23.05.24

OPENCAMPUS.sh

Einführung in Data Science und maschinelles Lernen

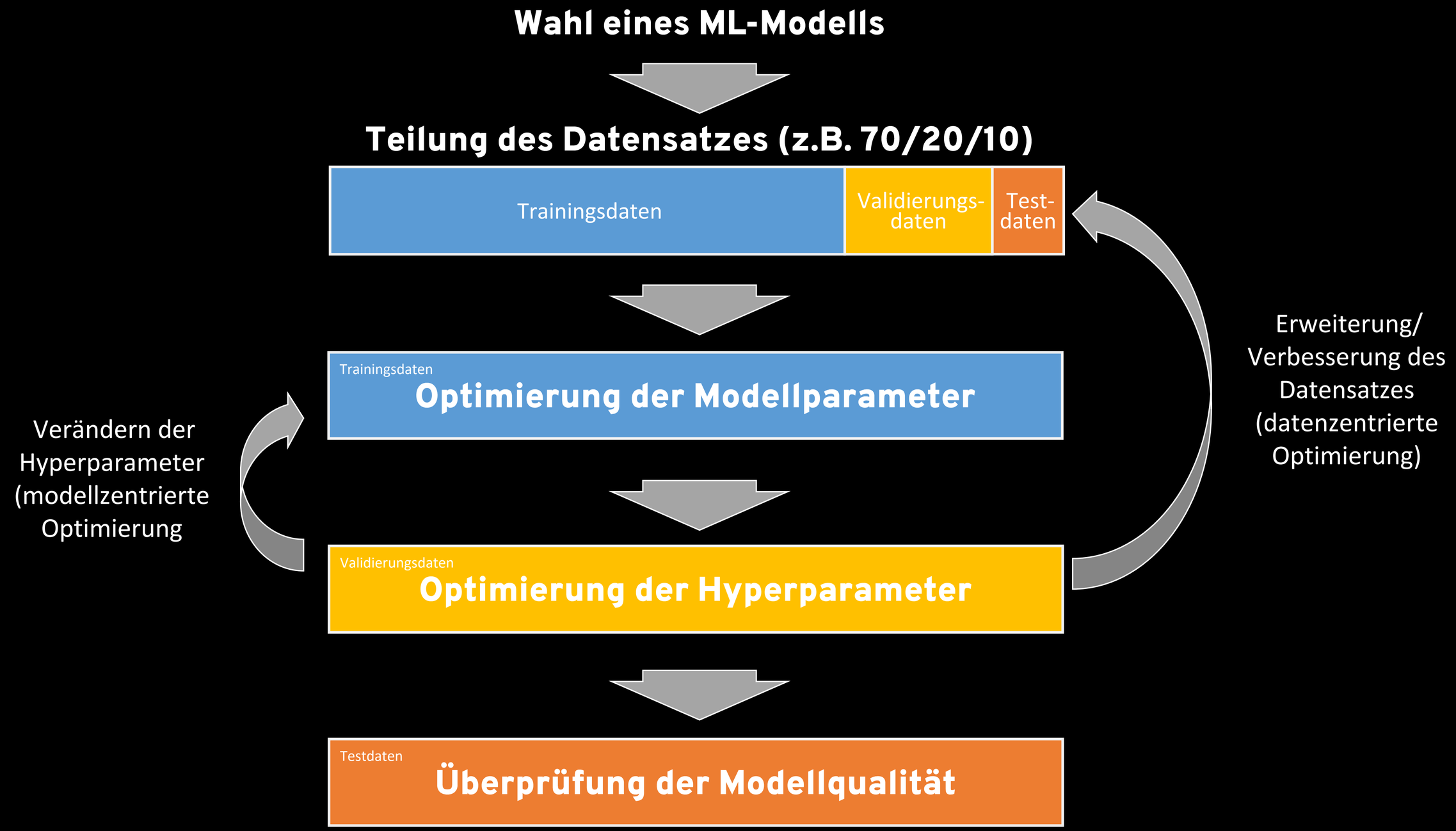
EINFÜHRUNG IN MASCHINELLES LERNEN

- Vorgehen zum Trainieren von maschinellen Lernalgorithmen
- Definition der linearen Regression
- Kostenfunktionen
- Optimierungsfunktionen

