



# **Demystifying AI – Wie lernen Maschinen eigentlich**

## **CCC Camp 2023**

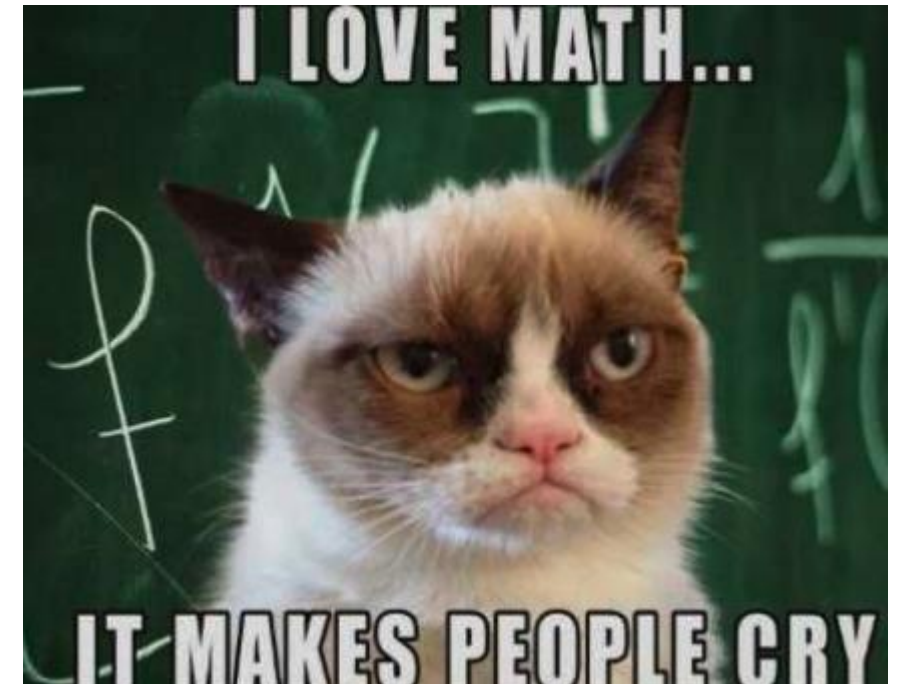
Jannes Quer



**Jannes Quer**

- Arbeitet bei bei SRLabs
- 2 Jahre Postdoc im Bereich ML in der Bioinformatik und der Moleküldynamik
- Erfahrung als Data Scientist bei einer Big Four Beratung
- Promotion an der Freien Universität Berlin im Bereich Moleküldynamik
- Studium Angewandte Mathematik in Lübeck mit dem Schwerpunkt Bildgebung

■ [jannes@srlabs.de](mailto:jannes@srlabs.de)



# Durch den aktuellen Hype wird KI überschätzt

## Buzzword

Alles muss jetzt KI haben.  
Sogar eine einfache Kaffeemaschine.

## Komplexität

Interesse der Hersteller,  
Produkt als komplex darzustellen.

## Bias

Wir sehen nur die beeindruckenden Beispiele  
auf Twitter

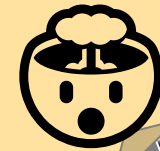
## Sprache

„Es kann sprechen, es muss intelligent sein!“

***Ein schöner Tag beginnt  
mit einem guten Kaffee!***

### “Künstliche Intelligenz:

Die am häufigsten bezogenen  
Produkte ordnen sich  
automatisch auf den  
Plätzen 1-4 an.”



# Neuronale Netze versuchen das Gehirn durch angewandte Statistik zu imitieren

## Künstliche Intelligenz (KI)

Computer imitieren menschliches Denken und Verhalten.

## Machine Learning (ML)

Statistische Algorithmen ermöglichen Implementierung von KI durch Lernen aus Daten.

## Deep Learning

Teilbereich des ML, der neuronale Netze verwendet.

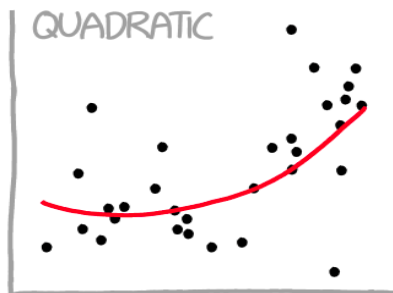
# Generative KI hat viel mehr Spielraum, um gute Antworten zu produzieren

## Spezialisierte KI

Spezialisierte KI kann nur Klassifikations- oder Regressionsprobleme lösen. Sie wird für eine einzelne Aufgabe trainiert und kann auch nur diese lösen.



Klassifikation

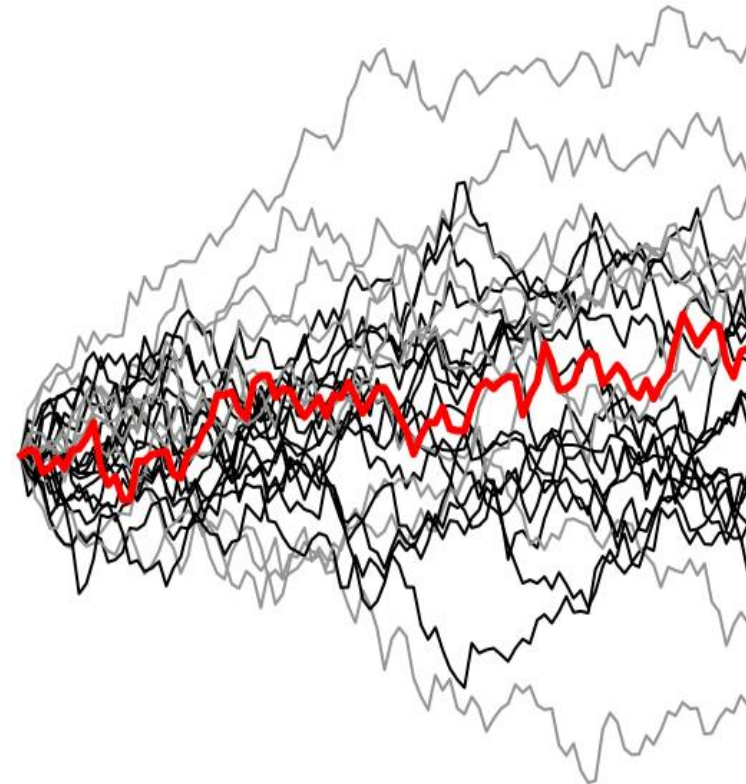


"I WANTED A CURVED  
LINE, SO I MADE ONE  
WITH MATH."

Regression

## Generative KI

Generative KI hat sehr viele Möglichkeiten ein gutes und überzeugendes Ergebnis zu erzielen.

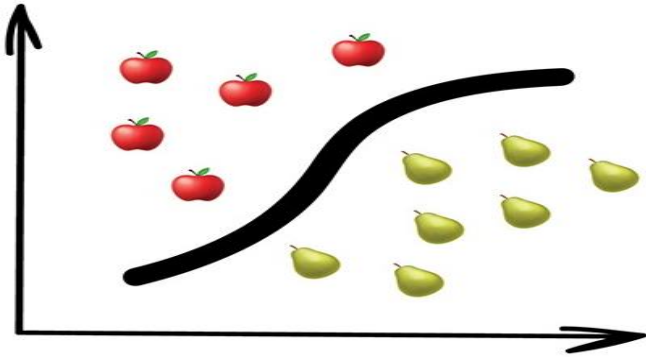


Gute und  
überzeugende  
Antwort



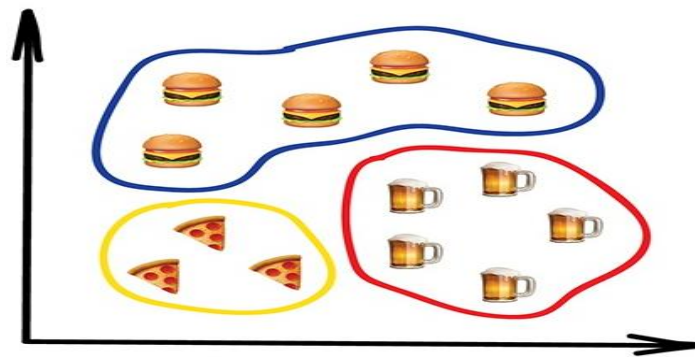
# Es gibt drei unterschiedliche Arten des Machine Learnings

## Supervised learning



- Label sind bekannt
- Ziel ist das Mapping von Input zu Output zu lernen
- Beispiel: Klassifikation, Regression

## Unsupervised learning



- Label sind unbekannt
- Ziel ist es Muster und Gemeinsamkeiten in Daten zu lernen
- Beispiel: Recommender Systems

## Reinforcement learning



- Daten werden erzeugt
- Ziel ist es ein Verhalten in einer Umgebung zu lernen
- Beispiel: Alpha Go

