 162, "wind_gust_spd": 3.4, "wind_spd": 1.4}, {"app_max_temp":3.4,"app_min_temp":-5.9,"clouds":97,"clouds_hi":85,"clouds_low":79,"clouds_mid":95,"datetime":"2024-01-17","dewpt":-1.4,"high_temp":7.5,"low_temp":5.3,"max_dhi":null,"max_temp":6.1,"min_temp":-2.9,"moon_phase":0.598092,"moon_phase_lunation":0.22,"moonrise_ts":1705485820,"moonset_ts":1705453433,"ozone":334.7,"pop":7,"precip":6.8,"pres":935.1,"rh":82,"slp":998.9,"snow":21,"snow_depth":84,"sunrise_ts":1705474654,"sunset_ts":1705506725,"temp":1.3,"ts":1705446660,"uv":0.6,"valid_date":"2024-01-17","vis":18.596,"weather":{"icon":"f01d","description":"Freezing rain","code":511}, "wind_cdir": "SSE", "wind_cdir_full": "south-southeast", "wind_dir": 159, "wind_gust_spd": 5.2, "wind_spd": 2.7}, {"app_max_temp":7.7,"app_min_temp":-3.7,"clouds":99,"clouds_hi":52,"clouds_low":62,"clouds_mid":89,"datetime":"2024-01-18","dewpt":2.1,"high_temp":7.7,"low_temp":-2.6,"max_dhi":null,"max_temp":7.7,"min_temp":-0.1,"moon_phase":0.598092,"moon_phase_lunation":0.25,"moonrise_ts":1705573281,"moonset_ts":1705539883,"ozone":354.1,"pop":88,"precip":8.400001,"pres":934.5,"rh":81,"slp":997.5,"snow":3,"snow_depth":87,"sunrise_ts":1705561009,"sunset_ts":1705593211,"temp":5.2,"ts":1705532460,"uv":0.7,"valid_date":"2024-01-18","vis":17.289,"weather":{"icon":"s04d","description":"Mix snow/rain","code":610}, "wind_cdir": "W", "wind_cdir_full": "west", "wind_dir": 263, "wind_gust_spd": 7.6, "wind_spd": 3.7}, {"app_max_temp":-2.8,"app_min_temp":-6.8,"clouds":23,"clouds_hi":23,"clouds_low":48,"clouds_mid":24,"datetime":"2024-01-19","dewpt":-4.9,"high_temp":-0.3,"low_temp":-6.3,"max_dhi":null,"max_temp":-0.3,"min_temp":-3.7,"moon_phase":0.705812,"moon_phase_lunation":0.29,"moonrise_ts":1705660907,"moonset_ts":1705630974,"ozone":412.1,"pop":70,"precip":0.8,"pres":956.8,"rh":80,"slp":1022.4,"snow":8,"snow_depth":95,"sunrise_ts":1705647361,"sunset_ts":1705679698,"temp":-1.8,"ts":1705618860,"uv":1.9,"valid_date":"2024-01-19","vis":24.128,"weather":{"icon":"s01d","description":"Light snow","code":600}, "wind_cdir": "W", "wind_cdir_full": "west", "wind_dir": 272, "wind_gust_spd": 5.2, "wind_spd": 2.6}, {"app_max_temp":-2.8,"app_min_temp":-3.8,"clouds":41,"clouds_hi":1,"clouds_low":4,"clouds_mid":0,"datetime":"2024-01-20","dewpt":-8.4,"high_temp":-0.2,"low_temp":-7.3,"max_dhi":null,"max_temp":-0.2,"min_temp":-6.3,"moon_phase":0.80054,"moon_phase_lunation":0.32,"moonrise_ts":1705748822,"moonsat_ts":1705721976,"ozone":374.5,"pop":10,"precip":965.2,"rh":72,"slp":1030.2,"snow":0,"snow_depth":95,"sunrise_ts":1705733711,"sunset_ts":1705766187,"temp":-3.8,"ts":1705785260,"uv":2,"valid_date":"2024-01-20","vis":24.128,"weather":{"icon":"c01d","description":"Broken clouds","code":803}, "wind_cdir": "SE", "wind_cdir_full": "southeast", "wind_dir": 144, "wind_gust_spd": 3.7, "wind_spd": 1.7}, {"app_max_temp":1.2,"app_min_temp":-10.2,"clouds":37,"clouds_hi":28,"clouds_low":3,"clouds_mid":22,"datetime":"2024-01-21","dewpt":-8.2,"high_temp":-3.2,"low_temp":-1.6,"max_dhi":null,"max_temp":-1.6,"min_temp":-7.3,"moon_phase":0.878729,"moon_phase_lunation":0.36,"moonrise_ts":1705837180,"moonset_ts":1705812614,"ozone":356.9,"pop":8,"precip":967.3,"rh":66,"slp":1032.3,"snow":0,"snow_depth":95,"sunrise_ts":1705820058,"sunset_ts":1705852676,"temp":-2.4,"ts":170591660,"uv":2.4,"valid_date":"2024-01-21","vis":24.128,"weather":{"icon":"c02d","description":"Scattered