MÖGLICHE MODELLE FÜR DAS MASCHINELLE LERNEN

In diesem Kurs behandelte:

- **Lineares Modell**
- **Neuronales Netz**
- **(Support Vektor Maschine)**


Wahl eines ML-Modells
Teilung des Datensatzes (z.B. 70/20/10)
Trainingsdaten
Validierungs-daten
Test-daten
Trainingsdaten
Optimierung der Modellparameter
Verändern der Hyperparameter (modellzentrierte Optimierung
Validierungsdaten
Optimierung der Hyperparameter
Erweiterung/ Verbesserung des Datensatzes (datenzentrierte Optimierung)
Testdaten
Überprüfung der Modellqualität

TEILUNG EINES DATENSATZES (OHNE ZEITREIHEN!)

(1) Mischung der Reihenfolge (Randomisierung)

(2) Definition der Größen von Trainings-, Validierungs- und Testdatensatz in Prozent

(3) Teilung des Datensatzes

TEILUNG EINES DATENSATZES MIT ZEITREIHEN

(1) Ordnen der Zeitreihe

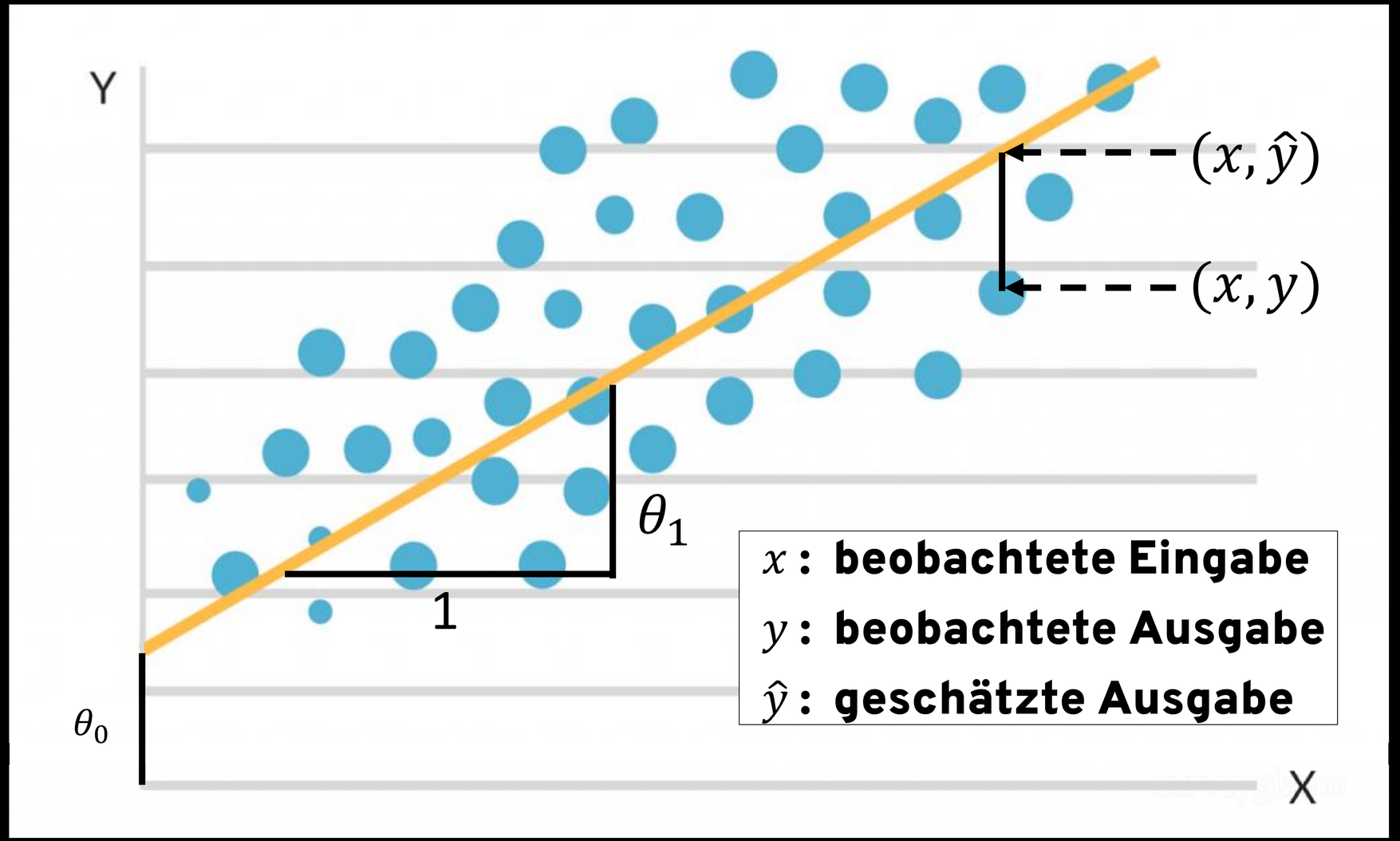
(2) Definition der Zeitreihenfenster für Validierungs- und Testdatensatz

