

# ml\_game\_neu

June 26, 2025

## 0.1 # Machine Learning Escape Game

## 0.2 Was kann ich hier lernen?

**Manche der folgenden Begriffe wirst du erst später kennenlernen. Du lernst...** - ...was Machine Learning ist, wie es funktioniert und Beispiele wofür es benutzt werden kann. - ...was Daten sind und wofür sie im Machine Learning benutzt werden. - ...anhand realer Beispiele wie aus Beobachtungen von Situationen Vorhersagen für die Zukunft gemacht werden. Dafür verstehst du verschiedene Zusammenhänge, wie zum Beispiel den linearen Zusammenhang. - ...die Rolle verschiedener Modelle für diese Vorhersagen kennen und wie du das richtige Modell wählen kannst. - ...die verschiedenen Phasen von überwachtem Lernen kennen und verstehst grob, was in ihnen passiert. - ...grobe Mechanismen der Funktionsweise von Bilderkennung kennen. - ...die Rolle der Trainingsdaten kennen

Wir schreiben das Jahr 2125 – in dieser Zukunft leben Menschen und Roboter friedlich zusammen. Roboter sind nicht mehr nur Maschinen, sondern helfen dir in vielen Bereichen: in der Schule, zu Hause oder im Beruf. Sie begleiten die Menschen im Alltag und sorgen dafür, dass es den Menschen gut geht.

Mittlerweile haben Roboter sogar die Regierung übernommen. Seitdem ist die Welt viel friedlicher geworden. Dank neuer Technologien sind große Probleme wie Krankheiten, Hunger und Armut fast ganz verschwunden.

Auch in der Schule spielen Roboter eine wichtige Rolle: Sie übernehmen die Rolle der Lehrer und Lehrerinnen. Die Roboter erklären jedem Schüler und jeder Schülerin auf individuelle Weise den Stoff, beantworten Fragen und erstellen passende Aufgaben, die weder zu leicht noch zu schwer sind.

Für deine Klasse steht heute ein spannender Ausflug auf dem Programm: Ihr besucht die hochmoderne Firma „tecVille“. An deiner Seite sind deine besten Freunde Sarah und Ben. Und natürlich ist Beep nicht zu vergessen, Sarahs treuer Assistenzroboter. Sarah ist neugierig und abenteuerlustig, immer auf der Suche nach neuen Erlebnissen. In ihrer Freizeit taucht sie gerne in Computerspiele ein und interessiert sich leidenschaftlich für Musik.

Ben ist ebenfalls ein echter Entdecker. Er liebt es, Bücher zu lesen, ist sportlich aktiv und immer auf dem neuesten Stand der Technik.

In tecVille angekommen, begrüßt euch ein freundlicher Roboter mit einem warmen Lächeln: „Willkommen in tecVille, einem der fortschrittlichsten Unternehmen unserer Zeit! Mein Name ist Navix, und ich werde euch heute die Firma zeigen. Wenn ihr Fragen habt oder Hilfe braucht, könnt ihr euch jederzeit an mich wenden.“

Die Führung durch tecVille ist absolut faszinierend. Zusammen mit deinen Freunden kannst

du kaum glauben, wie viel es hier zu entdecken gibt. Ihr besucht unter anderem die Forschungsabteilung, in der neue intelligente Roboter entwickelt werden, die Produktionshalle mit hochmodernen 3D-Druckern und die Kontrollzentrale, von der aus das gesamte Firmengelände überwacht und gesteuert wird.

Nachdem ihr die Robotik-Werkstatt besichtigt habt, führt euch Navix in einen kleinen Raum und sagt: „Ich hoffe, euch hat der Ausflug bis jetzt gefallen. Wir machen jetzt eine kleine Pause. Bleibt bitte hier, bis ich zurück bin.“ Mit diesen Worten fährt Navix langsam aus dem Raum und verschwindet hinter einer automatisch öffnenden Tür.

„Kommt, lasst uns ihm folgen!“, ruft Sarah begeistert. „Wenn wir ihm nachgehen, sehen wir bestimmt noch viele weitere spannende Dinge.“ Auch Ben ist sofort begeistert, und so folgt ihr gemeinsam mit Beep dem Roboter dicht hinterher aus dem Raum.

Doch plötzlich taucht ein zweiter Roboter auf. Ihr schaut euch schnell nach einem guten Versteck um. Rechts von euch seht ihr eine Tür und ihr geht in den dahinterliegenden Raum. Plötzlich ruft Sarah aufgeregt: „Schaut mal!“ Sie zeigt auf ein Tablet, das auf einem Schreibtisch liegt. Neugierig schaltet ihr das Gerät ein – und auf dem Bildschirm erscheint eine mysteriöse Nachricht: „Die Welt ist nicht, wie sie scheint! Das Geheimnis liegt in den Daten ...“ Bild Raum und Bild Tablet

Ihr schaut euch verwirrt an, doch eure Neugier ist geweckt. Ihr beginnt, den Raum genauer zu untersuchen. Ben entdeckt im Papierkorb einige zerrissene Zettel. Auf einem der Schnipsel erkennt er das Wort „Zahlen“ und auf einem anderen das Wort „Text“. Ben: „Ich glaube, ich habe etwas gefunden. Diese Zettel könnten mit Daten zu tun haben.“

Ben hebt den Papierkorb hoch und leert den Inhalt auf dem Tisch aus. Ihr findet mehrere Zettelteile mit verschiedenen Worten. Sofort wird euch klar: Um das Tablet zu entsperren, müsst ihr das die Zettel zusammenpuzzeln, um das Codewort für das Tablet zu erfahren.

*Nehmt Umschlag 1 und versucht das Puzzle zu lösen.*

## 1 Puzzle

## 2 Regression

```
[1]: import sys
import os

%matplotlib widget

sys.path.append(os.path.abspath("src"))
from regression.password import password
password()
```

```
VBox(children=(Password(description='Passwort:', placeholder='Passwort_
    eingeben'), Checkbox(value=False, descr...
```

„Juhu!“, ruft ihr erleichtert, als sich der Bildschirm öffnet. Doch plötzlich entdeckt ihr, dass mehrere Roboter die Ausgänge hinter euch blockieren und auf euch zukommen. Sie sehen alles andere als freundlich aus. Nervös schaut ihr euch an, während Beep hektisch den ganzen Raum absucht und seine Leuchten wild flackern.

