

**30.05.24**

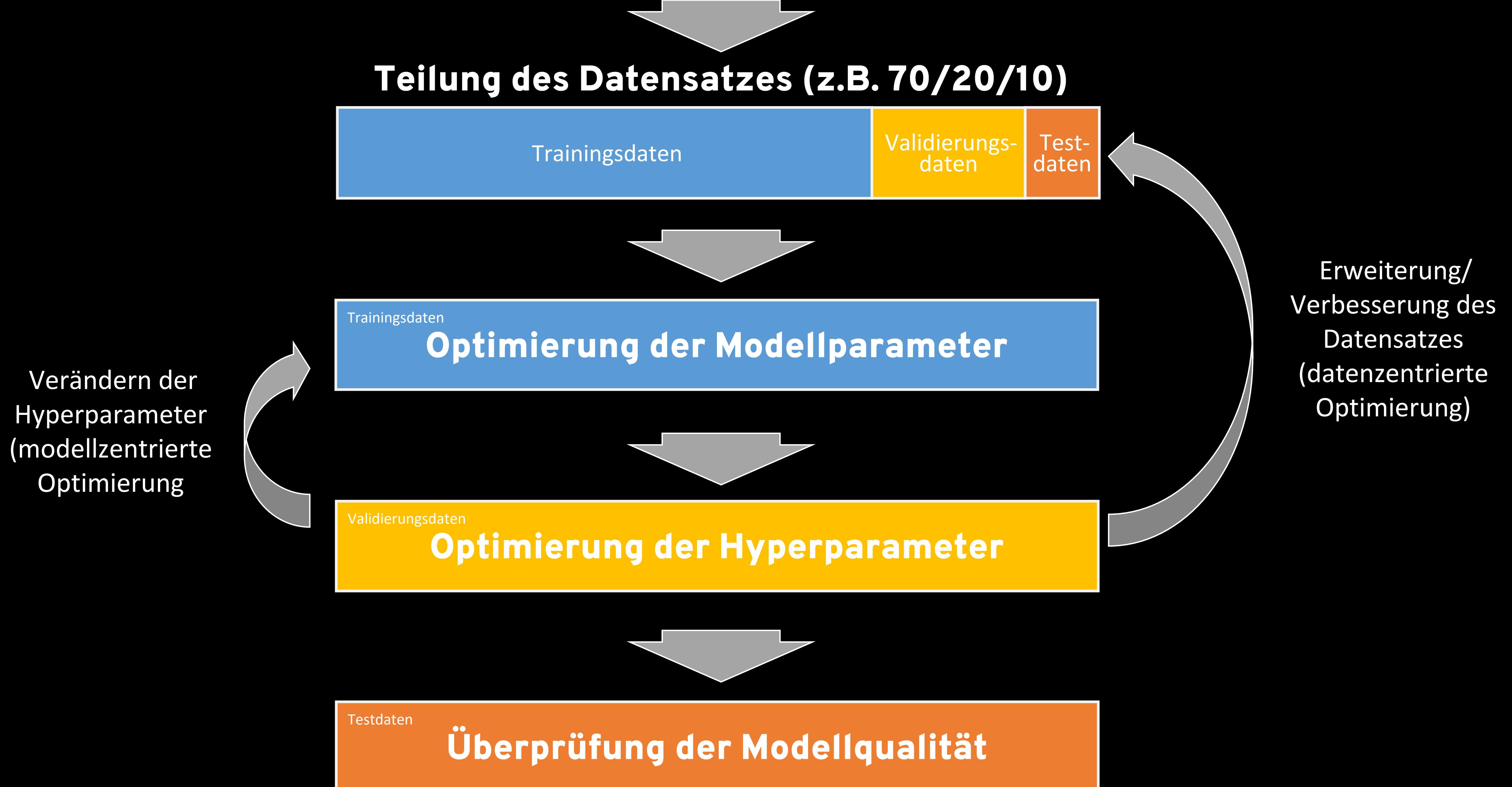
# Einführung in Data Science und maschinelles Lernen

## OVERFITTING UND MODELL-EVALUATION

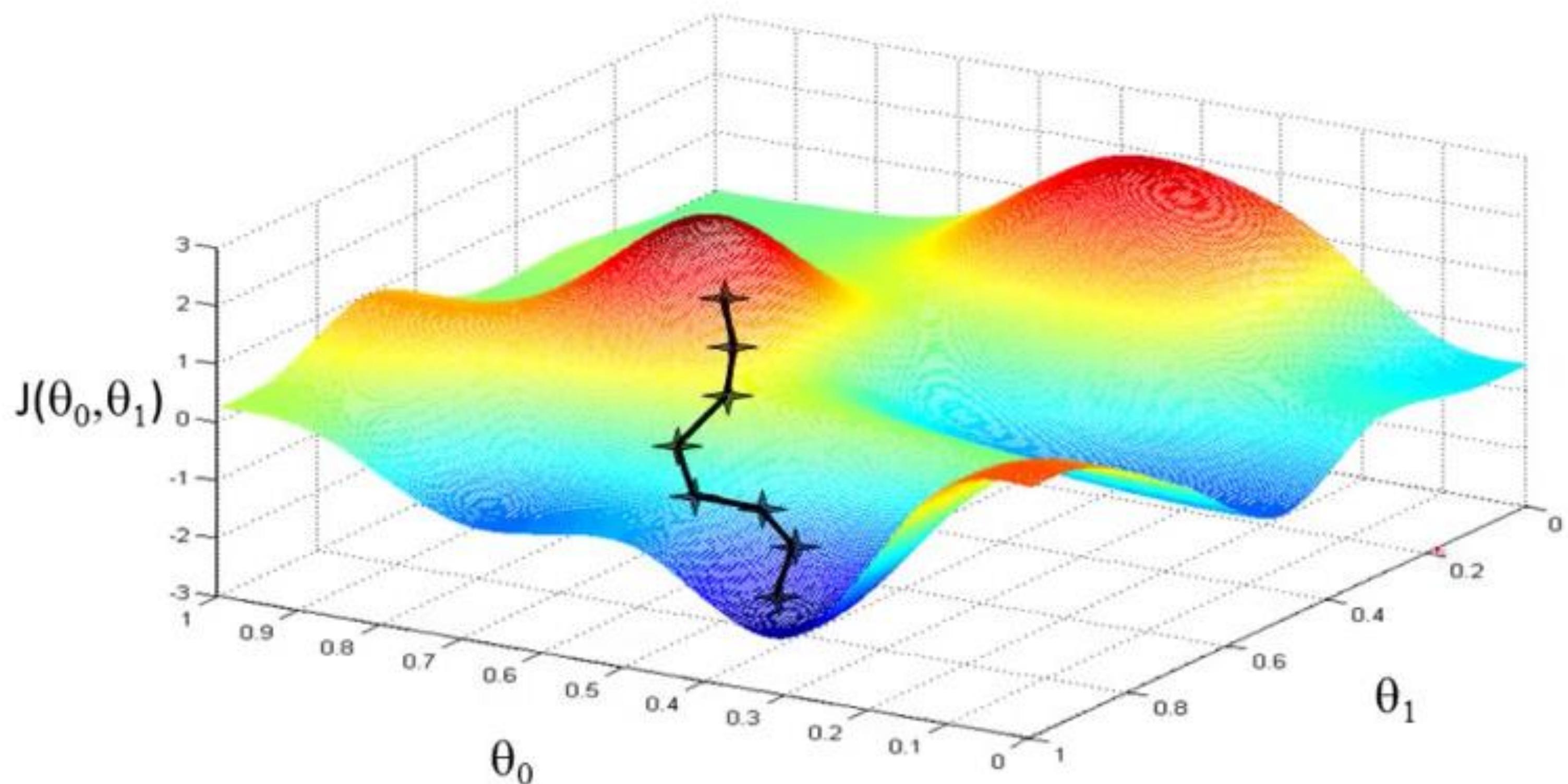
- **Wiederholung wichtiger Begriffe des ML**
- **Interaktionseffekte**
- **Overfitting und Regularisierung**
- **Modellgütekriterien**
- **Einführung in neuronale Netze**