(3) Teilung des Datensatzes

LINEARES MODELL

$$\hat{y} = \theta_0 + \theta_1 x$$

$$= h_x\,(\theta_0, \theta_1)$$



KOSTENFUNKTION (LOSS FUNCTION)

Zur Berechnung der Funktion mit den optimalen Parametern θ_0 und θ_1:

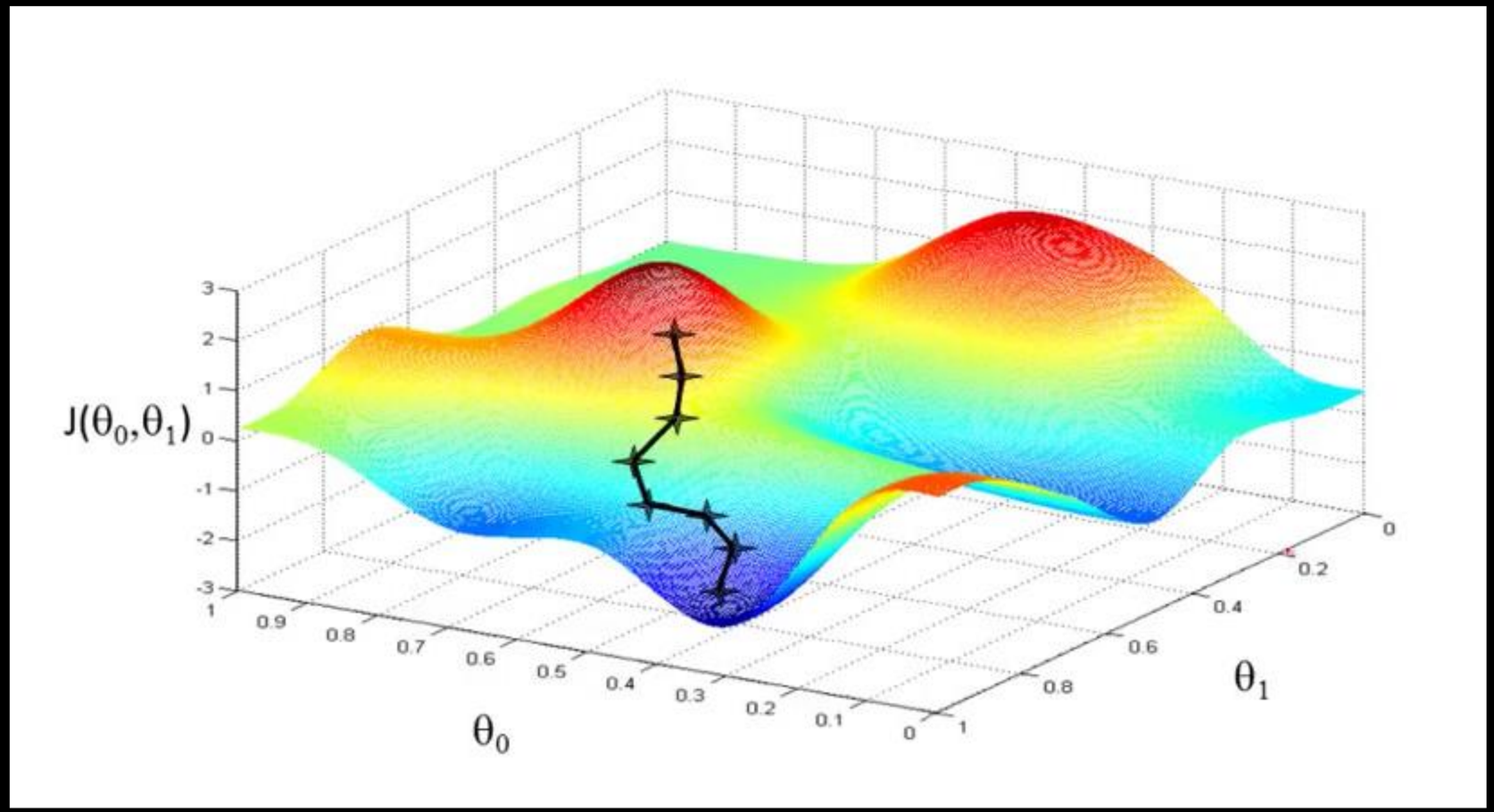
$$J_x(\theta_0, \theta_1) = \qquad h_x(\theta_0, \theta_1) - y$$