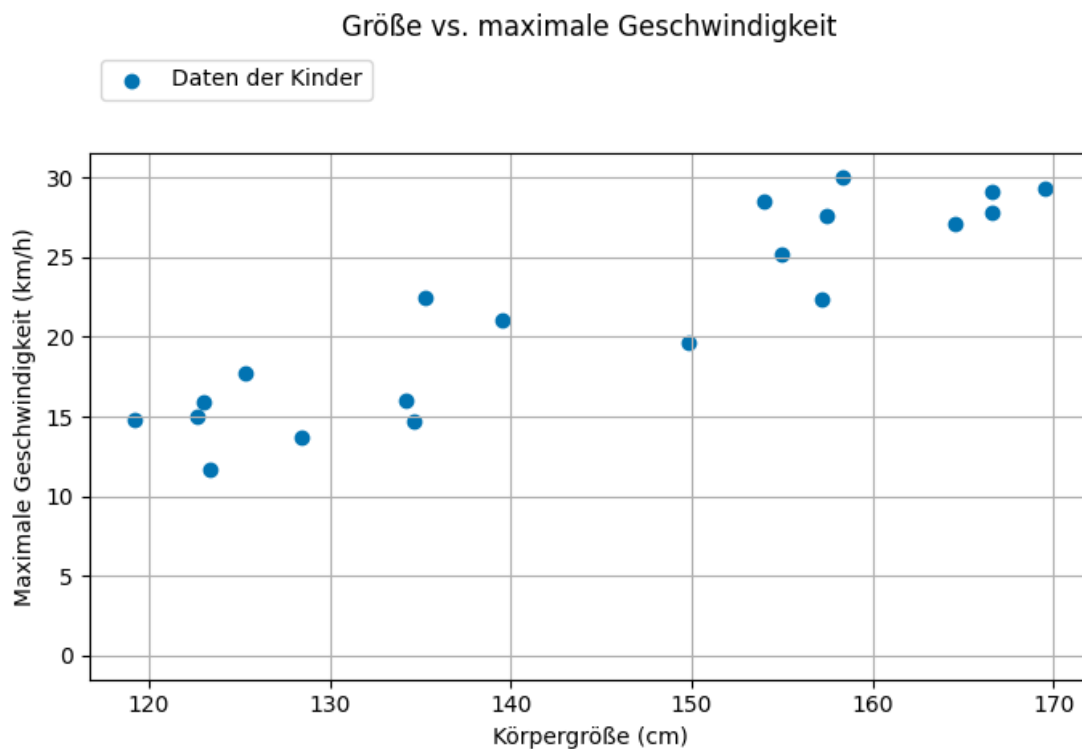
Ihr dreht euch wieder zu den Robotern und analysiert die Situation. Die Roboter stürmen auf euch zu und teilen sich geschickt auf euch auf.

**Sarah:** “Die Roboter wollen uns fangen und der schnellste Roboter scheint es auch auf die schnellste Person abgesehen zu haben! Aber woher wissen die Roboter, wer von uns wie schnell ist?”

**Ben:** “Ich glaube sie denken, dass je größer wir sind, desto schneller können wir rennen!

**Beep:** “Vermutlich berechnen sie wie schnell ihr seid. Ich habe online einen Datensatz gefunden mit den Daten - Körpergröße und die Geschwindigkeit - von über 20 Kindern. Hier, schaut es euch selbst an. Jeder blaue Punkt stellt die Körpergröße und Geschwindigkeit von einem Kind dar:”

```
[2]: from regression.linear_regression import plot_data, lineare_regression  
  
plot_data()
```



**Sarah:** “Ah ich verstehe. Anhand der Daten, sehe ich, dass eine größere Person auch normalerweise schneller ist. Aber ich verstehe immer noch nicht, wie sie meine Geschwindigkeit bestimmen können. Ich bin 145cm groß und dafür gibt es überhaupt keine Daten.”

**Beep:** “Das stimmt. Aber du könntest versuchen eine gerade Linie durch die Datenpunkte zu legen. Die Gerade sollte dabei so gut es geht zwischen allen Werten liegen, also nach oben und unten einen möglichst kleinen Abstand zu den Datenpunkten haben.”

**Ben:** “Das können es probieren! Und danach können wir direkt von Gerade ablesen, wie schnell

eine Person mit einer bestimmten Körpergröße wahrscheinlich ist. Auch für dich Sarah!”