**Ziel des Lernens ist immer eine mathematische Funktion zu lernen, die von Input auf Output abbildet.**

# Lernen wird durch Modelloptimierung erreicht

## Daten

- Es braucht eine große **Menge an Daten**
- Die **Daten** müssen **gut gelabelt** sein, da die Daten die Regeln enthalten
- **Feature Engineering**

## Training

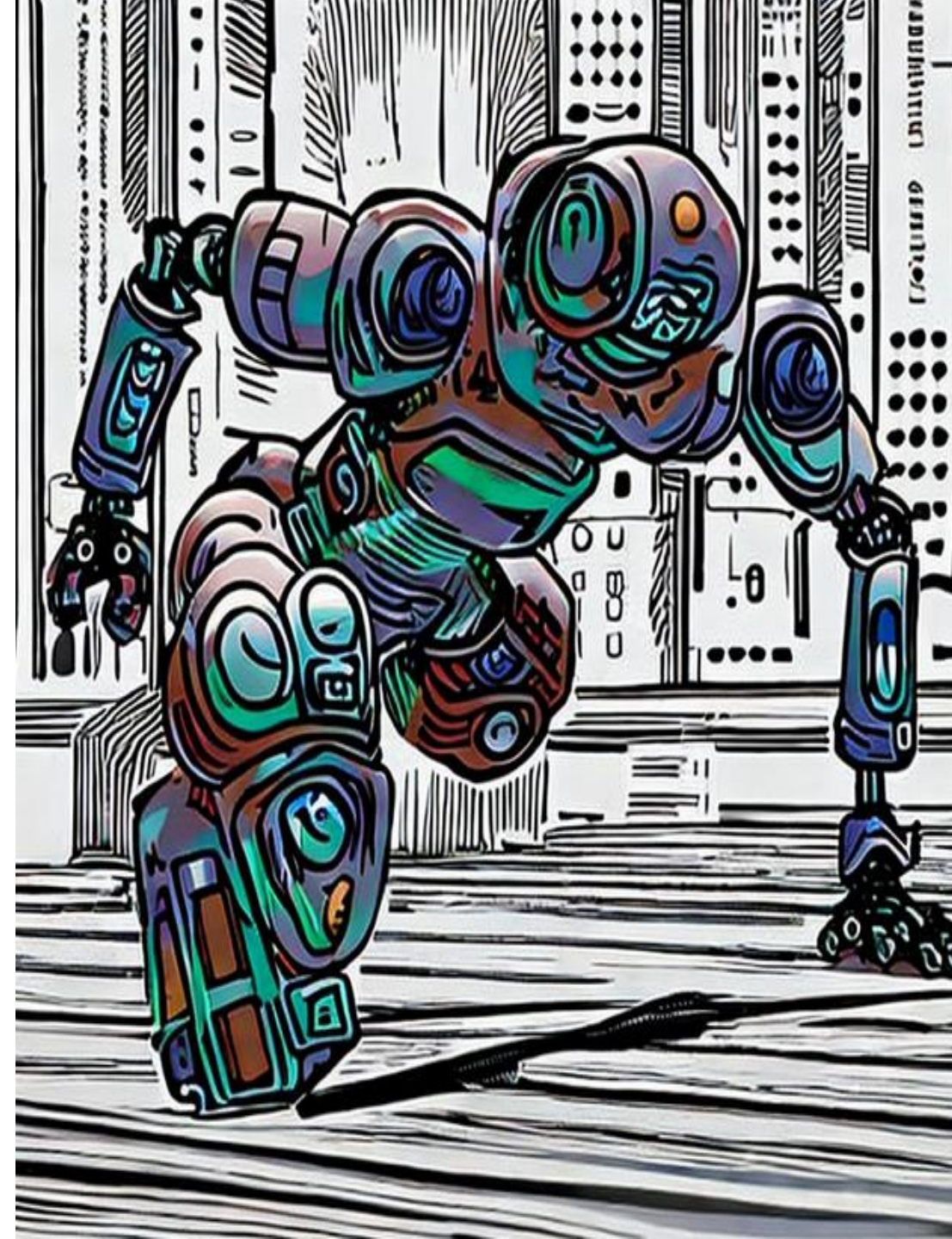
- Wahl der **Architektur**, der **Lossfunktion** und des **Optimierungsalgorithmus**
- **Mathematische Optimierung**, um die bestmöglichen Gewichte zu finden, die einen kleinen Trainingsfehler erreichen

## Generalisierung

Identifizieren der **Parameter**, die

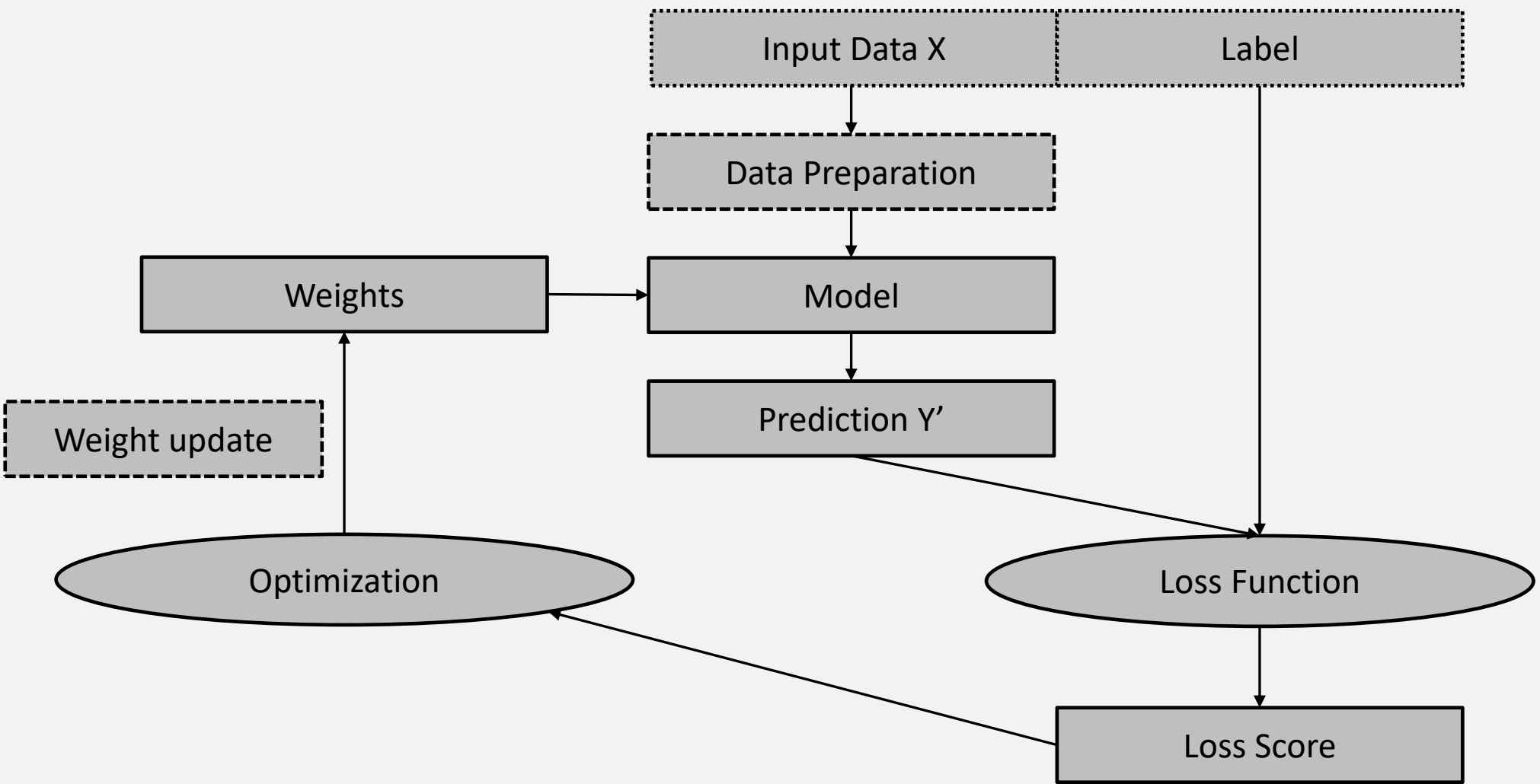
1. eine gute **Anpassung an die Trainingsdaten** und
2. eine **gute Verallgemeinerung** ermöglichen