# **TEAMZUSAMMENSETZUNGEN**

# Wahl eines Prognosemodells



# OPTIMIERUNGSFUNKTION (OPTIMIZER)



- Iteratives Verfahren (Gradient Descent), um das Minimum der Kostenfunktion zu finden.
- Die Lernrate („Learning Parameter“) beschreibt die Schrittgröße zur Annäherung an das Minimum.

Quelle: <https://www.coursera.org/learn/machine-learning>

# OPTIMIERUNG DER MODELLPARAMETER

## *Forward Propagation*

- Berechnung des vorhergesagten Wertes auf Basis der aktuellen Modellgleichung
- Berechnung der Kosten bzw. des Loss

## *Backward Propagation*

- Berechnung des Gradienten (d.h. aller partiellen Ableitungen), um die Richtung des Minimums zu bestimmen.
- Anpassung der Modellparameter im Ausmaß der definierten Lernrate:

$$\text{Neuer Wert} = \text{Alter Wert} - \text{Lernrate} \times \text{Partieller Gradient}$$

# MODELLPARAMETER VS. HYPERPARAMETER

## *Modellparameter*

- Parameter, die während des Trainings optimiert werden (insbesondere die Gewichte).

## *Hyperparameter*

- Parameter, die vor dem Training gesetzt werden (etwa die Lernrate oder die Anzahl der Schichten in einem neuronalen Netz).

# FEATURES, LABELS, PARAMETER

**Machine Learning**

**Feature, Input Variable**

**Label, Target Variable,  
Output**

**Gewichte (Weights),  
(Modell-)Parameter**

**Statistik**

**Unabhängige Variable**

**Abhängige Variable**

**Koeffizienten**

**Beschreibung**

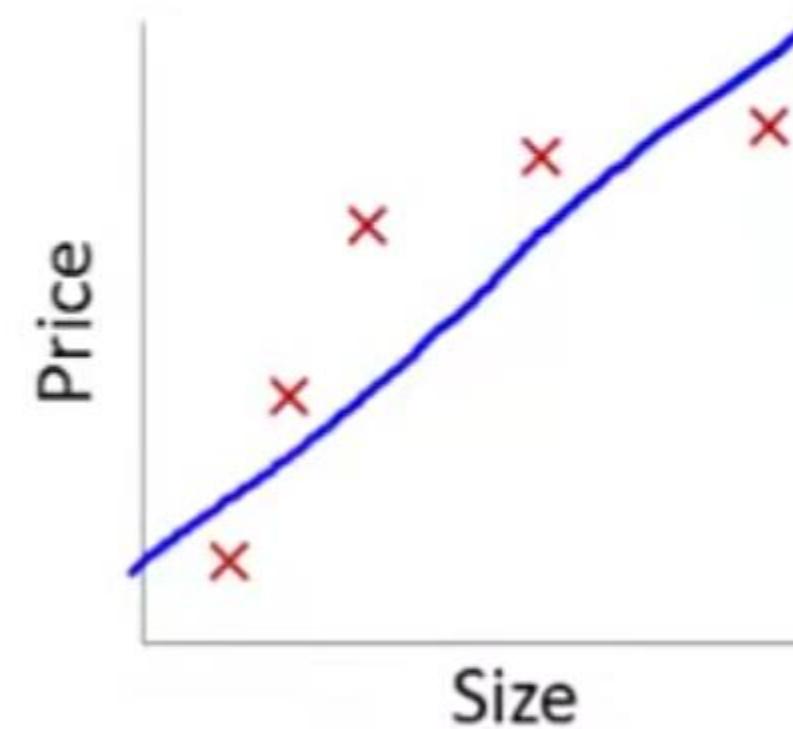
**Eingangsdaten, die für die  
Vorhersage genutzt werden**