Mean Absolute Error (MAE)

$$J_x(\theta_0, \theta_1) = \frac{1}{m} \sum (h_x(\theta_0, \theta_1) - y)^2$$

Mean Squared Error (MSE)

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:**

```
Neuer Wert = Alter Wert - Lernrate × Partieller Gradient
```

MODELLPARAMETER VS. HYPERPARAMETER

Modellparameter

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

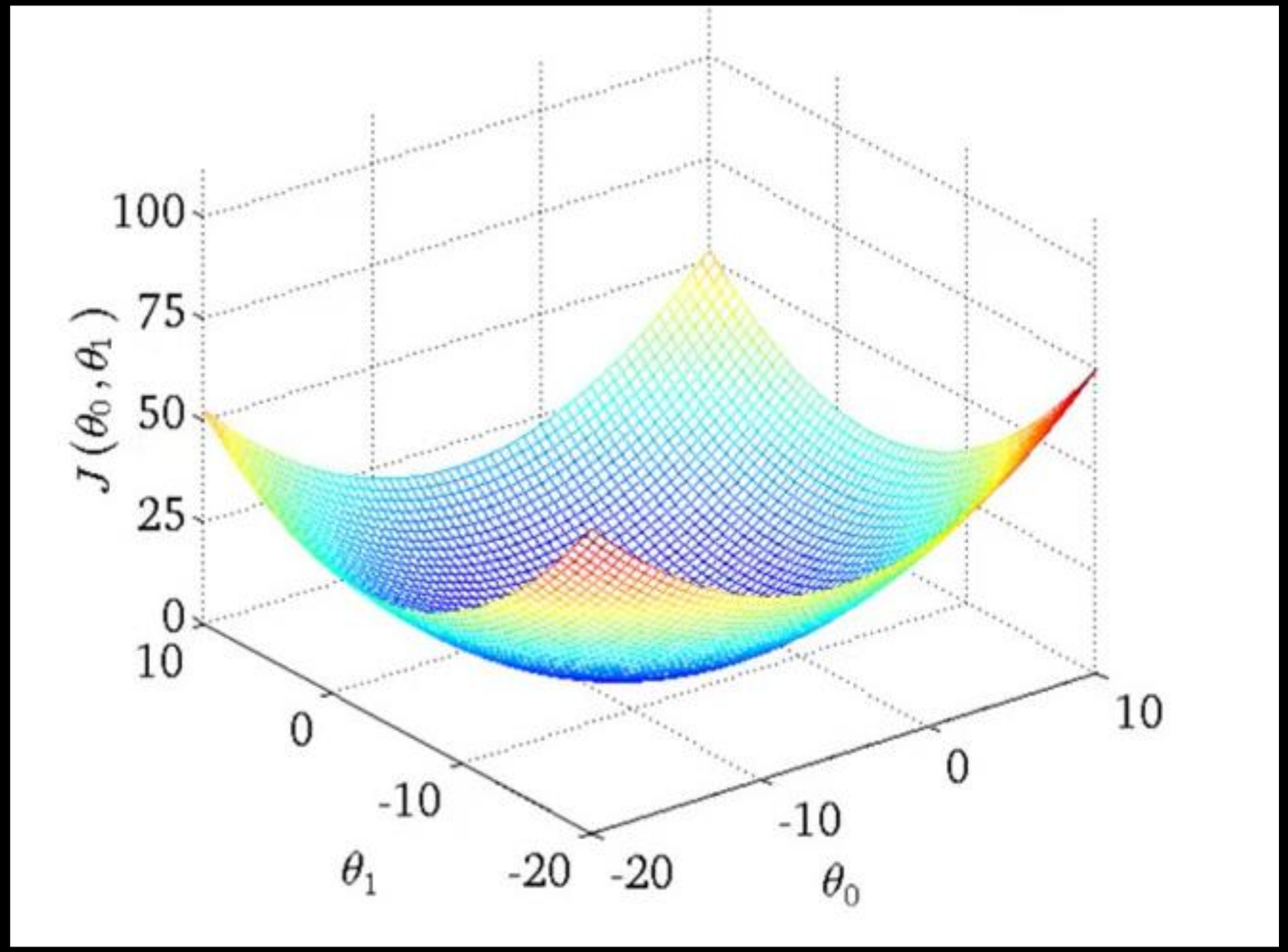
Hyperparameter

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

FEATURES, LABELS, PARAMETER

Machine Learning	Statistik	Beschreibung
Feature, Input Variable	Unabhängige Variable	Eingangsdaten, die für die Vorhersage genutzt werden
Label, Target Variable, Output	Abhängige Variable	Beobachtete Ergebnisse, anhand derer das Modell trainiert wird
Gewichte (Weights), (Modell-)Parameter	Koeffizienten	Werden im Rahmen des Trainings optimiert („gelernt“)

MINIMIERUNG BEI LINEAREN MODELLEN



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

- **Für lineare Modelle ist die Kostenfunktion konvex und besitzt keine lokalen Minima.**
- **Hier können auch andere statistische Verfahren als Gradient Descent genutzt werden, insbesondere wenn das Modell nur wenige Variablen umfasst.**