Die **Messung** der Generalisierung erfolgt mit **ungenutzten Trainingsdaten**



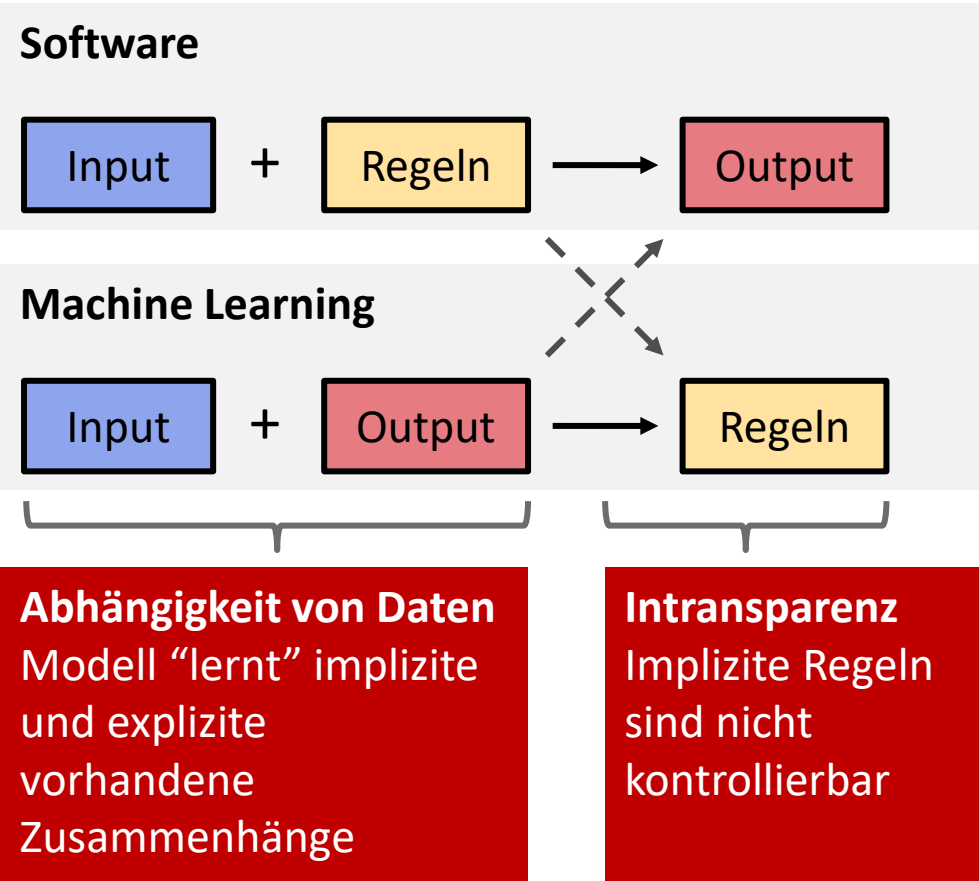
# Lernen ist ein iterativer Algorithmus

## Beispiel: Supervised Learning





# Machine Learning lernt Regeln aus Daten

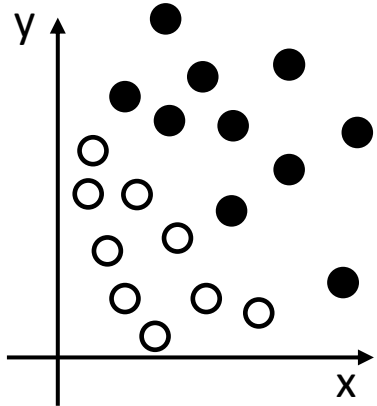


# Daten Vorverarbeitung ist der essentielle Teil des Machine Learnings

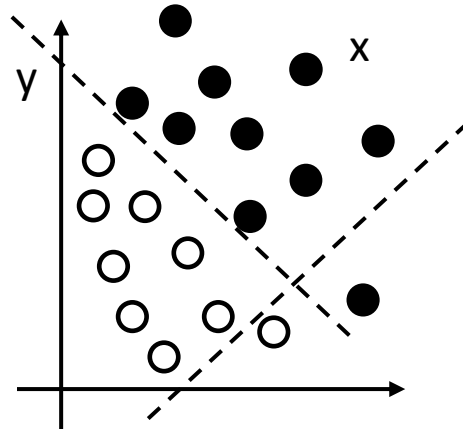
## Daten Vorverarbeitung

- Vorverarbeitung beinhaltet **Cleaning** und **Feature Engineering**
- **Feature Engineering** nutzt Domänenwissen um Charakteristika, Eigenschaften und Attribute aus den Rohdaten zu extrahieren
- **Kategorische Daten** müssen so **umgewandelt** werden, dass sie **mathematisch transformierbar** sind

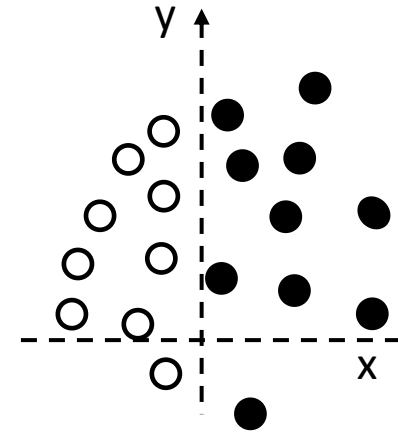
### Rohdaten



### Transformation



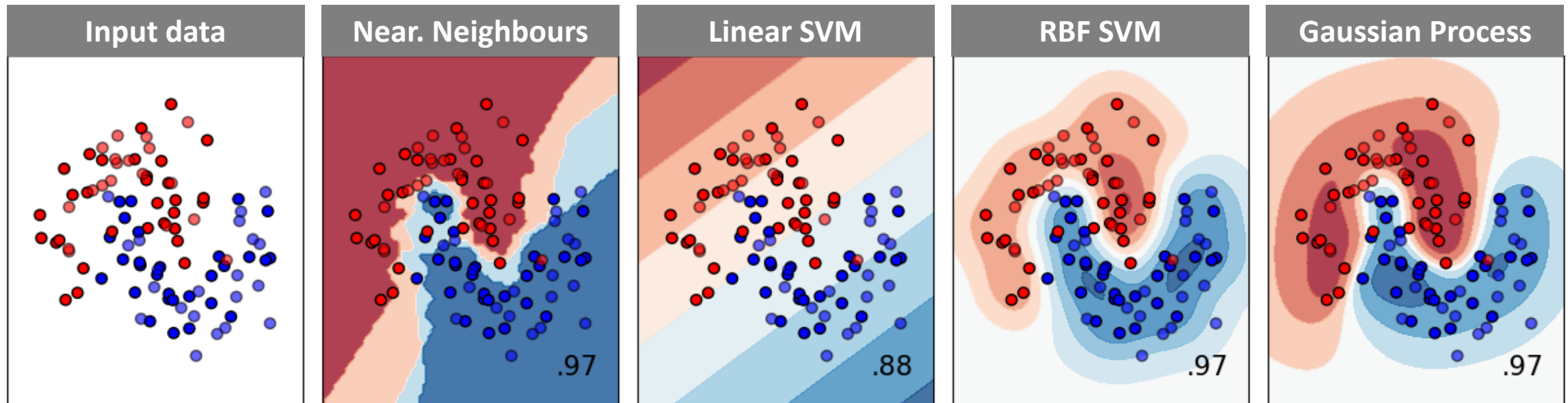
### Bessere Darstellung



# Es gibt viele unterschiedliche Modelle, die jeweils ihre Vor- und Nachteile haben

## Fun Facts

- Ein Modell ist eine **parametrische Repräsentation** einer mathematischen Funktion
- Die Wahl eines Modells führt **immer implizit Annahmen** mit ein
- Beispiel: Lineare Regression  $f(\mathbf{x}, \mathbf{a}) = \sum_{i=1}^N a_i x_i$
- Logistische Regression, Entscheidungsbäume, Random Forest, Support Vector Machines, ...