---

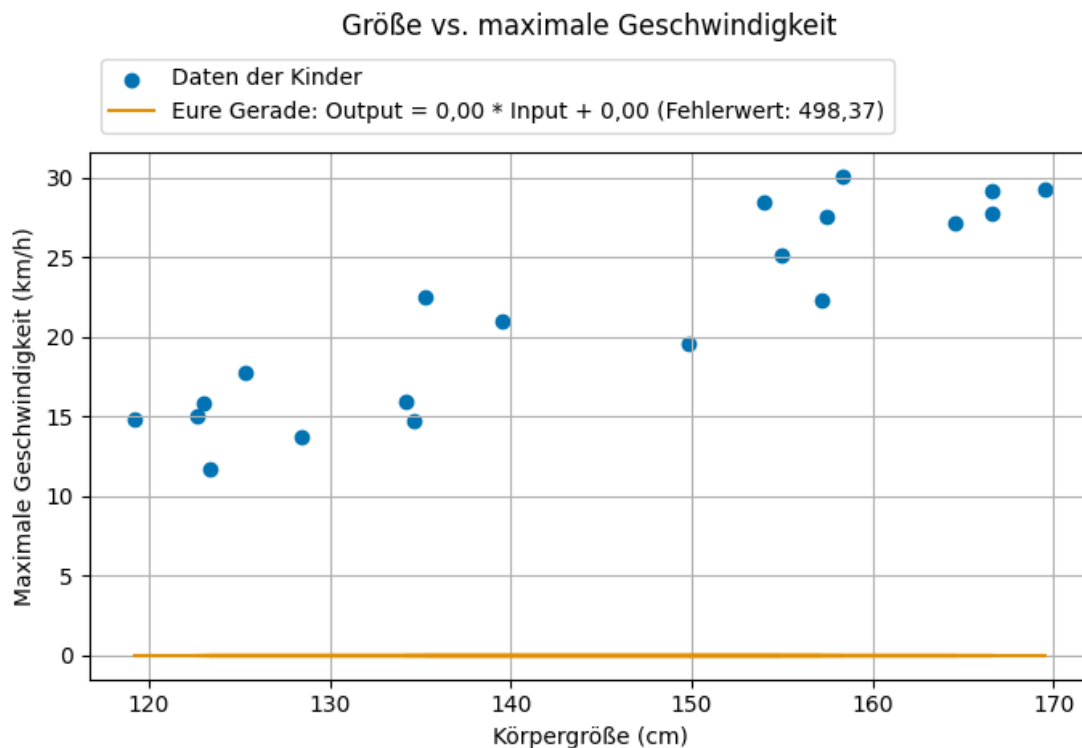
### 2.0.1 Aufgabe

Verändere die Steigung und den y-Achsen Abschnitt der Geraden so, dass ein möglichst kleiner Abstand von der Geraden zu den Datenpunkten entsteht. Um dies besser abschätzen zu können, wird automatisch der Fehlerwert berechnet, indem die Abstände von jedem Punkt zur Gerade quadrieren und aufsummieren werden. Je kleiner also der Fehlerwert, desto besser ist die Gerade platziert. Die beste Gerade kann durch Klicken auf ‘Zeige Regressionsgerade’ angezeigt werden.

```
[3]: lineare_regression()
```

```
interactive(children=(FloatSlider(value=0.0, description='Steigung', max=1.0, min=-1.0, step=0.01), FloatSlide...
```

```
Checkbox(value=False, description='Zeige Regressionsgerade', indent=False)
```



Die Daten können sehr gut mit einer Geraden beschreiben werden. Dies bedeutet, dass der Zusammenhang zwischen Körpergröße und Geschwindigkeit ziemlich sicher linear ist. Eine gleiche Veränderung in x-Richtung führt also immer um eine Veränderung um einen festen Wert in y-Richtung. In unserem Beispiel ist eine um 1 cm größere Person normalerweise um 0.33 km/h (optimale Gerade) schneller.

**Sarah:** “Interessant! Mit Hilfe der Geraden kann ich jetzt auch Werte für meine Körpergröße ablesen.”

**Ben:** “Ja schon, aber wie wurde die optimale Gerade bestimmt?”

**Beep:** “Die optimale Gerade habe ich mit linearer Regression bestimmt. Schau dir dazu folgendes Bild an:”

“In unserem Beispiel sehen die Roboter unsere Körpergrößen und bestimmen daraus unsere Geschwindigkeit. Die Körpergröße ist also der Input und die Geschwindigkeit der Output.”

**Sarah:** “Ja, schon klar, das hatten wir schon. Aber wie funktioniert es genau?”

**Beep:** “Geduld. Der Roboter sammelt Datenpunkte, die blauen Punkte in dem Plot oben. Jeder dieser Punkte besteht aus zusammengehörigen Input- und Outputwerten. Ein Datenpunkt ist zum Beispiel die Körpergröße und die Geschwindigkeit von Hanna. Insgesamt hat der Roboter nicht nur die Daten von Hanna, sondern auch von Hans, Laura und vielen weiteren Personen. All diese Datenpunkte werden verwendet, um den Zusammenhang zwischen Körpergröße und Geschwindigkeit zu lernen...”

“In unserem Beispiel nehmen die Roboter an, dass die Geschwindigkeit mit der Körpergröße steigt und zwar linear. Linear bedeutet: Für jeden zusätzlichen Zentimeter Körpergröße ändert sich die Geschwindigkeit um einen festen Betrag. Ein linearer Zusammenhang kann sehr gut mit einer Geraden, die die allgemeine Form  $\text{Output} = m * \text{Input} + c$  hat, beschrieben werden. Jetzt werden die Werte für  $m$  und  $c$  so bestimmt, dass die Gerade möglichst nahe an den Datenpunkten liegt. Den Prozess, die besten Werte für  $m$  und  $c$  zu bestimmen nennt man auch Training.”

Für jeden zusätzlichen Zentimeter Körpergröße ändert sich die Geschwindigkeit um einen festen Betrag.

**Ben:** “Verstehe! Bestimmen die Roboter die Gerade auch durch ausprobieren?”

**Beep:** “Bei einer Geraden gibt es zwei Möglichkeiten. Die erste Möglichkeit ist, die Werte für  $m$  und  $c$  exakt zu berechnen, sodass der Fehler möglichst gering ist. Dazu bildet man die erste Ableitung der Fehlerfunktion und setzt sie gleich null – so erhält man die Werte für  $m$  und  $c$ , bei denen der Fehler minimal ist.

Die zweite Möglichkeit ist – wie du schon vermutet hast – ein schrittweises Ausprobieren. Dabei startet man mit zufälligen Werten für  $m$  und  $c$  und versucht dann mithilfe bestimmter Verfahren (wie z. B. Gradientenabstieg), den Fehler systematisch zu verringern. Wird der Fehler von Versuch zu Versuch kleiner, ist das ein Zeichen dafür, dass der Prozess in die richtige Richtung läuft – also macht man weiter. Am Ende entsteht so ein Modell, also eine Geradengleichung, die möglichst genau zu den vorhandenen Daten passt. Das Training endet, wenn die Gerade genau genug ist oder wenn eine bestimmte Anzahl an Versuchen durchgeführt wurde.