**Beobachtete Ergebnisse, anhand  
derer das Modell trainiert wird**

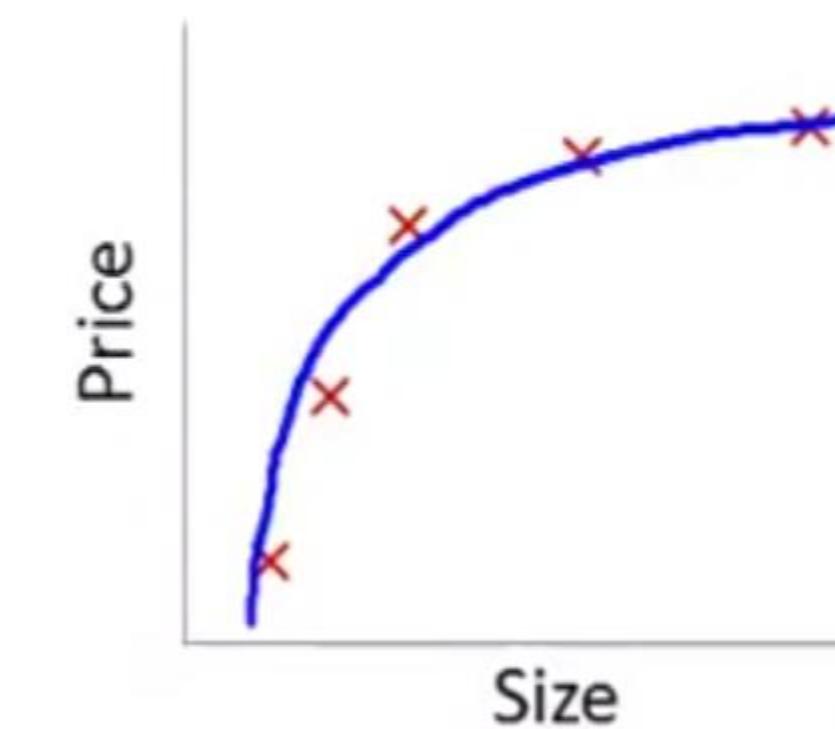
**Werden im Rahmen des Trainings  
optimiert („gelernt“)**

# OVERFITTING

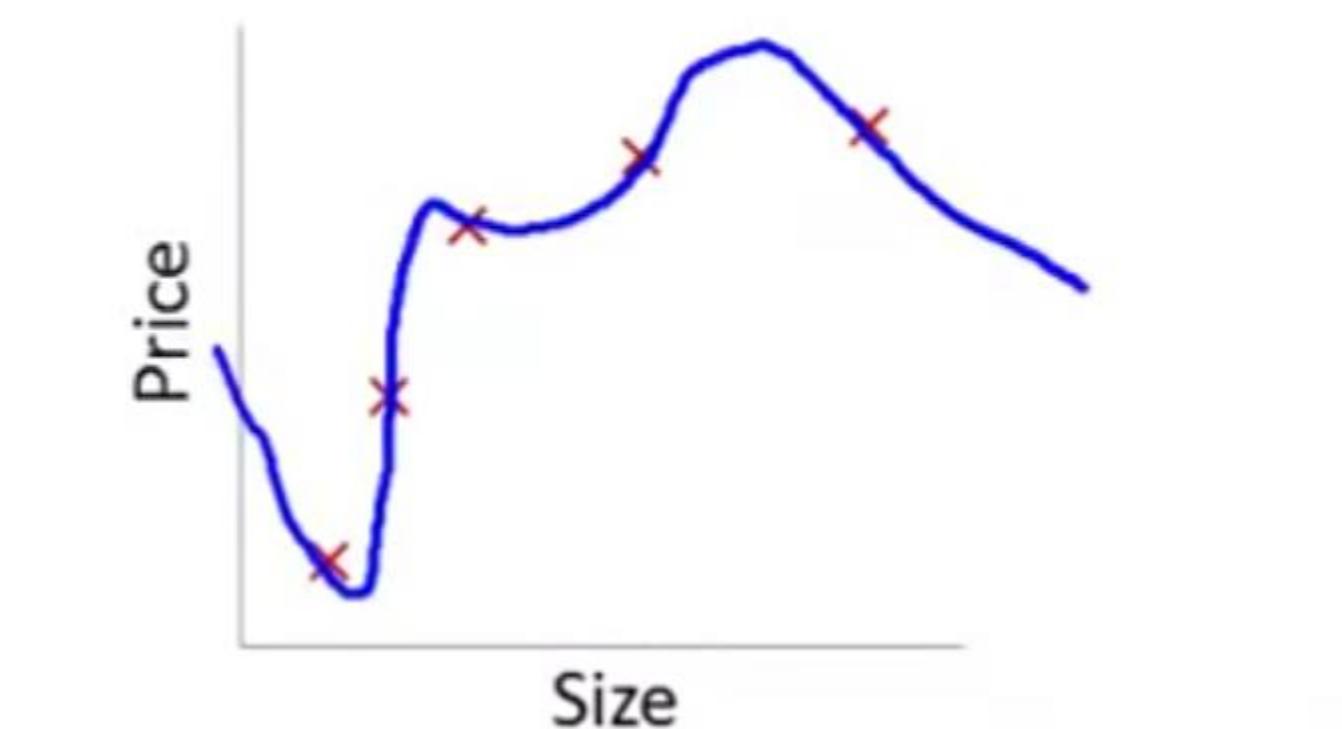
Example: Linear regression (housing prices)



$\rightarrow \theta_0 + \theta_1 x$   
"Underfit" "High bias"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$   
"Just right"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$   
"Overfit" "High variance"

**Overfitting:** If we have too many features, the learned hypothesis may fit the training set very well ( $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$ ), but fail to generalize to new examples (predict prices on new examples).

# STRATEGIEN ZUR VERMEIDUNG VON OVERFITTING

Options:

1. Reduce number of features.
  - — Manually select which features to keep.
  - — Model selection algorithm

---

2. Regularization.
  - — Keep all the features, but reduce magnitude/values of parameters  $\theta_j$ .
  - Works well when we have a lot of features, each of which contributes a bit to predicting  $y$ .

# REGULARISIERUNG

„Bestrafen“ der Verwendung von Variableninformation im Rahmen der Kostenfunktion

Lineares Modell mit mehreren Variablen  $x_1, x_2$  und vielen möglichen weiteren:

$$h_x(\theta_0, \theta_1, \theta_2, \dots) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots$$

(mit  $\theta_0, \theta_1, \theta_2, \dots$  als den zu schätzenden Modellparametern)

Kostenfunktion mit Regularisierung:

$$J_x(\theta_0, \theta_1, \theta_2, \dots) = \frac{1}{m} \left[ \sum_m (h_x(\theta_0, \theta_1, \theta_2, \dots) - y)^2 + \lambda(|\theta_0| + |\theta_1| + |\theta_2| + \dots) \right]$$

(mit  $\lambda$  als Regularisierungsparameter)