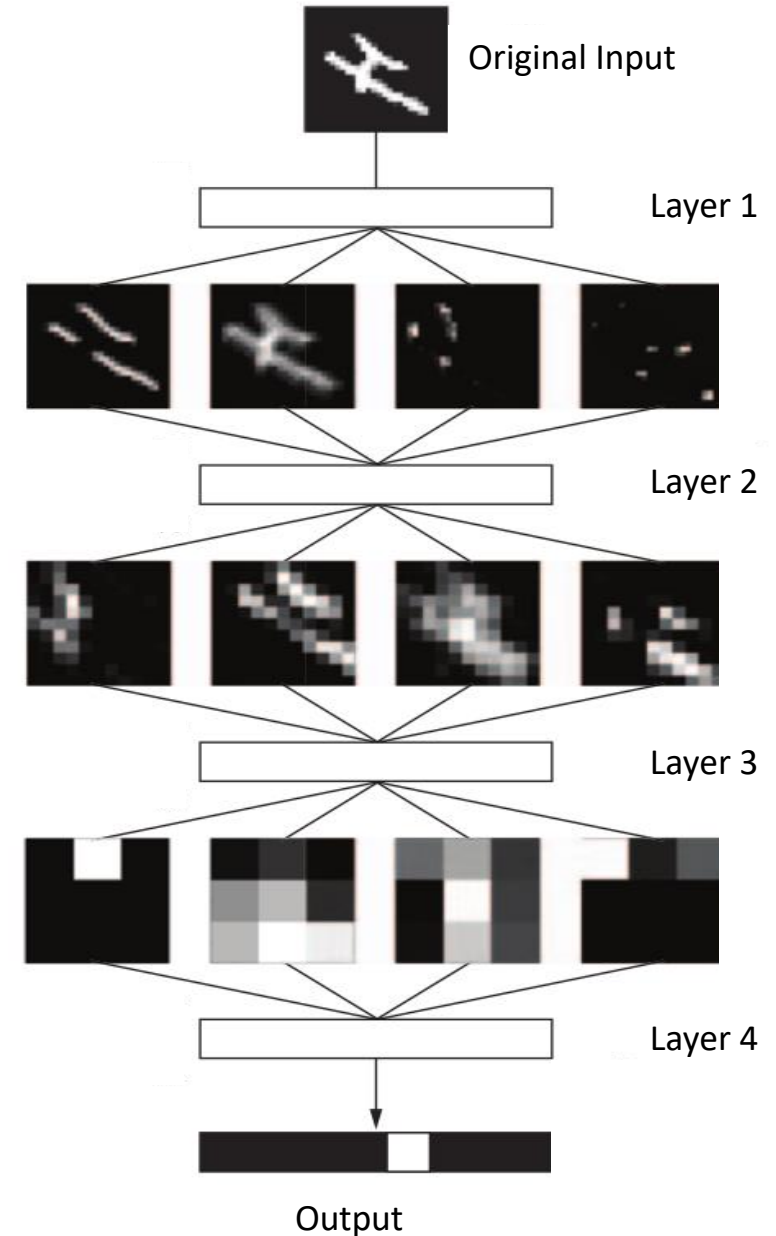
# Deep Learning ist das Lernen mit tiefen neuronalen Netzen

## Layer

- Deep Learning heißt so, weil **viele Layer hintereinander** gesetzt werden
- Der Aufbau und die Verknüpfung der Layer heißt **Architektur**
- Layer **transformieren** die eingehenden Daten

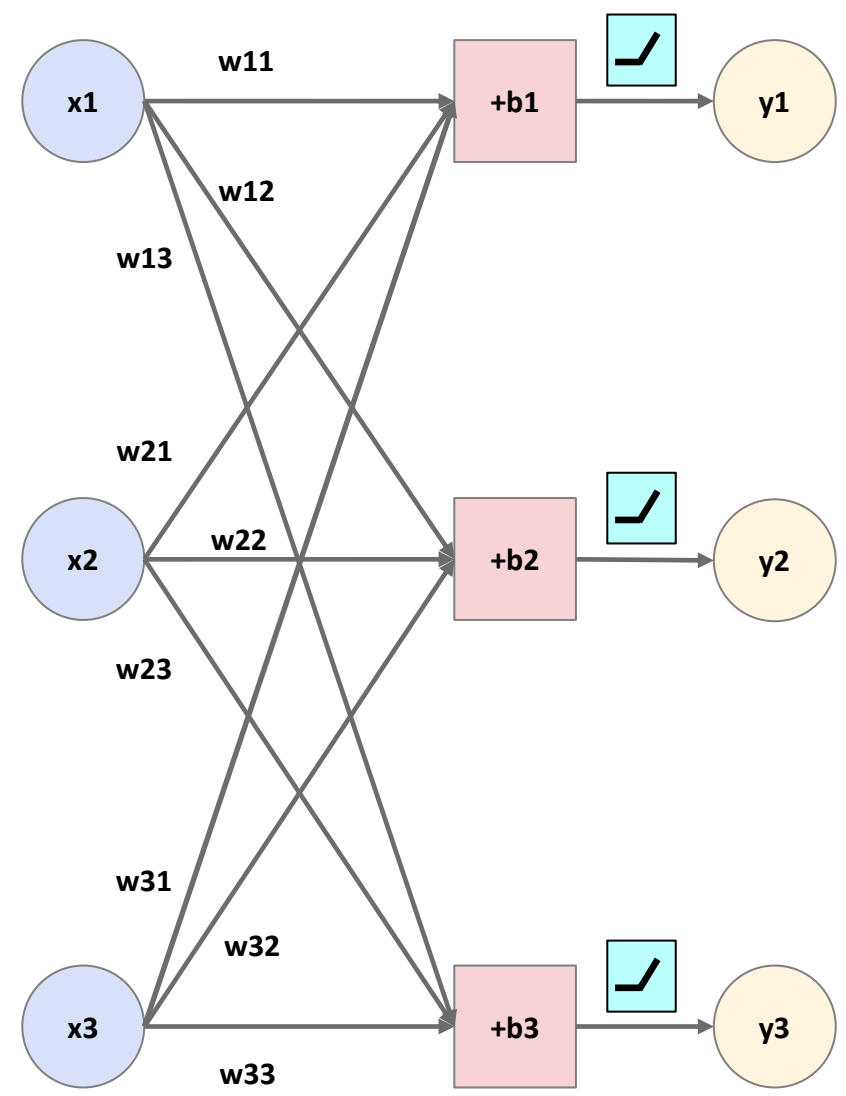
## Features

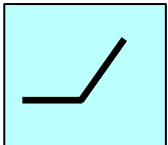
- **Feature Engineering** wird **nicht** mehr **benötigt**
- In jedem Layer iterative eine zunehmend **aussagekräftigen Darstellung** gelernt
- Es ist **nicht möglich** die **gelernten Features zu verstehen**

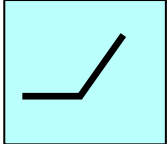


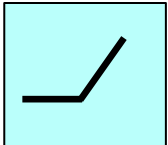


# Ein Neuronales Netz lässt sich auch als mathematische Formel darstellen



=   $w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + b_1$

=   $w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + b_2$

=   $w_{13} x_1 + w_{23} x_2 + w_{33} x_3 + b_3$

# Das Lernen ist eine stochastische Optimierung

1

## Vorbereitung

- Festlegen von Modell und Architektur
- Vorinitialisieren der Gewichte

2

## Lossfunktion (Bsp: Empirisches Risiko)

- Auswahl ist abhängig vom Problem

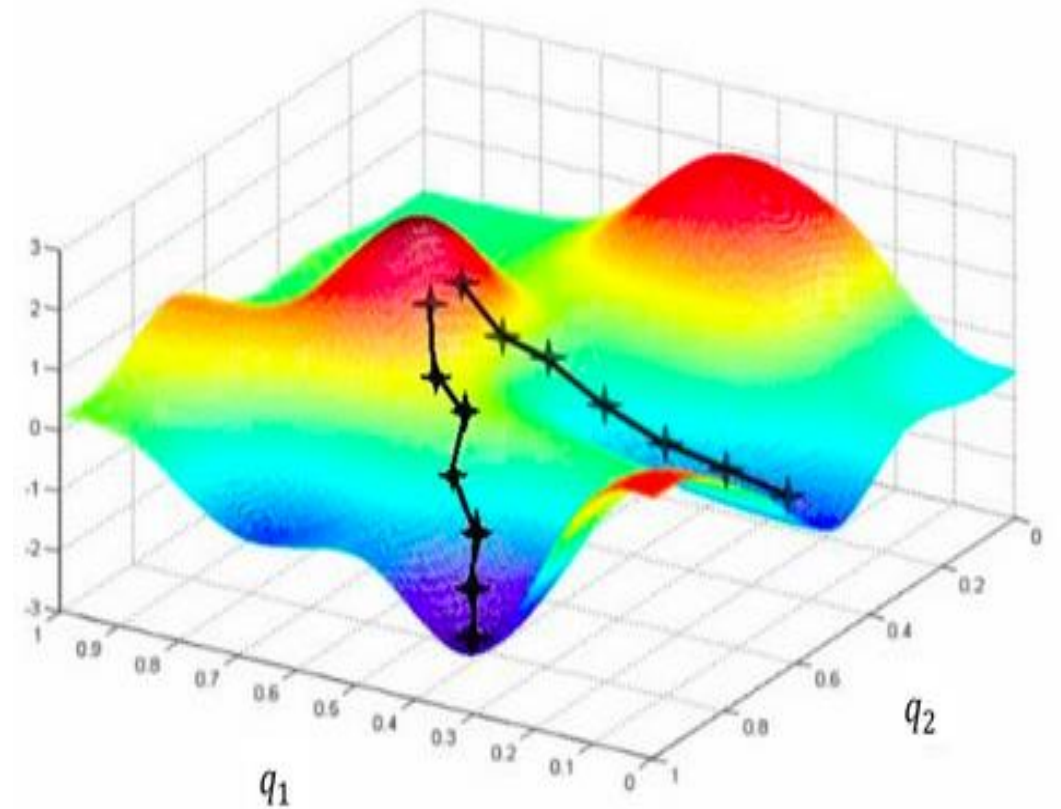
$$J(x, y, \alpha) = \frac{1}{2} \sum (y_i - \varphi(x_i, \alpha))^2$$