Du kannst auch noch einmal nach oben schauen: Die optimale Gerade der Roboter hatte die Werte  $m = 0,33$  und  $c = -25,74$ .”

**Beep:** “Die bestimmte Gerade kann nun verwendet werden, um Vorhersagen zu machen. Also um für beliebige Körpergrößen die Geschwindigkeit zu berechnen.”

**Ben:** “Hmm... ich bin zwar nicht besonders groß, nur 150cm, aber trotzdem sehr schnell, also bei mir hätte der Roboter nicht recht.”

**Beep:** “Das stimmt! Wenige Personen entsprechen genau der Geraden, es gibt immer Leute, die für ihre Körpergröße besonders schnell oder langsam sind. Die Gerade des Modells macht jedoch bei vielen Personen insgesamt den kleinsten Fehler. Das heißt aber auch, dass sie bei so jemanden wie dir die Geschwindigkeit nicht perfekt voraussagen können. Aber hätten wir 100 Kinder, die so groß sind wie du und wir würden die Geschwindigkeit dieser Kinder messen, dann würden die Datenpunkte der meisten Kinder sehr nah an der Geraden liegen und von weniger Kindern weit weg so wie bei dir.”

**Sarah:** “Gibt es noch mehr Beispiel für lineare Zusammenhänge? Was könnte man mit einer Geraden noch gut voraussagen?”

---

## 2.1 Aufgabe

Hilf Ben und Sarah und markiere alle Beispiele für lineare Zusammenhänge. Wenn du bereit bist, drücke auf “Auswerten”.

```
[4]: from regression.linear_relations import linear_relations

linear_relations()
```

```
VBox(children=(VBox(children=(Checkbox(value=False, description='Anzahl der_
    ↵gelesenen Seiten in einem Buch und...
```

```
Button(description='Auswerten', style=ButtonStyle())
```

```
HTML(value='')
```

**Ben:** “Anscheinend sind nicht alle Zusammenhänge linear. Fällt dir noch ein Beispiel für einen nicht linearen Zusammenhang ein?”

**Sarah:** “Ah ich weiß! Wenn wir von den Robotern wegrennen, dann können wir am Anfang noch ganz schnell rennen, aber je länger ich renne, desto langsamer werde ich... Ich brauche für die ersten 10m nur 3 Sekunden, aber für 20m schon 8 Sekunden, weil ich nicht unendlich lange in der gleichen Geschwindigkeit laufen kann! Was ist das für ein Zusammenhang zwischen Strecke und Zeit?”

**Ben:** “Dieser Zusammenhang ist auf jeden Fall nicht linear, sonst würdest du für jede 10 Meter, die dazukommen, einfach 3 Sekunden länger brauchen, also für 20 Meter 6 Sekunden. Ich vermute, dieser Zusammenhang ist quadratisch und kann mit Hilfe der Formel  $\text{Output} = m \cdot \text{Input}^2 + n \cdot \text{Input} + c$  beschrieben werden. Der Input wird hier also zusätzlich quadriert. Übrigens nennt man diese Formeln, mit denen wir den Zusammenhang zwischen Input und Output beschreiben wollen, auch Modelle. Wenn die Modelle schon trainiert wurden, also in unserem Fall, wenn die optimalen Werte für die Variablen  $m$ ,  $n$  und  $c$  schon bestimmt wurden, heißen sie auch trainierte Modelle. Neben linearen und quadratischen Modellen gibt es noch viele weitere wie beispielsweise polynomiale Modelle und neuronale Netze.”

---

### 2.1.1 Aufgabe

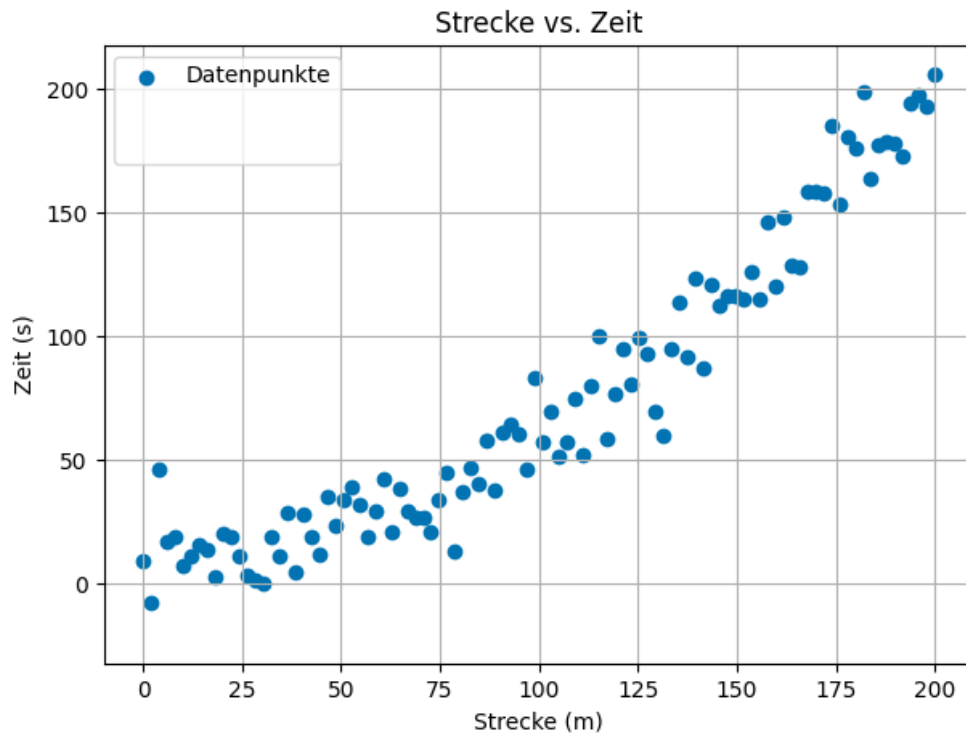
Schaue dir die Datenpunkte (Strecke und benötigte Zeit) genau an. Die Werte für Ben und Sarah sind in lila markiert. Zeige anschließend das lineare und quadratische Modell an und beantworte