# BEISPIEL-NOTEBOOK

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** repo-template-intro-to-data-science-and-ml [Codespaces: fictional waddle]
- File List:** overfitting.py U, overfitting.ipynb U (selected), interaction effects.py U, interaction effect.ipynb U
- Toolbar:** Python 3.10.13, Code, Markdown, Run All, Restart, Clear All Outputs, Variables, Outline, ...
- Left Sidebar:** Search, Help, File, Cell, Kernel, Help, Python icon.
- Section Header:** Demonstrating Overfitting and the Importance of Feature Selection
- Text Content:** This notebook illustrates the phenomenon of overfitting in statistical models, particularly in the context of regression analysis. Overfitting occurs when a model learns not only the underlying pattern but also the noise in the training data, leading to poor performance on new, unseen data. This is often a result of using excessively complex models or including irrelevant features in the model.  
We will explore how different regression models respond to a mix of relevant and irrelevant features and demonstrate the utility of techniques like Ridge Regression for mitigating overfitting.
- Section Header:** Import Libraries
- Code Block:** # Importing necessary libraries for data handling, mathematical operations, and plotting  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression, Ridge  
from sklearn.metrics import mean\_squared\_error
- Cell Number:** [1]
- Language:** Python
- Bottom Navigation:** Codespaces: fictional waddle, main\*, 53, 57, 8, Cell 1 of 14, Layout: German

# INTERAKTIONSEFFEKTE

# BEISPIEL-NOTEBOOK

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** repo-template-intro-to-data-science-and-ml [Codespaces: fictional waddle]
- File List:** overfitting.ipynb, interaction effects.py, interaction effect.ipynb (selected)
- Toolbar:** + Code, + Markdown, Run All, Restart, Clear All Outputs, Variables, Outline, Python 3.10.13, Edit, Delete
- Section Header:** Analysis of Sales Influenced by Season and Temperature
- Description:** This notebook aims to explore how different seasons and temperature levels affect sales through a synthetic dataset. We will generate data representing sales across different seasons and temperatures, hypothesizing that interactions between these factors can significantly impact sales outcomes.
- Text:** We will visually and statistically analyze the main effects of each factor and their interaction to understand their impacts better. The analysis will conclude with fitting linear models to quantify these effects.
- Code Cell:** Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

- Output:** [1] 6.9s Python
- Section Header:** Generate Synthetic Dataset
- Bottom Status Bar:** Codespaces: fictional waddle, main\*, 0 0 8, Ln 17, Col 1, Spaces: 4, LF, Cell 1 of 13, Layout: German

# BATCHES, STEPS UND EPOCHE

## **Batch**

- **Die Gesamtmenge an Trainingsdaten wird in separate Teilgruppen mit gleicher Größe eingeteilt.**
- **Standardgröße eines Batches ist 32.**

## **Step**

- **Die Backward-Propagation mit einem Batch (alle Gewichte werden einmal optimiert)**

## **Epoche**

- **Optimierung des Modells anhand des vollständigen Trainingsdatensatzes:  
Number of Steps x Batch Size = Training Sample Size**
- **Je nach Modell genügen sehr wenige Epochen oder sind mehrere hundert oder tausend Epochen notwendig zur Optimierung.**

# MODELLGÜTEKRITERIEN FÜR REGRESSIONSAUFGABEN

errors: **forecast - actual** (auch: residuals)

mae: **mean(abs(errors))**

mape: **mean(abs(errors/actual))**

mse: **mean(errors^2)**

rmse: **sqrt(mean(errors^2))**

rse: **sum(errors^2) / sum( (actual-mean(actual))^2 )**

**$r^2 = 1 - rse$**

Video (3 Minuten) mit Erklärung und Darstellung der Kriterien:

<https://www.coursera.org/lecture/machine-learning-with-python/evaluation-metrics-in-regression-models-5SxtZ>

# MODELLGÜTEKRITERIEN FÜR KLASSEIFIKATIONSAUFGABEN

$$\text{Accuracy} = \frac{\text{Anzahl richtiger Vorhersagen}}{\text{Gesamtzahl der Vorhersagen}}$$

$$\text{Precision} = \frac{\text{Anzahl richtiger positiver Vorhersagen}}{\text{Gesamtzahl der Vorhersagen}}$$

$$\text{Recall} = \frac{\text{Anzahl richtiger positiver Vorhersagen}}{\text{Anzahl tatsächlich positiver Fälle}}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

		Tatsächliche Beobachtungen	
		<i>positiv</i>	<i>negativ</i>
Vorhersagen	<i>positiv</i>	true	false
	<i>negativ</i>	false	true

Blog mit genauerer Erklärung der Metriken:

<https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>

# Level up with the largest AI & ML community

Join over 15M+ machine learners to share, stress test, and stay up-to-date on all the latest ML techniques and technologies. Discover a huge repository of community-published models, data & code for your next project.

 [Register with Google](#)[Register with Email](#)

## Who's on Kaggle?

### Learners

Dive into Kaggle courses, competitions & forums.



### Developers

Leverage Kaggle's models, notebooks & datasets.



### Researchers

Advance ML with our pre-trained model hub & competitions.



≡ kaggle

+ Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

# Competitions

Grow your data science skills by competing in our exciting competitions. Find help in the [documentation](#) or learn about [Community Competitions](#).

Host a Competition

Search competitions

Filters

All Competitions

Everything, past & present

Featured

Premier challenges with prizes

Getting Started

Approachable ML fundamentals

Research

Scientific and scholarly challenges

Community

Created by fellow Kagglers

Playgr

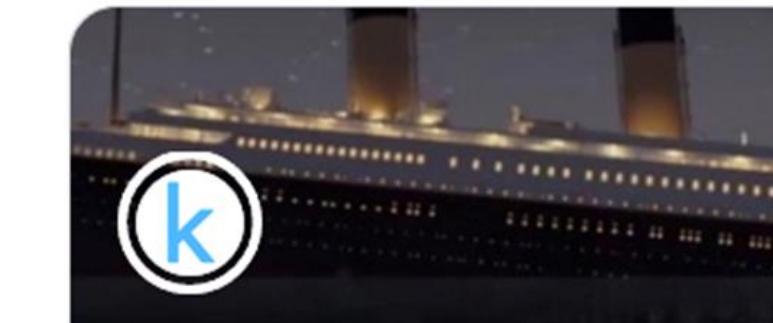
Fun practice problems

Get Started

See all

## New to Kaggle?

These competitions are perfect for newcomers.



Titanic - Machine Learning from Disaster

Start here! Predict survival on the Ti...  
Getting Started  
15573 Teams

Knowledge

Ongoing



House Prices - Advanced Regression...