3

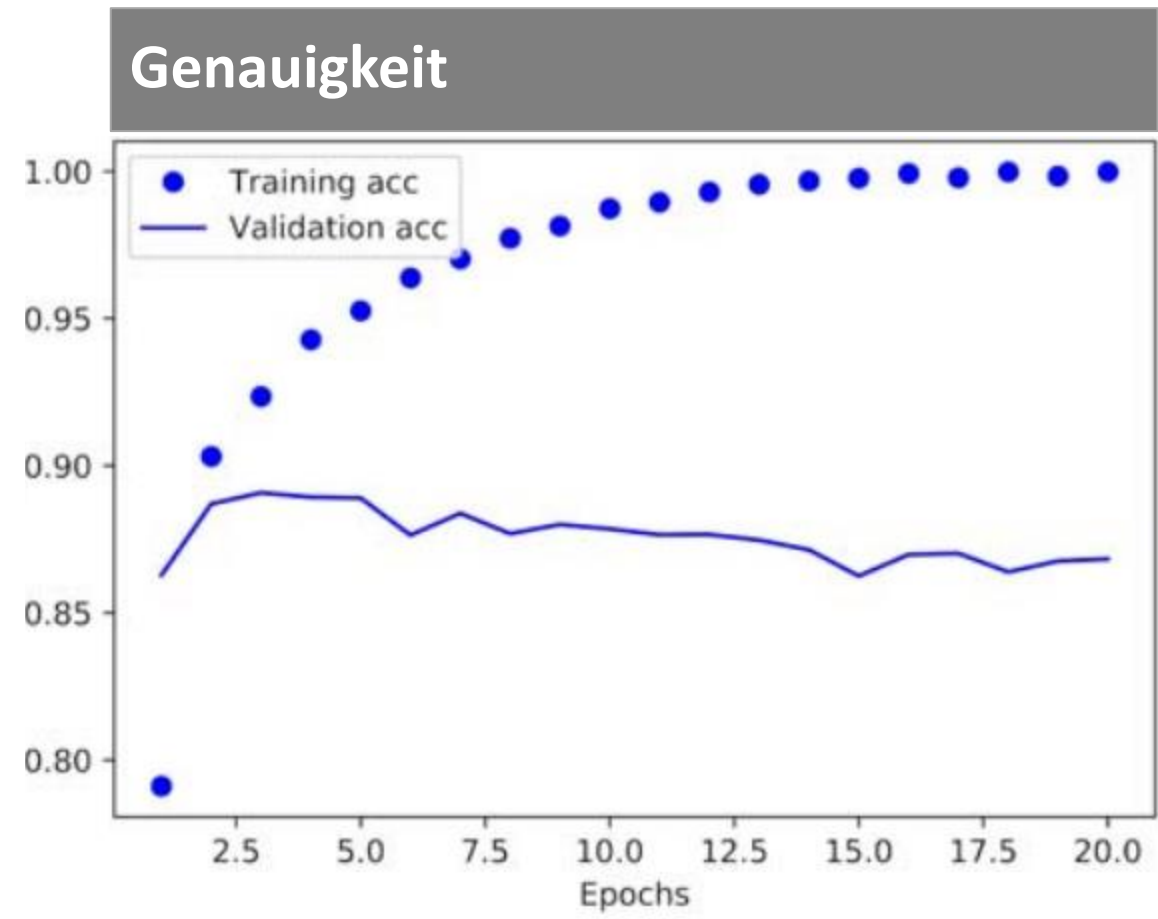
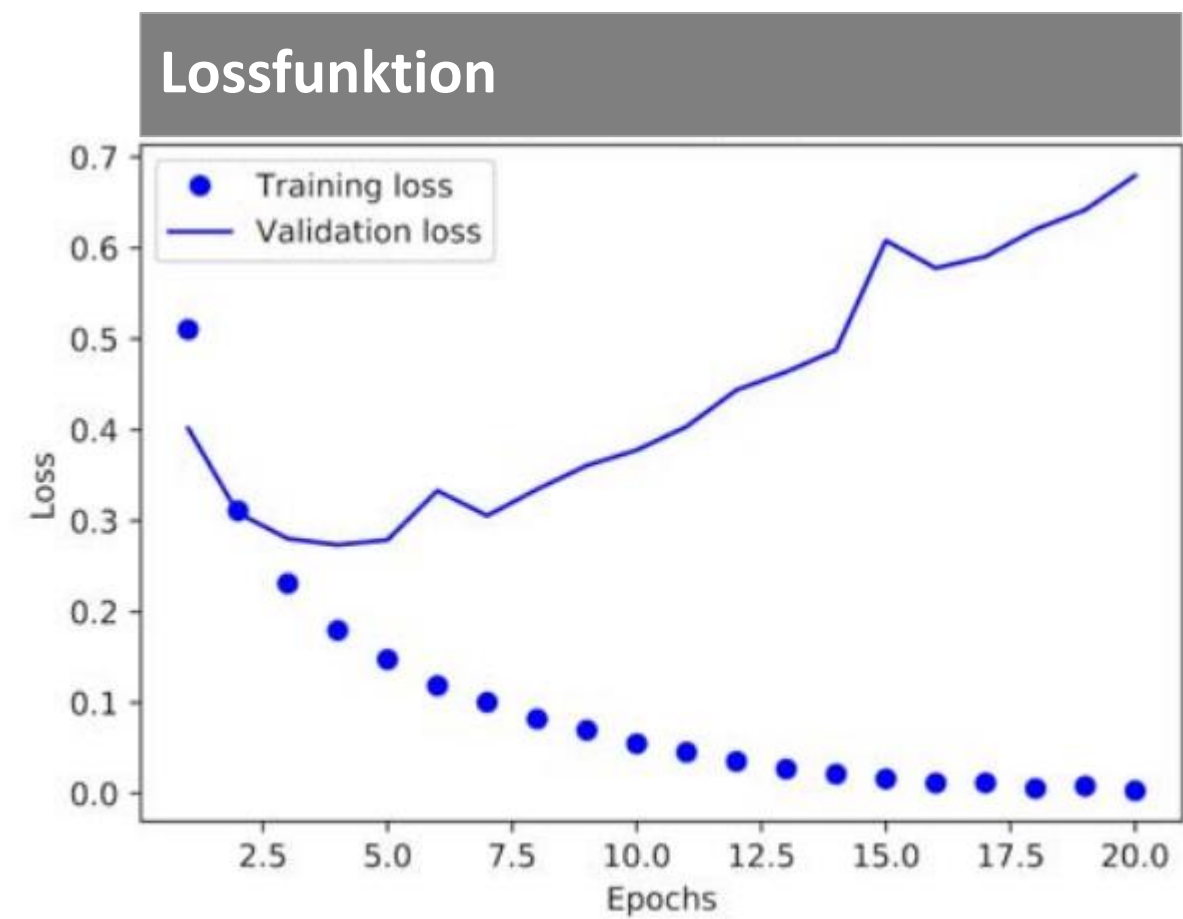
## Gradientenabstieg

- Gradient der Lossfunktion (Ableitung) ist einfach zu berechnen
- Gradient zeigt in Richtung der größten Veränderung