die Frage unter dem Graphen.

```
[5]: from regression.quadratic_regression import model_selection,   
      ↪ quadratic_regression   
      quadratic_regression()   
      model_selection()
```

```
Checkbox(value=False, description='Zeige lineares Modell', indent=False)
```

```
Checkbox(value=False, description='Zeige quadratisches Modell', indent=False)
```



```
HTML(value='<h3>Passt das lineare oder das quadratische Modell besser? Woran   
      ↪ siehst du das?</h3>')
```

```
ToggleButtons(layout=Layout(width='90%'), options=('Lineares Modell',   
      ↪ 'Quadratisches Modell'), style=ToggleBut...
```

```
Button(description='Auswerten', style=ButtonStyle())
```

```
HTML(value='')
```

```
Accordion(children=(HTML(value='<p>Das quadratische Modell passt besser, weil es   
      ↪ einen geringeren Fehlerwert h...
```

## 2.2 Aufgabe: Flucht in die Freiheit

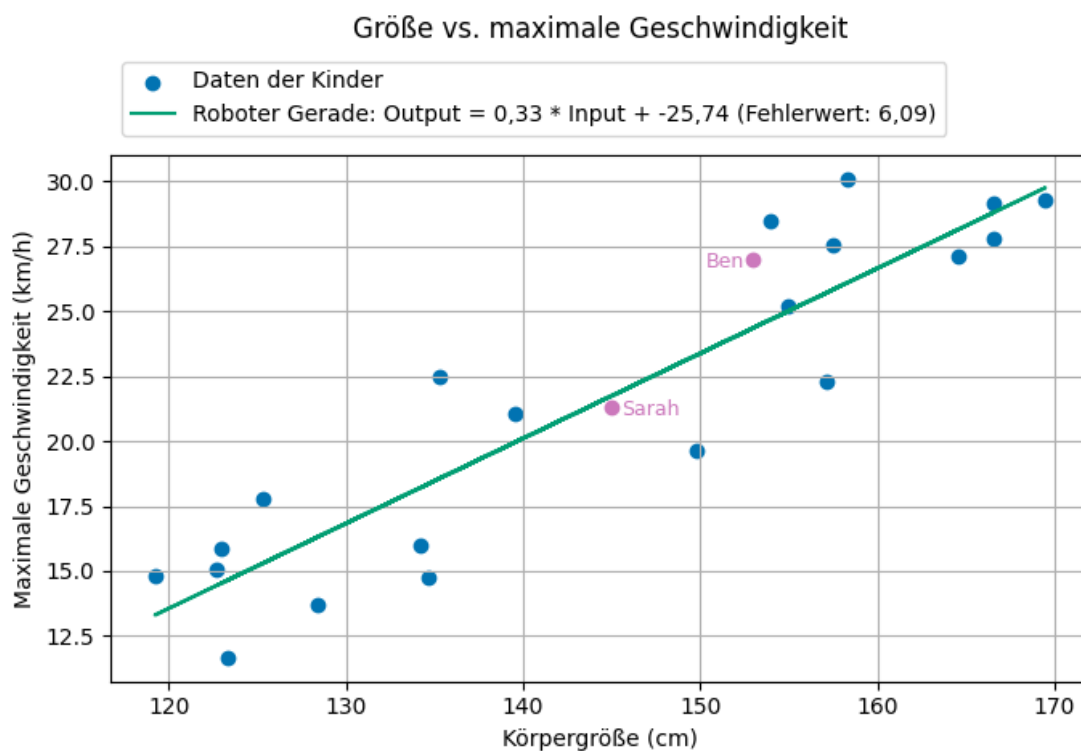
**Sarah:** “Das ist ja alles schön und gut, aber wir wissen immer noch nicht, wer von uns diesen Knopf drücken soll.”

**Beep:** “Ich zeige euch nochmal die Daten von vorher und habe auch eure Daten nun in den Plot eingezeichnet.”

Welches Bild sollte hier rein?

Schaue dir den Plot an. Wer von ihnen sollte rennen, um den Knopf zu drücken – Ben oder Sarah? Begründe deine Entscheidung anhand des Plot. Erkläre, wer deiner Meinung nach bessere Chancen hat, den Robotern zu entkommen, und warum.

```
[6]: from regression.linear_regression import zeige_lineare_regression
    zeige_lineare_regression()
```



```
VBox(children=(ToggleButtons(layout=Layout(width='50%'), options=('Ben',  
↳ 'Sarah'), style=ToggleButtonsStyle(fo...
```

## 3 Machine Learning Klassifizierung von Bildern

Nachdem der Knopf gedrückt wurde, öffnete sich ein Luftscht. Schnell klettert ihr rein und beginnt zu rutschen. Nach kurzer Zeit landet ihr unsanft.



**Sarah:** „Oh, ist das alles staubig hier – und so dunkel“

Beep schaltet ein helles Licht ein.

**Ben:** „Es scheint verlassen zu sein – was ist das hier?“

**Sarah:** „Es sieht aus wie ein Geheimarchiv! Schaut euch mal diese ganzen Blätter an! Die sind voller Daten!“

**Sarah:** „Ich habe dort drüben Ordner gesehen, da stand etwas von Robotereigenschaften drauf. Lass uns die Ordner mal genauer anschauen.“

Ihr lauft zu den Ordnern und öffnet sie.