Predict sales prices and practice fea...  
Getting Started  
4911 Teams

Knowledge

Ongoing



Spaceship Titanic

Predict which passengers are transp...  
Getting Started  
2555 Teams

Knowledge

Ongoing

View Active Events





Search competitions

Filters

+ Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More



### LLM - Detect AI Generated Text

⋮

Identify which essay was written by ...

Featured · Code Competition

1740 Teams

\$110,000

2 months to go



### Open Problems - Single-Cell...

⋮

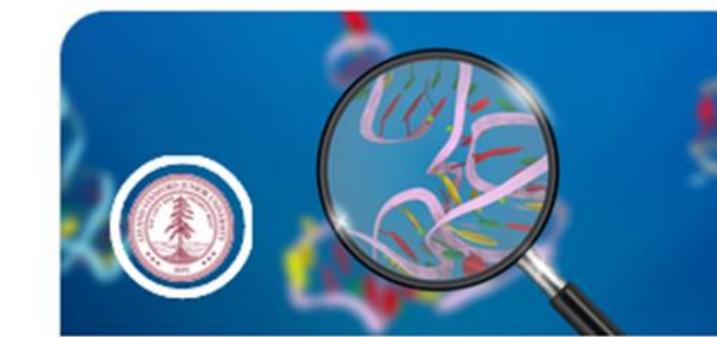
Predict how small molecules change...

Featured

1120 Teams

\$100,000

3 days to go



### Stanford Ribonanza RNA Folding

⋮

Create a model that predicts the str...

Research

676 Teams

\$100,000

10 days to go



### Optiver - Trading at the Close

⋮

Predict US stocks closing movements

Featured · Code Competition

3524 Teams

\$100,000

23 days to go



### NFL Big Data Bowl 2024

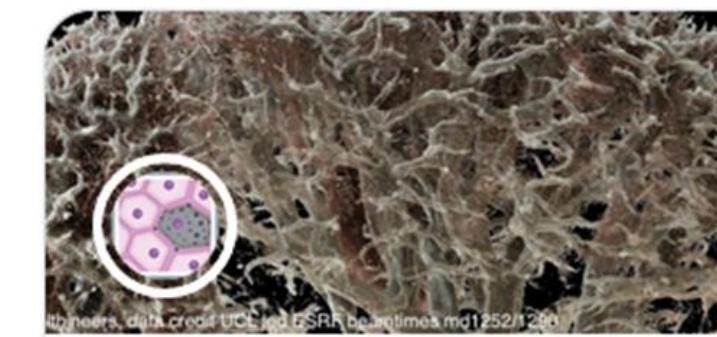
⋮

Help evaluate tackling tactics and st...

Analytics

\$100,000

a month to go



### SenNet + HOA - Hacking the Human...

⋮

Segment vasculature in 3D scans of ...

Research · Code Competition

149 Teams

\$80,000

2 months to go



### Linking Writing Processes to Writing...

⋮

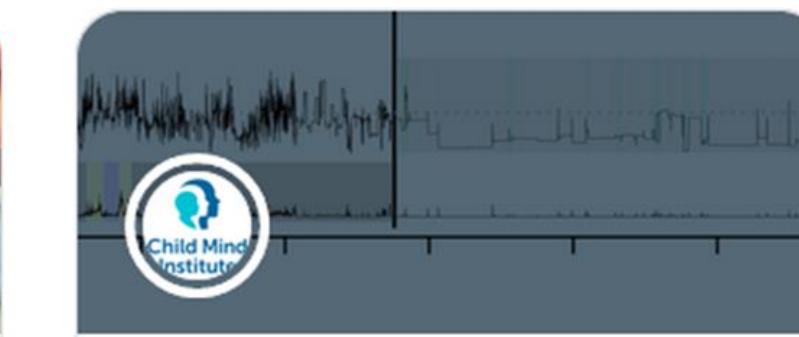
Use typing behavior to predict essa...

Featured · Code Competition

1139 Teams

\$55,000

a month to go



### Child Mind Institute - Detect Sleep States

⋮

Detect sleep onset and wake from w...

Featured · Code Competition

1762 Teams

\$50,000

8 days to go

View Active Events

[Create](#)[Home](#)[Competitions](#)[Datasets](#)[Models](#)[Code](#)[Discussions](#)[Learn](#)[More](#)

Research Prediction Competition

## Stanford Ribonanza RNA Folding

Create a model that predicts the structures of any RNA molecule



Stanford University · 676 teams · 10 days to go (3 days to go until merger deadline)

\$100,000  
Prize Money[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#)[Join Competition](#)

...

## Leaderboard

[Raw Data](#)[Refresh](#)

Search leaderboard

[Public](#)[Private](#)

This leaderboard is calculated with approximately 20% of the test data. The final results will be based on the other 80%, so the final standings may be different.

[Prize Contenders](#)

#	Team	Members	Score	Entries	Last	Join
1	HandFold		0.13733	85	40m	<a href="#">Join</a>
2	DI		0.13773	108	3h	<a href="#">Join</a>

[View Active Events](#)

# TESTEN DER VORHERSAGEN AUF KAGGLE

**kaggle**

Search

STEFFEN · COMMUNITY PREDICTION COMPETITION · A MONTH TO GO

Submit Prediction ...

## Bakery Sales Prediction Summer 2024

Based on observed bakery sales for 6 product categories over 3 years, participants will predict the sales for the year following.

Overview Data Code Models Discussion Leaderboard Rules Team Submissions Host

### Overview

This project is part of the [introductory course](#) to data science and machine learning at [opencampus.sh](#).

The dataset was provided by meteolytix. For more than 10 years meteolytix was specializing on machine learning based sales forecasting, with models using more than 400 input variables and doing over one million predictions each day.

**Start**  
17 hours ago

**Close**  
a month to go

**Competition Host**  
Steffen

**Prizes & Awards**  
Kudos  
Does not award Points or Medals

**Participation**  
0 Entrants  
0 Participants  
0 Teams  
0 Submissions

**Tags**  
Mean Absolute Percentage Error

**Create**

Code

Discussions

Learn

More

Your Work

VIEWED

- Bakery Sales Predic...
- Misconfigured Bake...
- Bakery Sales Predic...
- How to delete an ho...
- Functions and Getti...

EDITED

- Exercise: Syntax, Va...