BEISPIEL EINES LINEAREN MODELLS

```
import pandas as pd

import statsmodels.formula.api as smf


# Load the dataset

url = "https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/house_pricing_data/house_pricing_train.csv"

house_pricing = pd.read_csv(url)


# Fit the linear model

mod = smf.ols('price ~ sqft_lot15 + C(condition)', data=house_pricing).fit()


# Print the summary

print(mod.summary())
```

ERGEBNIS DES LINEAREN MODELLS

OLS Regression Results

Dep. Variable:	price	R-squared:	0.014
Model:	OLS	Adj. R-squared:	0.014
Method:	Least Squares	F-statistic:	49.39
Date:	Thu, 23 May 2024	Prob (F-statistic):	5.82e-51
Time:	08:50:41	Log-Likelihood:	-2.4603e+05
No. Observations:	17290	AIC:	4.921e+05
Df Residuals:	17284	BIC:	4.921e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.261e+05	7.05e+04	4.625	0.000	1.88e+05	4.64e+05
C(condition)[T.2]	-2.305e+04	7.69e+04	-0.300	0.764	-1.74e+05	1.28e+05
C(condition)[T.3]	2.031e+05	7.06e+04	2.878	0.004	6.48e+04	3.41e+05
C(condition)[T.4]	1.8e+05	7.07e+04	2.546	0.011	4.14e+04	3.19e+05
C(condition)[T.5]	2.762e+05	7.12e+04	3.880	0.000	1.37e+05	4.16e+05
sqft_lot15	1.1382	0.103	11.002	0.000	0.935	1.341