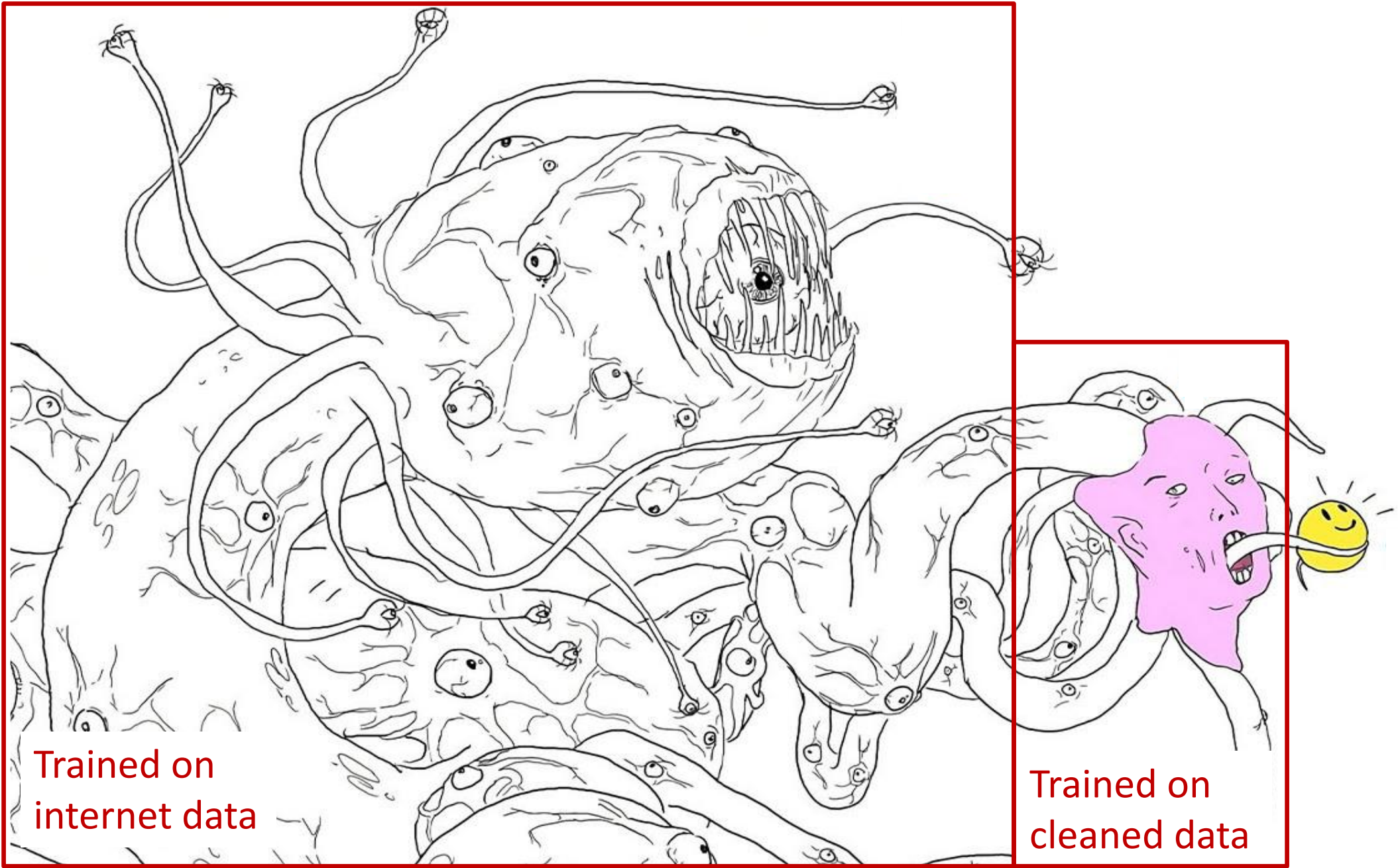
$$\alpha_{new} = \alpha_{old} - \varepsilon \nabla J(x, y, \alpha)$$



# Das Lernen minimiert die Lossfunktion und maximiert die Genauigkeit



Unterschiedliche Trainingsmethoden können kombiniert werden



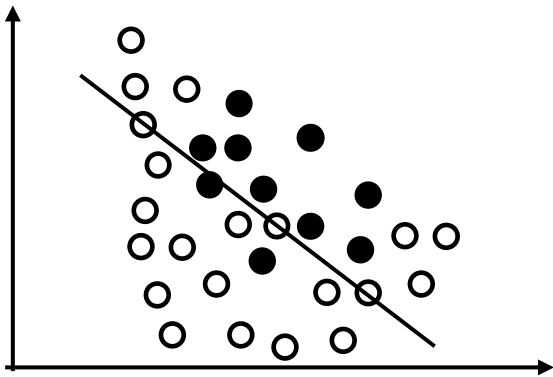
ChatGPT



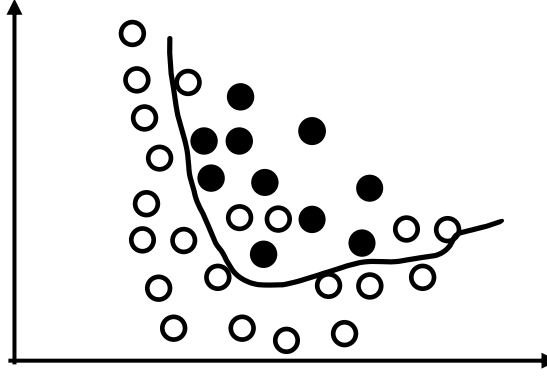
## Güte eines Modells

- Das Training ist immer ein **Trade-off** zwischen **Overfitting** und **Underfitting**
- Ein **Modell** wird meistens **nicht 100% korrekt** sein
- Die **Anwendung** bestimmt darüber **welcher Fehler akzeptierbar ist** und welcher nicht
- Welches **Modell das Beste** ist, ist mathematisch **nicht zu beweisen**