View Active Events

# BREAKOUT

- **Ladet den Test-Datensatz und die Sample Submission aus Kaggle herunter. Schaut Euch die Struktur der Datensätze an.**
- **Versucht eine Vorhersage für den Test-Zeitraum zu erstellen.**

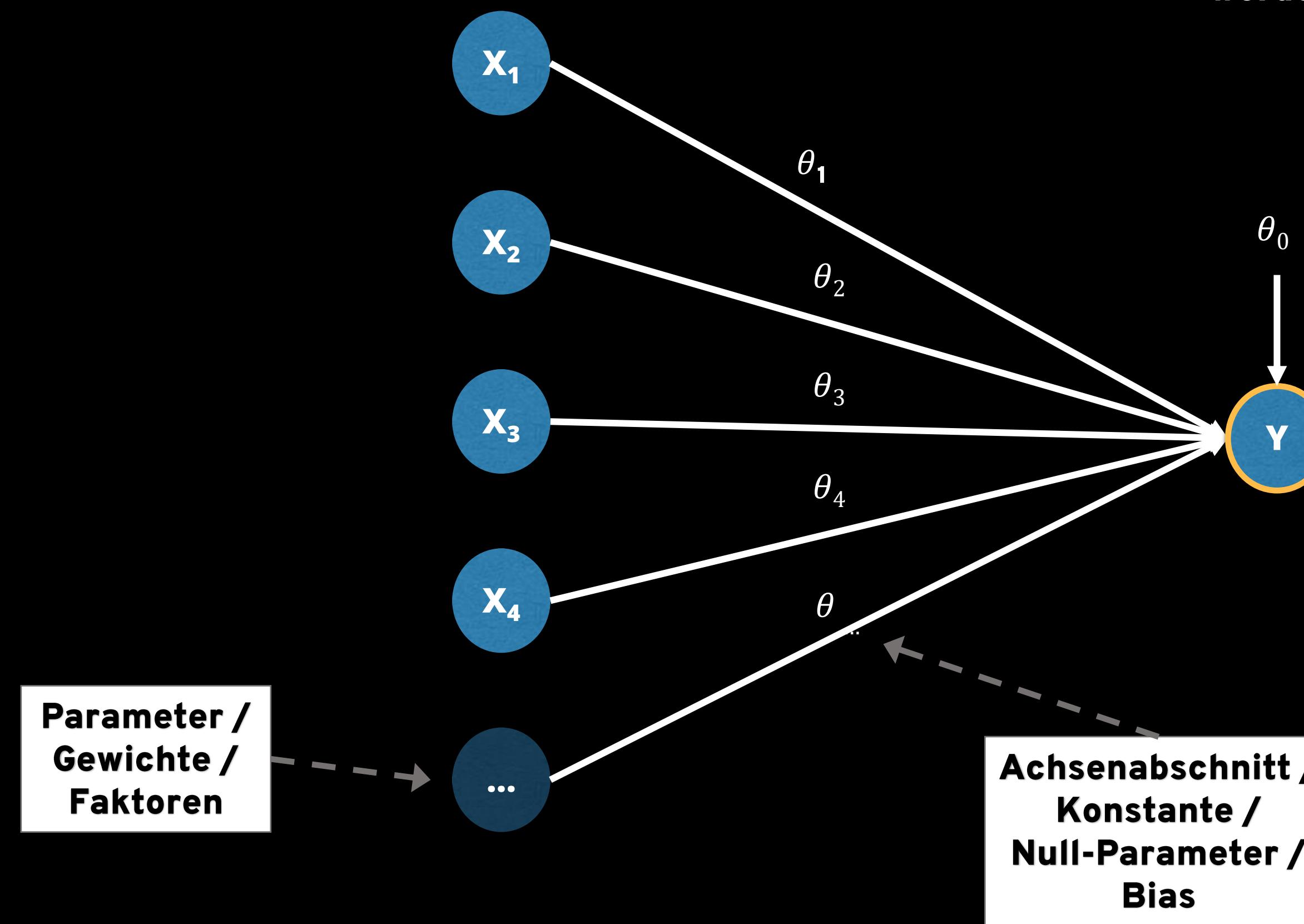
# ZUSAMMENFASSUNG LINEARE REGRESSION

## Input Layer

Elemente sind die Input-Variablen; auch genannt: Input-Features oder Input-Dimensionen.

## Output Layer

Nutzt eine „*Aktivierungsfunktion*“ (hier lineare Funktion) mit der die Parameter  $\theta$  der eingehenden Schicht zusammengefasst werden.



# EIGENSCHAFTEN DES LINEAREN MODELLS

- **Sowohl ohne als auch mit Regularisierung ist die Optimierung der Parameter für das lineare Modell sehr leicht und schnell möglich.**  
→ Für einfache Modelle ist es einfach optimierte Parameter zu erhalten.
- **Einfacher zu optimierende Modelle haben in der Regel stärkere Annahmen über die Zusammenhänge der Variablen (hier lineare Zusammenhänge).**  
→ Eine optimale Selektion und Kodierung der Variablen entsprechend der Annahmen ist umso wichtiger.  
→ Ggf. können die tatsächlichen Zusammenhänge nicht modelliert werden.