clouds","code":802}, "wind_cdir": "S", "wind_cdir_full": "south", "wind_dir": 170, "wind_gust_spd": 3.6, "wind_spd": 1.8}, {"app_max_temp":3.9,"app_min_temp":-4.8,"clouds":56,"clouds_hi":71,"clouds_low":45,"clouds_mid":53,"datetime":"2024-01-22","dewpt":-4.4,"high_temp":6.7,"low_temp":4.5,"max_dhi":null,"max_temp":6.7,"min_temp":-1.6,"moon_phase":0.938051,"moon_phase_lunation":0.39,"moonrise_ts":1705926136,"moonset_ts":1705902633,"ozone":383,"pop":25,"precip":70000005,"pres":959,"rh":60,"slp":1022.4,"snow":0,"snow_depth":95,"sunrise_ts":1705906492,"sunset_ts":1705939166,"temp":2.7,"ts":1705878060,"uv":2.4,"valid_date":"2024-01-22","vis":22.448,"weather":{"icon":"c03d","description":"Broken clouds","code":803}, "wind_cdir": "SW", "wind_cdir_full": "southwest", "wind_dir": 221, "wind_gust_spd": 7, "wind_spd": 3.3}], "lat": "48.13743", "lon": "11.57549", "state_code": "02", "timezone": "Europe/Berlin"}]
```

Abbildung 19 Wetterdaten von der Website

Auf der Hauptseite der App ist oben eine Tabelle abgebildet, welche die Niederschlagsmenge und die maximal und minimal Temperaturen des heutigen Tages und auch die von den nächsten zwei Tagen anzeigt (siehe Abb. (18)). Die Wetterdaten werden von einer Website abgelesen. Unter der Tabelle steht die aktuelle nFK und die nötige Gießmenge. Durch Betätigen der Gießkanne rechts bestätigt man, dass man die angezeigte Gießmenge gegossen hat (siehe Abb. (18)). Danach berechnet das Modell die neue Bodenfeuchte und zeigt sie auf der Einstellungsseite an (siehe Abb. (18)). Zur Erleichterung der Benutzung der App ist ganz unten ein Tipp-Kasten vorhanden, welcher verschiedene Hinweise und Garten-Tipps zur Verfügung stellt.

Damit man die Einstellungen auch ändern kann, wurde ein Knopf programmiert, womit man von der Hauptseite zur Einstellungsseite wechseln kann. Falls man glaubt, dass die Bodenfeuchte falsch berechnet wurde, kann der Benutzer diese ändern.

## 6 Zusammenfassung

Wasser ist eine kostbare Ressource und wird öfters verschwendet, zum Beispiel beim übertriebenen Gießen. Deswegen wurde ein KI-Modell trainiert, welches den optimalen Gießbetrag berechnet. Dadurch wird viel Wasser sowie auch Müll gespart. Das Modell schätzt als erstes die Bodenfeuchte. Danach überprüft es, ob das Wasser im Boden für die nächsten drei Tage ausreichen wird. Falls nicht, wird empfohlen, die angezeigte Menge pro Quadratmeter zu gießen. Die Lösung wurde in eine Android App umgesetzt, womit die Benutzer optimal gießen können.

### 6.1 Probleme

- Das KI-Modell wurde in Python programmiert und musste dann in Android übertragen werden.  
Lösung: durch eine onnx Datei wurde das Modell transportiert
- Einige Regression Modelle sind nicht geeignet, zum Beispiel die lineare Regression. Sie führt zu negativen Ergebnissen, wobei sie eigentlich nicht unter 0 gelangen dürfte. Dann haben wir XGBoost verwendet, doch dieses Modell kann nicht in eine onnx Datei umgewandelt werden.  