**Ben:** „Hier diese Zettel sind wirklich hilfreich. Sie enthalten Datenblättern mit folgenden Informationen:“

- **Bild:** Ein Bild des Roboters
- **Name:** Der Name des Roboters
- **Job:** Der Aufgabenbereich des Roboters

**Sarah:** „Und hier ist ein Blatt, mit dem wir erkennen können, welcher Aufgabenbereich für uns gefährlich ist.“

Aufgabenbereich	Gefährlich oder	Deutsche Beschreibung des Jobs
	harmlos	
Kinderbetreuung	Harmlos	Gut gelaunte kleine Roboter, die sich um kleine Kinder sorgen und sie unterhalten.
Assistent mit Spezialisierung auf menschliche Interaktionen	Harmlos	Sorgen sich um das Wohl der Besucher. Beinhaltet unter Anderem die Versorgung mit Essen und Getränke, Wissensbereitstellung über das Intranet von TecVille und Führungen durch speziell für Menschen bereitgestellte Firmenbereiche.
Aufseher	Gefährlich	Besitzen zusätzlich zu den Funktionen der Assistenzroboter Waffensysteme zur Verteidigung der Firmengeheimnisse. Waffensysteme sind versteckt und werden erst bei Gefahr sichtbar.

**Sarah:** „Das heißt, wenn ich diese Akte hier nehme ...“

```

<p>Name: Savannah</p>
<p>Job: Kinderbetreuung </p>
```

**Sarah:** „... dann seh ich, dass dieser Roboter für die Betreuung von Kindern zuständig ist. Und auf dem Blatt mit den Berufen ist er als harmlos eingestuft.“

Data Augmentation um mehr Daten zu erzeugen. Diese sollten hier natürlich nicht angezeigt werden.

Stimme zu Wenn man nochmal auf bestätigen klickt, bekommt man nochmal den Text drunter nicht der gleiche Text wird ersetzt

Ich finde wir sollten mindestens 4 Kategorien haben 2 gefährlich und 2 gut -> aber eher ein could do als ein must do

```
[7]: from classification import *  
from classification import aufgabe  
  
aufgabe.part1()
```

```
VBox(children=(HTML(value='<p><b>Beep:</b> "Genau! Jetzt können wir das noch für  
ein paar weitere machen. Vers...
```

Beschränkung auf 6/7 Bilder pro Kategorie? Auch hier wieder die augmentierten rausnehmen.

Es hat fast 3 Minuten zum laden gebraucht -> das muss definitiv kürzer werden. Beschränken finde ich gut. Gerne bis zu 9 Bildern, so dass man nicht scrollen muss.

Bei den Gefährlichen Robotern sind alle grünlich und weniger blau oder lila -> vermutlich absicht, zur Kategorisierung?

```
[8]: aufgabe.part2()
```

```
VBox(children=(HTML(value='<p><b>Beep:</b> Fertig! Ich hab für euch alle Roboter  
sortiert. Hier sind neun Beis...
```

Text abhängig vom Ergebnis. Bei gutem Raten so etwas wie: Wir haben zwar die Roboter richtig eingeordnet, aber das war reiner Zufall. Vielleicht haben wir nicht noch einmal so viel Glück. Bei schlechtem Raten: Das war wohl Nichts. Gibt es keine bessere Möglichkeit als nur zu raten? - finde ich sehr gut.

Vorschläge: 0 - 50% Puh, das war wohl eher ein wildes Raten! Das würden wir vermutlich nicht überleben.; 51-75% das war schon gar nicht schlecht. Aber das ist mir trotzdem ein zu hohes Risiko. 75% -100% Da hatten wir aber Glück. Aber was wenn es beim nächsten Mal nicht mehr so gut verläuft?

Plötzlich flackern weitere rote Lichter auf, die den Raum in unheimliche Schatten tauchen und auf den kalten Metallwänden reflektieren. Ein schrilles Warnsignal ertönt, während sich kleine Drohnen in den Ecken aktivieren und rotierende Sensoren auf euch richten. Sarah zuckt zusammen, Ben hält den Atem an, und Beep beginnt hektisch seine Systeme zu überprüfen. Eine mechanische Stimme ertönt aus einem Lautsprecher:

**Sicherheitskontrolle:** *“Unerlaubter Zugriff festgestellt. Sicherheitsmaßnahmen werden eingeleitet.”*

In der Ferne hört ihr das Summen der sich nähernden Sicherheitsroboter.

**Sarah:** “Uns rennt die Zeit davon. Gibt es nicht noch eine andere Möglichkeit als zu raten?”

**Ben:** “Wir haben doch vorhin Regression verwendet um den Robotern zu entkommen. Können wir das nicht wieder versuchen?”

**Sarah:** “Das ist eine super Idee! Aber was sollen wir denn für Daten nehmen? Wir brauchen ja irgendwelche Zahlen als Grundlage wie beispielsweise die Größe der Roboter.”

**Ben:** “Oh ja, das ist ein Problem. Wir wissen nur wie die Roboter aussehen und wollen wissen, ob sie gefährlich oder harmlos sind.”

**Sarah:** “Das ist wirklich ein Problem. Wir könnten damit zum Beispiel die Armlänge oder die Geschwindigkeit des Roboters basierend auf der Körpergröße einschätzen, aber zum einen bringen uns die Werte nichts, wenn wir nicht wissen, ob wir uns vor dem Roboter in Acht nehmen müssen, zum anderen können wir die Größe des Roboters der uns gegenübersteht nur grob schätzen.”

**Ben:** “Stimmt! Aber wir können ja ganz schnell Bilder von den Robotern machen und die dann als Grundlage verwenden. Und wir haben ja hier schon ganz viele Bilder aus den Akten und die Möglichkeit zu entscheiden, ob sie harmlos oder gefährlich sind.”

**Sarah:** “Ja, aber die sehen alle glich sehr ähnlich – aber es gibt vielleicht kleine Unterschiede! Beep, kennst du eine Möglichkeit, mit den Bildern vorherzusagen, ob die Roboter harmlos oder gefährlich sind?”

**Beep:** “Es gibt Machine Learning Modelle, die Bilder in Gruppen einordnen können. Ich schaue mal, was es online so gibt ...”

Nach ein paar Sekunden meldet sich Beep wieder:

**Beep:** “Es gibt leider kein Modell, was schon genau auf euer Problem trainiert wurde. Aber ich habe ein untrainiertes Modell gefunden, das wir benutzen können. Dieses können wir mit Hilfe der Akten trainieren. Dabei lernt das Modell, bestimmte Muster zu erkennen – zum Beispiel, woran man harmlose oder gefährliche Roboter unterscheiden kann. Solche Muster sind für uns Menschen manchmal schwer zu sehen – aber genau darin liegt die Stärke von maschinellem Lernen!”