# NEURONALE NETZE

## Input Layer

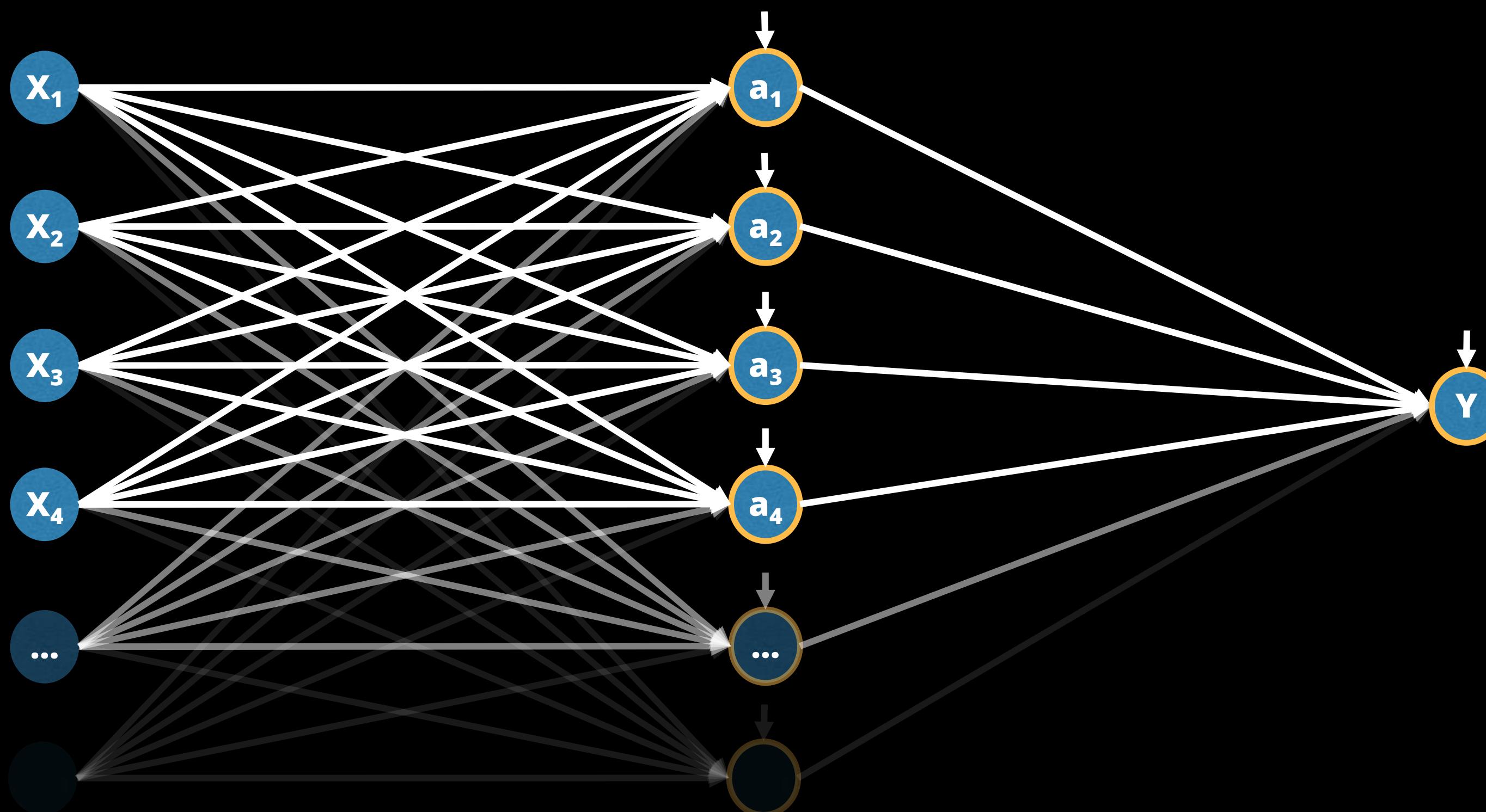
Besteht aus Input-Variablen/Features/Dimensionen

## Hidden Layer

Fasst mit Hilfe von Aktivierungsfunktionen und geschätzten Gewichten die Werte der vorherigen Schicht in jeweils einem Neuron zusammen.

## Output Layer

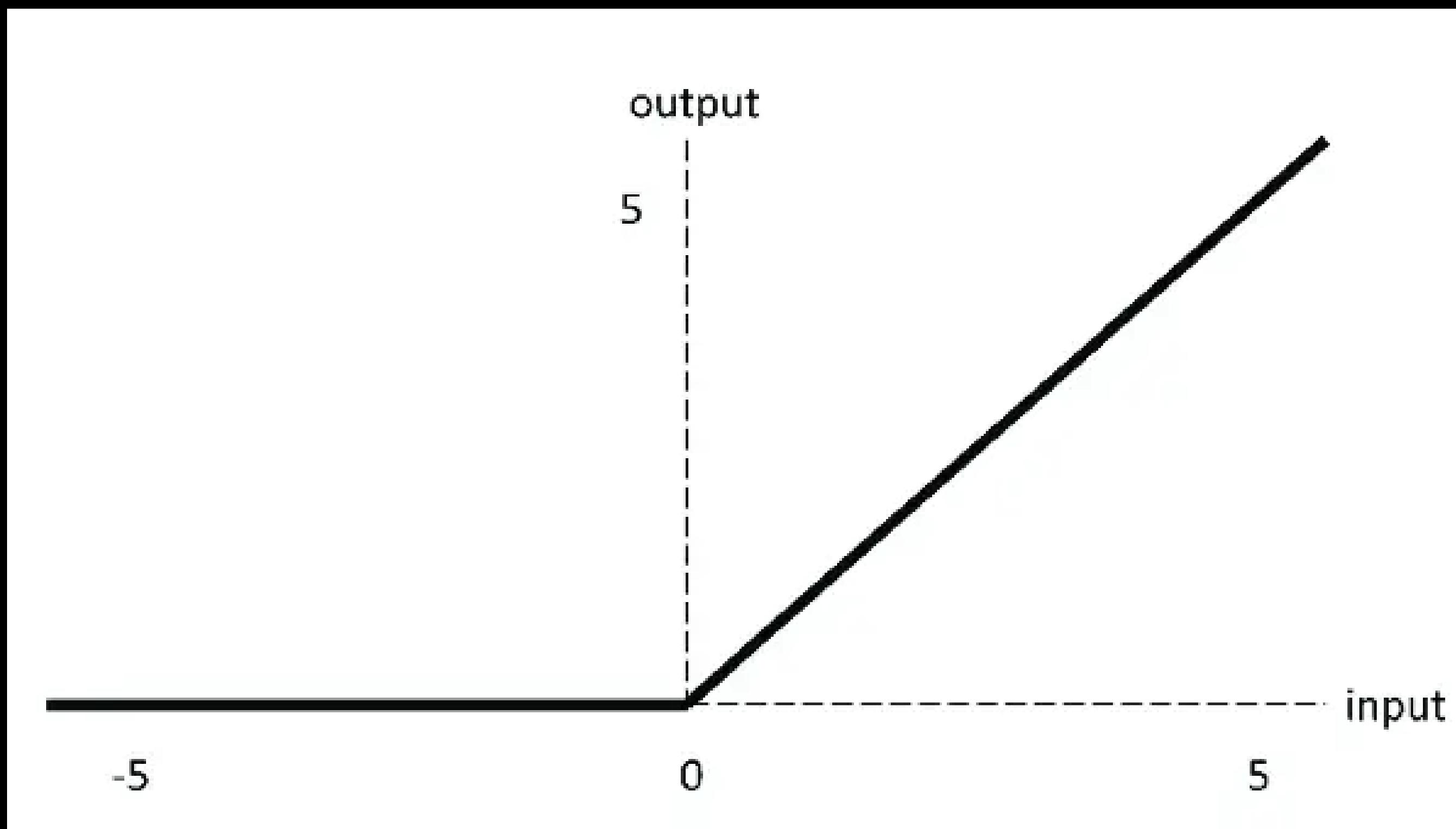
Fasst ebenfalls mit Hilfe von Aktivierungsfunktion und geschätzten Gewichten die Werte der vorherigen Schicht zusammen.