Lösung: Gradient Boosten Algorithmus
- Da einige Fälle in unserer Datensammlung nicht vorhanden sind, kommt es manchmal zu Ungenauigkeiten.  
Beispiel: Im Sommer treffen niedrige Temperaturen und Regen oft zusammen. Als das Modell im Winter getestet wurde, wurde entdeckt, dass die Bodenfeuchte auch bei trockenen Tagen steigt.  
Lösung: wurde nicht gelöst, aber es wurde beobachtet, dass es durch ausreichend gesammelte Daten möglich ist

### 6.2 Weitere Schritte

Ein KI-Modell ist immer nur so genau wie die gegebenen Daten. Desto mehr Daten, desto genauere Ergebnisse hat das Modell. Da wir nicht das ganze Jahr Daten gesammelt haben ist unser Modell ein wenig ungenau.

Durch das Messen im Sommer, hat unser Modell Schwierigkeiten im Winter. Das Modell hat keine Beispiele von kalten Temperaturen, dadurch entstehen ungenauere Ergebnisse. Wenn hohe Temperaturen herrschen und es nicht regnet, glaubt das Modell, dass die Bodenfeuchte steigen wird. Das passiert, da es im Sommer fast immer regnet, wenn die Temperaturen steigen (Modell weiß nicht, ob die Bodenfeuchte wegen dem Regen oder den

hohen Temperaturen steigt). Um das zu verbessern könnte man ein ganzes Jahr lang messen und das Modell würde lernen, dass die Bodenfeuchte ohne Wasserquelle nicht steigen kann.

Noch eine Idee wäre mehrmals in einem Tag zu messen, da es unterschiedliche Wirkungen der Bodenfeuchte am Morgen und am Abend gibt. Man könnte auch mehr Features betrachten z.B. den Wind und die Luftfeuchtigkeit. Wir haben ein Gießmodell für schluffigen Boden programmiert, aber man kann auch für andere Boden- und Pflanzenarten Modelle programmieren.

Neben den echten Daten könnte man auch fiktive Daten hinzufügen, die dem Modell zeigen, dass zum Beispiel die Bodenfeuchte nicht unter 0 sinken kann.

Auch wo man misst, spielt eine große Rolle. Es ist am besten waagerecht und ungefähr 10 cm tief zu messen.

Unser Bodenfeuchtesensor zeigt keine Zahlen über 50 an, wodurch das Modell noch ein bisschen ungenauer wird. Doch starke Regenniederschläge stellen die Ungenauigkeit wieder richtig. Durch den Regenniederschlägen steigt die Bodenfeuchte bis zu 100 % nFK, wodurch unser Modell genau weiß wie viel Bodenfeuchte der Boden haben müsste.

## 7 Quellenangaben

[1]

<https://www.fraunhofer.de/content/dam/zv/de/forschungsthemen/gesundheit/Wasserspezial/Forschung%20f%C3%BCr%20die%20Wassernutzung%20von%20morgen.pdf>

- [2] <https://www.gardena.com/at/c/entdecke/garteninspiration/gartenmagazin/zehn-goldene-regeln-der-bewaesserung>
- [3] <https://www.umweltbundesamt.de/daten/private-haushalte-konsum/wohnen/wassernutzung-pivater-haushalte#direkte-und-indirekte-wassernutzung>
- [4] <https://sensebox.de/>
- [5] <https://scikit-learn.org/>
- [6]  
<https://api.weatherapi.com/v1/forecast.json?q=48.8567,2.3508&key=a70c759bb2224e458f6120745240903&days=3>
- [7] <https://api.worldweatheronline.com/> (nur mit einem Konto erreichbar)

Abbildung 1. <https://www.bluecommunity.ch/wasser-weltweit>

Abbildung 2: <https://nachhaltig-sein.info/privatpersonen-nachhaltigkeit/der-pro-kopf-wasserverbrauch-in-deutschland-eine-ubersicht>

Abbildungen 3 und 4 – wurden selbst erstellt.

Abbildung 5:

<https://www.bgr.bund.de/DE/Themen/Boden/Produkte/Karten/Downloads/BUEK1000.pdf?blob=publicationFile&v=2>

Abbildungen 6 bis 19 – wurden selbst erstellt.

## 8 Unterstützungsleistungen

Unser Projekt wurde durch Christoph Bürgis und Jonas Röhrl unterstützt.