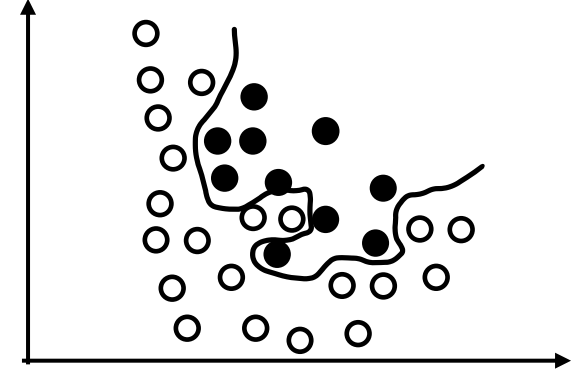
### Underfitting



### Bestes Model?!?



### Overfitting



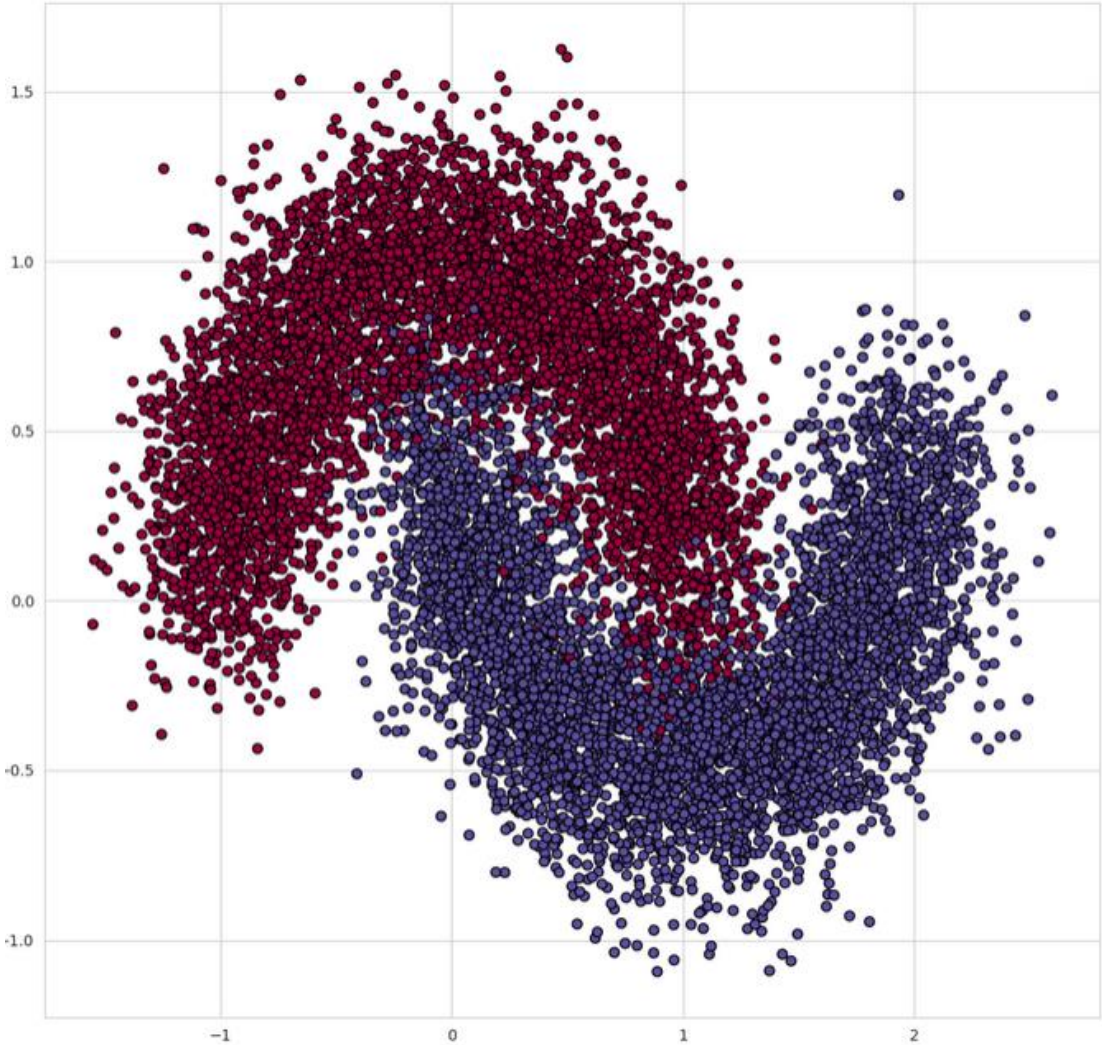
# Lasst uns mal ein Beispiel anschauen

## Experiment

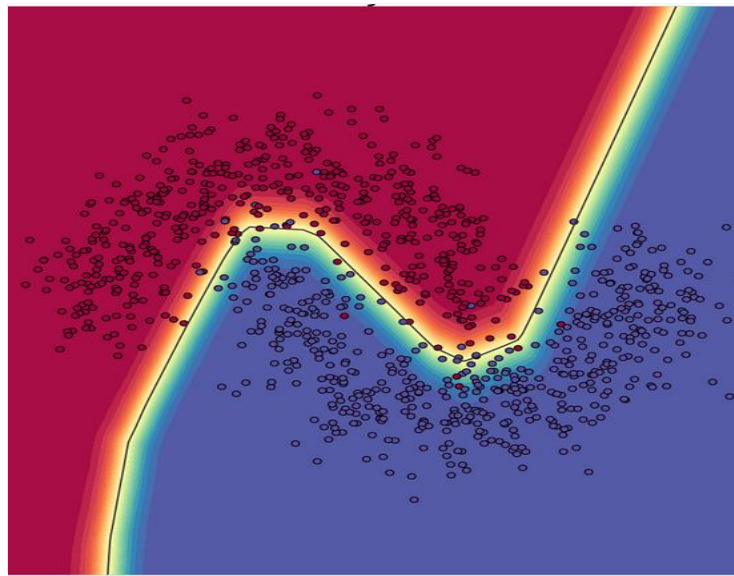
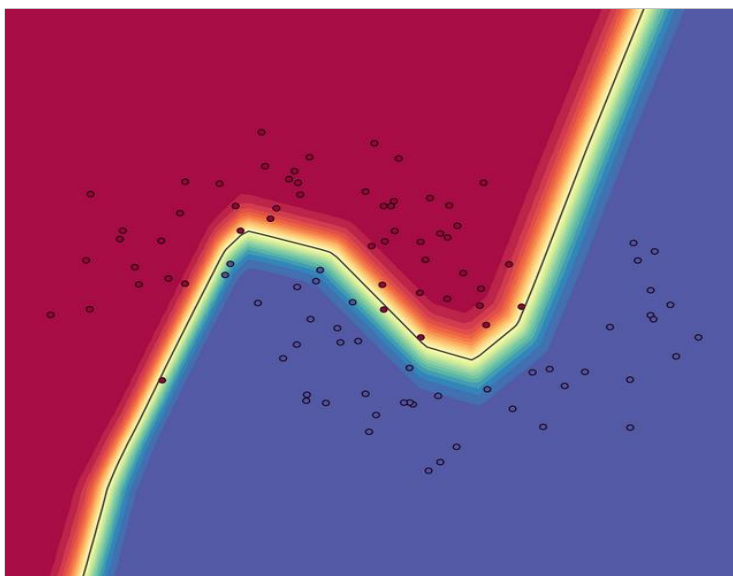
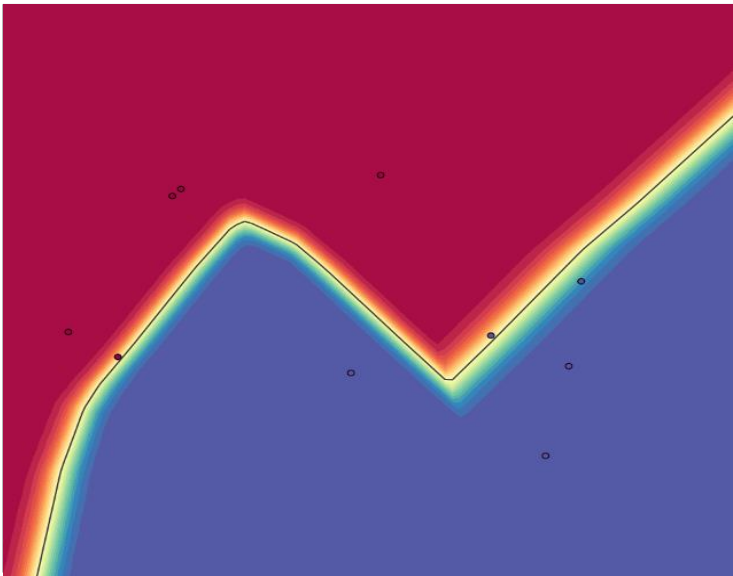
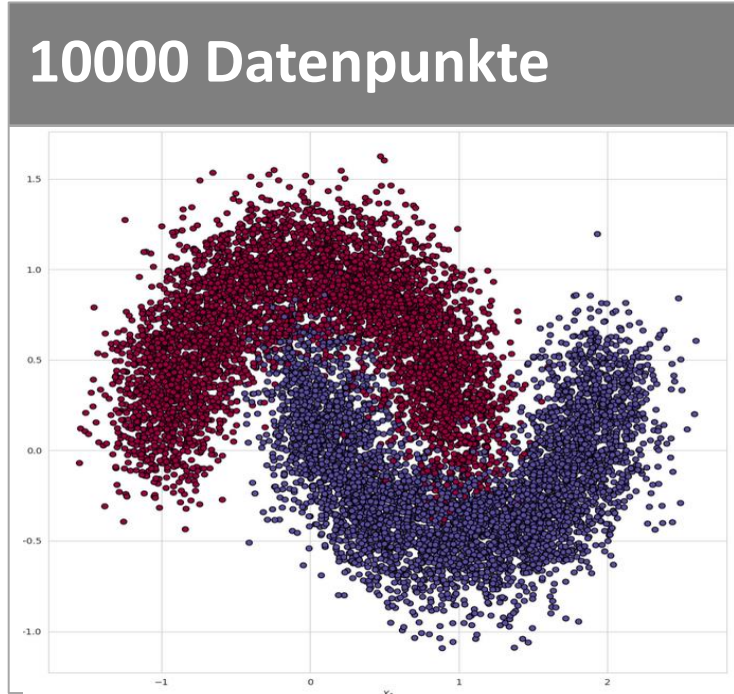
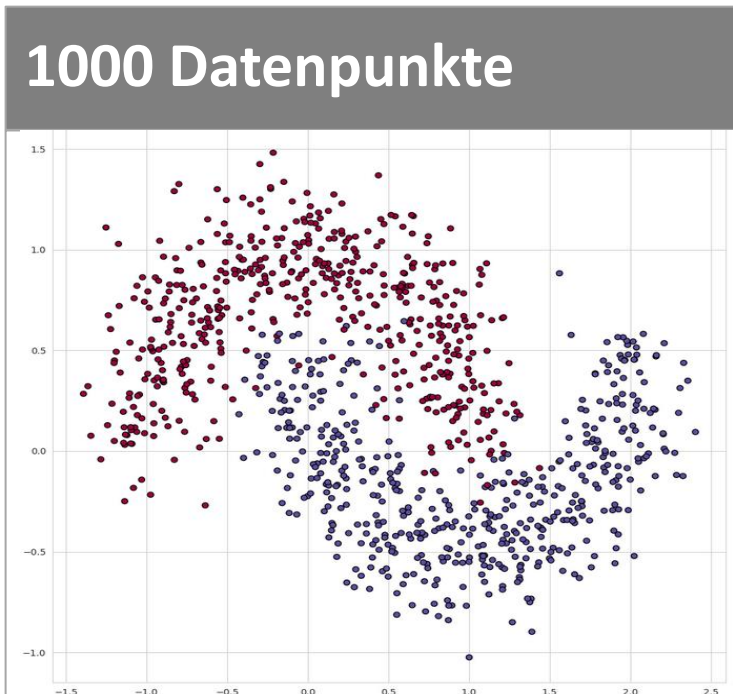
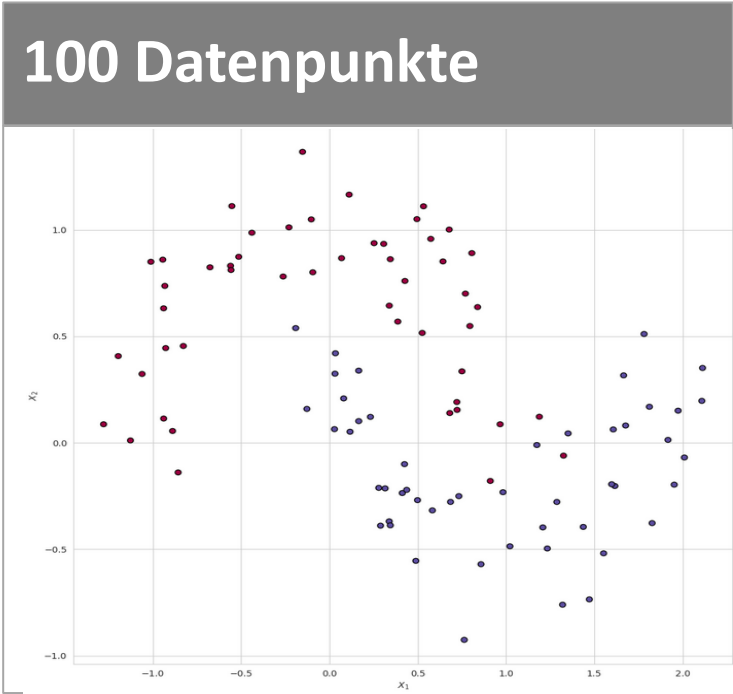
- **Klassifikation** von zwei Gruppen
- Daten haben **2 Feature** [x1,x2,Label]
- N Datenpunkte
  - 90% Trainingdata
  - 10% Testdata
- 10000 Gradientenschritte