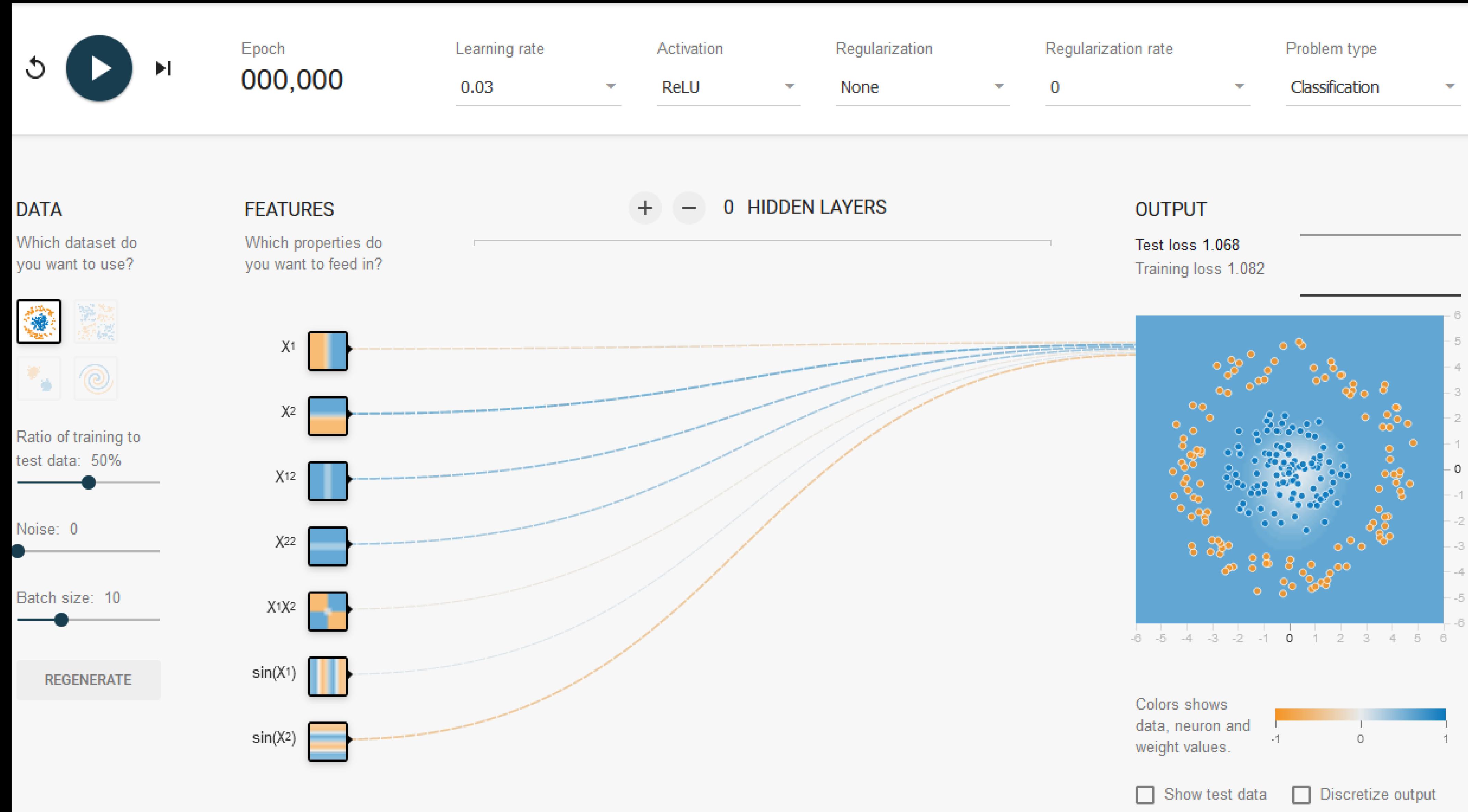
# EIGENSCHAFTEN DES NEURONALEN NETZES

- **Definition von zusätzlichen „Hidden Layern“ zwischen Input und Output Layer.**
  - **Statistik: Definition von latenten Variablen mit in der Regel unbekannter Bedeutung**
- **Nutzung nicht linearer Aktivierungsfunktionen.**
  - **Erlaubt Modellierung von Interaktionseffekten**
  - **Erlaubt Modellierung von nicht-linearen Effekten**

# DIE NICHT LINEARE AKTIVIERUNGSFUNKTION RELU



$$f(x) = \max(0, x)$$



# BREAKOUT

- Ruft folgendes Tool auf: <https://playground.tensorflow.org/>
- Definiert zwei Hidden Layer und probiert die Anzahlen der Neuronen so zu ändern, dass Ihr den spiralförmigen Datensatz vorhersagen könnt.
  - Habt ihr einen systematischen Ansatz gefunden, um zur Lösung zu kommen?
  - Inwieweit könnt Ihr das Ergebnis hinsichtlich der verwendeten Features interpretieren?

# HYPERPARAMETER

- **Wahl des Modells bzw. der Modellarchitektur**
- **Wahl der Aktivierungsfunktionen**
- **Wahl der Kostenfunktion**
  - MAE, MSE, ...
- **Wahl der Optimierungsfunktion**
  - Größe der Lernrate
- **Wahl der Batch-Größe**

**Je nach Modellarchitektur und gewählten Komponenten zahlreiche weitere...**

# LERNMATERIAL

- Dieses Video (12 Minuten) zur Einführung in Neuronale Netze an anschauen.

# AUFGABEN

- **Datensatz weiter um zusätzliche Variablen ergänzen, die für die Schätzung des Umsatzes relevant sein könnten.**
- **Die Vorhersagegüte Eures linearen Modells hier auf Kaggle testen!**
- **Mich zu Eurem Repo einladen und mir den Link via Mattermost schicken.**