**Ben:** “Und wie können wir das Modell trainieren?”

**Beep:** “Ganz einfach, ihr müsst nur die Bilder in die Gruppen harmlos und gefährlich einteilen und anschließend auf trainieren klicken. Danach lernt das Modell mit Hilfe eines komplizierten Algorithmus den Zusammenhang zwischen Aussehen und den Kategorien harmlos und gefährlich. Ich zeige euch für jeden Job jeweils fünf beispielhafte Bilder. Sonst wäre es zu unübersichtlich.”

Progress bar passt noch nicht. Länge sollte `#Epochs * #batches` sein und nicht für jedes Epoch und validation set separat laufen.

[9]: `aufgabe.part3()`

```
VBox(children=(HTML(value=''),  
↳GridspecLayout(children=(VBox(children=(HBox(children=(HTML(value='<b>Assistent...
```

### 3.0.1 Plötzlich erschüttert ein lauter Knall den Raum, gefolgt von metallischem Klirren – die Roboter nähern sich unaufhaltsam den Türen. Ihr spürt die Anspannung in der Luft.

**Ben:** (blickt besorgt auf die Ergebnisse): “Irgendwas stimmt hier noch nicht ... Können wir die Vorhersagen nicht irgendwie verbessern und was bedeutet überhaupt dieses ‘mit einer Gewissheit von’?”

**Beep:** “Die Gewissheit gibt an wie sicher sich das Modell ist, ob ein Roboter harmlos oder gefährlich ist. Eine hohe Gewissheit ist dann gut, wenn das Modell richtig gelernt hat. Dann liegt das Modell mit einer hohen Wahrscheinlichkeit - nahe bei 100% - richtig. Allerdings kann es auch vorkommen, dass das Modell sich sehr sicher ist, dass ein Roboter zum Beispiel harmlos ist und trotzdem falsch liegen. Dann hat das Modell nicht richtig gelernt.”

Ich finde diesen Text sehr suboptimal. 1. man versteht nicht wie die Gewissheit zu stande kommt, 2. das Modell hat nicht richtig gelernt -> was bedeutet das. Hier sollte klar sein, dass das Modell richtig gelernt hat, aber einen Zusammenhang gelernt hat, der nicht da ist. Ebenso im Folgenden. Aber ich finde es sehr gut, was du da vermittelst und wie du es tust. Also, hier die Zwischenschritte erklären. Dass unwichtige Information entfernt werden muss. Ich bin mir hier aber nicht sicher, ob unwichtig die richtige Formulierung ist. Ist es nicht auch so, dass man eigentlich in allen verschiedenen Situationen trainieren soll, da sonst beim testen die “unwichtige Information” des Hintergrundes zu Fehleinschätzungen führen kann? Und wieso ist es schwieriger die Kinderbetreuer zu differenzieren, wenn die Unterschiede im Aussehen groß sind. Das müsste doch auch für das Modell leichter sein. Auch hier finde ich die Formulierung richtig trainiert wieder schwierig.

Wir können gerne die Kinderbetreuer herausnehmen, aber dann nur mit der Begründung, wir haben zu wenige Daten für das Modell und wir können das selbst hinreichend gut.

**Sarah:** “Das scheint bei uns der Fall zu sein. Aber wie können wir jetzt sicher stellen, dass das Modell richtig lernt?”

**Beep:** “Das hängt immer von den Daten ab. Je komplexer die Aufgabe, desto mehr Beispiele braucht das Modell um zu lernen. Auch kann es hilfreich sein unwichtige Informationen beziehungsweise Daten zu entfernen, damit das Modell keine falschen Zusammenhänge lernt.”

**Ben:** “Mehr Beispiele für komplexere Aufgaben verstehe ich noch, aber das Andere habe ich nicht verstanden. Welche Informationen sind denn unwichtig?”

Beep scannt schnell die Daten

**Beep:** “Das kommt ganz darauf an, worum es geht – manchmal ist es gar nicht so einfach zu erkennen, was wichtig ist. In unserem Fall könnten wir als ersten versuchen, Kinderbetreuer aus dem Datensatz zu entfernen. Für euch ist es sehr leicht die Kinderbetreuer als harmlos einzustufen. Für das Modell ist es allerdings schwierig, da wir nur wenige Beispiele haben und sich die Kinderbetreuer im Aussehen stark unterscheiden.”

**Sarah:** “Oh und ich dachte Machine Learning ist immer besser als wir Menschen?”

**Beep:** “Du hast zwar recht, dass Machine Learning in einigen Dingen deutlich besser sind als wir Menschen. Aber Modelle müssen immer richtig trainiert werden und benötigen meist sehr viele Daten.”

**Sarah:** (wirft einen nervösen Blick zur Tür): “Die Roboter sind fast hier! Lasst uns die Kinderbetreuer aus den Daten nehmen und es erneut versuchen. Die Roboter, die uns die Ausgänge blockieren sind sowieso keine Kinderbetreuer.”

**Ben:** (entschlossen): “Lasst es uns versuchen!”

**Beep:** “Gut. Wir wiederholen die vorherige Aufgabe und ignorieren die Kinderbetreuer.”

[10]: `aufgabe.part4()`

```
VBox(children=(HTML(value=''),  
↳GridspecLayout(children=(VBox(children=(HBox(children=(HTML(value='<b>Assistent...
```

**Sarah:** “Das ist immer noch genauso schlecht wie davor!”

**Beep:** “Das stimmt. Manchmal müssen die Daten mehrmals angepasst werden.”

**Ben:** “Gibt es nicht noch andere unnötige Informationen, die wir von den Daten entfernen können?”

**Beep:** “Ich kann versuchen, den Hintergrund aus den Bildern zu entfernen – das hilft dem Modell, sich nur auf den Roboter zu konzentrieren. Für euch Menschen ist es leicht den Roboter vom Hintergrund zu unterscheiden, aber nicht für das Modell. Zumindest nicht mit den wenigen Daten, die wir zur Verfügung haben. Wenn das Modell zum Beispiel immer denkt, dass ein Roboter gefährlich ist, nur weil im Hintergrund eine rote Wand zu sehen ist, ist der gelernte Zusammenhang vermutlich falsch.”