## Ergebnisse

Trainingsdata	Accuracy
100	0.90
1000	0.98
10000	0.98



# Das Beispiel verdeutlicht die Datenabhängigkeit



# ML ist nur bei bestimmten Problemen überlegen

---

Eindeutige Daten sind in großer Menge verfügbar

---

**Daten**

**Komplexität**

---

Zusammenhänge sind **nicht offensichtlich** und lassen sich **nicht einfach abbilden**

---

**Evaluierbarkeit**

---

Es ist **leicht zu messen**, welches Modell oder welcher Lösungsweg eine **bessere Lösung** liefert

---



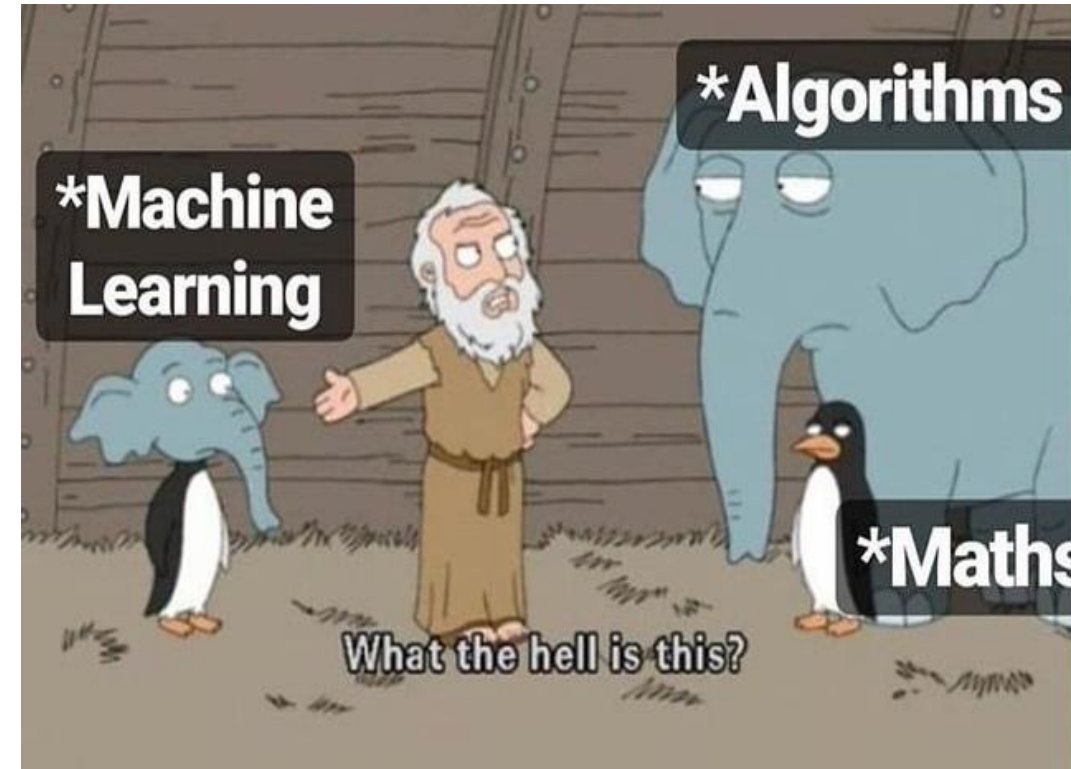
# Machine Learning ist ein wilder Mix aus Mathe und Programmierung

## Daten und Training

- Es gibt sehr **viele Parameter**, die ausgewählt werden müssen, **ohne dass man genau weiß was sie tun**
- Die **Regeln** sind in den **Daten versteckt** und es ist **schwer zu verstehen**, ob das „Richtige“ **gelernt worden ist**

## Anwendung

- Es braucht viel **Zeit und Erfahrung** eine Anwendung zu testen und **Edge Cases zu erkennen**
- Es braucht eine **konstante Überwachung** des Algorithmus
- Es kann ziemlich **schwierig** sein einen **Edge Case zu beheben**



Machine Learning wird nicht mehr weg gehen und deshalb musst du dich damit auseinandersetzen.

## Where to start

### Workshop @ Camp

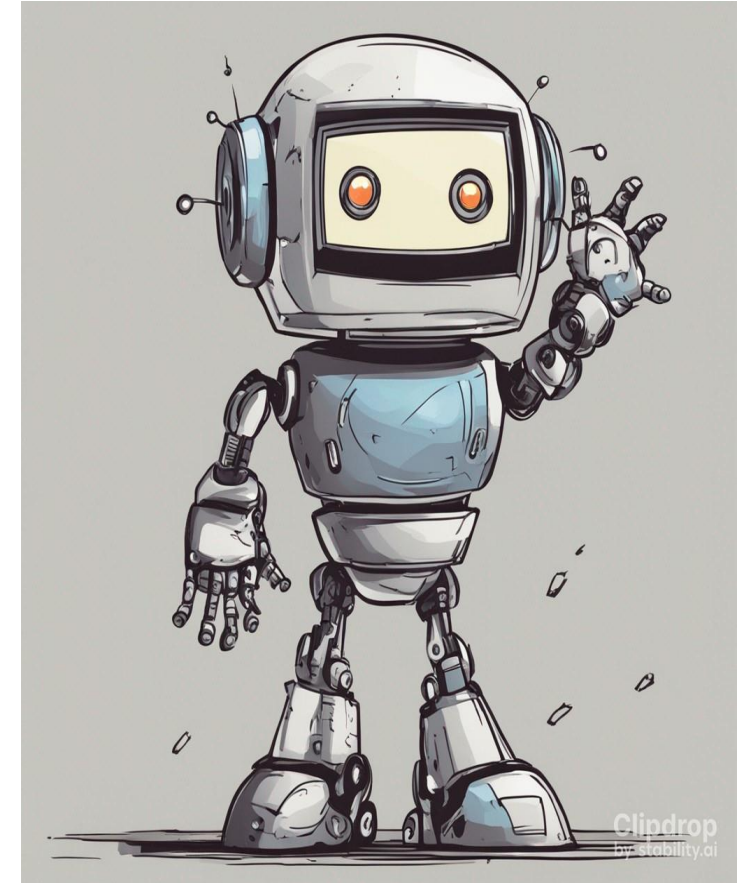
- Workshop @jugend hackt Machine Learning 101 2023-08-19 14h
- Email: jannes@srlabs.de

### Kaggle

<https://www.kaggle.com/>

### Bücher

- Deep Learning with Python, F. Challet
- M. P. Deisenroth et al  
<https://mml-book.github.io/book/mml-book.pdf>
- I. Goodfellow et al.  
<https://www.deeplearningbook.org/>



**Security  
Research  
Labs**

- [1,7,22] Stable Diffusion
- [2] <https://www.pinterest.com/pin/237916792788104970/>
- [3] [https://www.reddit.com/r/ich\\_iel/comments/xi8fs8/ichiel/](https://www.reddit.com/r/ich_iel/comments/xi8fs8/ichiel/)
- [5] <https://www.freecodecamp.org/news/chihuahua-or-muffin-my-search-for-the-best-computer-vision-api-cbda4d6b425d/>
- [6] <https://noeliagorod.com/2019/05/21/machine-learning-for-everyone-in-simple-words-with-real-world-examples-yes-again/>
- [3,8,10,12,15,17] Deep Learning with Python, Francois Challet
- [5,9] xkcd.com [2048, 1838]
- [11] <https://scikit-learn.org/>
- [13] <https://www.heise.de/select/ix/2017/9/1504455013673842>
- [14] [shashank-ojha.github.io/ParallelGradientDescent](https://github.com/shashank-ojha/ParallelGradientDescent)
- [16] <https://knowyourmeme.com/memes/shoggoth-with-smiley-face-artificial-intelligence>
- [21] <https://www.reddit.com/r/ProgrammerHumor>