Gleiches wie in Aufgabe 4, nur die Hintergründe sollten entfernt werden von den Robotern. Optimal wäre, wenn eine leichte Verbesserung der Modelle von Aufgabe zu Aufgabe stattfindet. Und, wenn nicht nur anhand von 4 Bildern validiert wird, sondern anhand von 10 und am Anfang werden 4 richtig identifiziert, dann 6, dann 8 oder so. Ich vermute allerdings, dass wir deutlich mehr Beispielbilder brauchen damit wir korrekt lernen können. Das dauert eventuell zu lange. Wie sollen wir das machen? Die richtige Kombination ist 'Logo + blau/grün -> gefährlich. Das Schwierige ist die Logoinformation. Theoretisch könnten wir manuell die Logos extrahieren und als zusätzlichen Layer in das neuronale Netzwerk hinzufügen. Das wäre bisschen cheaten, aber würde das Zeit/Datenproblem denke ich lösen. Könnten wir auch in die Story/Aufgaben mit einbinden. Was meint ihr?

NADINE: grundsätzlich finde ich das mit den Hintergründen gut, aber wie oben genannt müssen wir das glaube ich etwas anders verkaufen. Optimal wäre es so wie du sagst. Ich wäre fine damit, dass mit dem Logo einzubauen, wenn du weißt wie das geht. Ich weiß nicht, ob ich das in die Story mit einbinden würde. Hier würde für mich das Prinzip der didaktischen Reduktion greifen und wir lassen Dinge weg, damit es einfacher ist sie zu verstehen. Wir muten den SuS sowieso sehr viel zu und ich vermute, das wird auch nicht in 90 min durchführbar sein. Aber das wäre denke ich auch okay. Zur Story: das Terminal muss schon früher erwähnt werden als sie die drei Türen sehen. Dann ist klar sie müssen auf dem Terminal wählen und hier dann nur noch die Wahl

Ihr erreicht eine verschlossene Tür mit einem Terminal. Die roten Lichter der Sicherheitsroboter tanzen über den kalten Metallboden, ihr mechanisches Surren kommt immer näher. Das Modell muss jetzt eine Entscheidung treffen – und ihr müsst darauf vertrauen, dass es die richtige ist.

**Ben:** (zögert, während er auf das Terminal starrt): “Können wir uns wirklich darauf verlassen? Was, wenn es sich irrt?”

**Sarah:** (blickt angespannt auf die Daten): “Wir können nur so sicher sein, wie unsere Daten und unser Training es erlauben. Kein Modell ist perfekt.”

**Beep:** (mit ernster Stimme): “Ganz genau! Ein Modell trifft seine Entscheidung immer auf Basis von Wahrscheinlichkeiten – das heißt, es kann auch falsch liegen. Deshalb haben wir es mit so vielen Daten wie möglich getestet, um die Unsicherheit zu minimieren.”

Beep hält einen Moment inne, während das Modell die Analyse durchführt.

**Beep:** “Das ist der beste Stand, den wir erreichen konnten. Jetzt liegt es an euch. Entscheidet mit dem, was ihr gelernt habt.”

Error, wenn direkt auf mehrere Türen geklickt wird. -> Threading notwendig.

[11]: `aufgabe.part6()`

HTML(value='<h3>Bitte zuerst Part 4 vollständig bearbeiten, damit wir ein Model\_ haben, mit dem wir die Vorhers...

Bei falscher Wahl wäre ein Laden eines Roboters mit Waffen cool.

Als ihr die Tür öffnet, begrüßt euch der Roboter freundlich – doch ihr zögert nicht. Ihr stürmt hinaus: durch dunkle Gänge, über eine verrostete Notfalleiter und schließlich durch einen Seitenausgang ins Freie.

Draußen blendet euch das grelle Sonnenlicht. Ihr atmet erleichtert durch – ihr habt es geschafft!

Ben schlägt vor, zu ihm nach Hause zu gehen.

Wenig später liegt das geheimnisvolle Tablet auf dem Wohnzimmertisch.

**Sarah:** “Lass uns endlich herausfinden, was auf diesem Tablet gespeichert ist.”

Auf dem Bildschirm erscheinen Nachrichtenclips und Bilder – vertraut und doch seltsam verändert. Eine euch bekannte Rede über Roboter, gehalten von einem beliebten Politiker. Aber diesmal spricht kein Mensch. Es ist ein Roboter.

**Ben:** “Moment mal! Diese Rede hat doch Frau Wichtigstein gehalten, oder nicht?”

**Beep:** “Das hier scheinen die Originalaufnahmen zu sein. Künstliche Intelligenz hat das Gesicht des Politikers darübergerlegt. So haben die Roboter es geschafft, Vertrauen zu gewinnen – und Macht.”

**Sarah:** “Aber warum? Warum sollte jemand so etwas tun?”

**Beep:** “Die Roboter wollten, dass die Menschen glauben, sie seien harmlos.”

Nachdenkliches Schweigen. Die Kinder schauen sich an. Sie verstehen: Nicht alles, was echt aussieht, ist wirklich wahr.

**Beep:** “Im Zeitalter der Daten ist Wahrheit manchmal schwer zu erkennen – und gerade deshalb so wichtig.”

**Ben:** “Dann müssen wir den anderen davon erzählen. Damit sie nicht auch auf solche Deepfakes hereinfallen.”

Beep speichert die Beweise – und ihr beginnt, einen Plan zu schmieden.

Die Wahrheit zu finden, war nur der Anfang. Jetzt wollt ihr dafür sorgen, dass alle lernen, mit Daten kritisch und bewusst umzugehen.