Omnibus:	15642.173	Durbin-Watson:	2.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1054915.657
Skew:	4.122	Prob(JB):	0.00
Kurtosis:	40.368	Cond. No.	1.69e+06

...

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.69e+06. This might indicate that there are strong multicollinearity or other numerical problems.

Erklärte Varianz des Modells

Modell-parameter

Signifikanz bei p<.05

KENNWERTE DER REGRESSION

p-Wert

- Signifikanzwert
- Wahrscheinlichkeit, dass der zugehörige Wert in der Regression gleich null ist.
- Werte unter 0,05 (entspricht 5%) werden üblicherweise als signifikant betrachtet

(Adjustiertes) R^2

- Kennzahl zur Beurteilung der Güte einer Regression
- Wert zwischen 0 und 1, der der dem Anteil der erklärten Variation entspricht (1 entspricht 100%):

$$R^2 = \frac{\text{Erklärte Varianz}}{\text{Gesamtvarianz}}$$

- Das adjustierte R^2 bestraft das Hinzufügen zusätzlicher Variablen/Parameter

DataCamp Tutorial zur Linearen Regression:
https://www.datacamp.com/tutorial/essentials-linear-regression-python

BREAKOUT

- Definiert eine lineare Modellgleichung und führt mit Eurem Datensatz eine lineare Regression durch.

- Versucht das adjustierte R^2 des linearen Modells zu maximieren.

BEISPIEL DATENSATZTEILUNG (KEINE ZEITREIHE)

```python
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the dataset
url = "https://raw.githubusercontent.com/opencampus-sh/einfuehrung-in-data-science-und-ml/main/house_pricing_data/house_pricing_train.csv"
data = pd.read_csv(url)

# Set a random seed for reproducibility
random_state = 42

# First, split the data into training (70%) and remaining (30%)
train_data, remaining_data = train_test_split(data, train_size=0.7, random_state=random_state)

# Now split the remaining data into validation (2/3 of remaining) and test (1/3 of remaining)
validation_data, test_data = train_test_split(remaining_data, test_size=0.3333, random_state=random_state)

# Check the dimensions of the datasets
print("Training dataset dimensions:", train_data.shape)
print("Validation dataset dimensions:", validation_data.shape)
print("Test dataset dimensions:", test_data.shape)
```

BEISPIEL DATENSATZTEILUNG (ZEITREIHE)

```python
import pandas as pd

# Sample data
data = pd.DataFrame({
    'date': pd.date_range(start='2021-01-01', periods=100, freq='D'),
    'value': range(100)
})

# Ensure the data is sorted by date
data = data.sort_values(by='date')

print (data.head())

# Define your date thresholds
train_end_date = '2021-03-31'
validation_end_date = '2021-04-30'

# Convert to datetime if not already
data['date'] = pd.to_datetime(data['date'])

# Split the data based on the date thresholds
train_data = data[data['date'] <= train_end_date]
validation_data = data[(data['date'] > train_end_date) & (data['date'] <= validation_end_date)]
test_data = data[data['date'] > validation_end_date]

# Check the dimensions of the datasets
print("Training dataset dimensions:", train_data.shape)
print("Validation dataset dimensions:", validation_data.shape)
print("Test dataset dimensions:", test_data.shape)
```

BEISPIEL VORHERSAGE

```
# Create a new house with the following features
new_house = pd.DataFrame({
    'sqft_lot15': [5000],  # Square footage of lot
    'condition': [3]       # Overall condition of the house
})

# Use the model to predict the price of the new house
predicted_price = mod.predict(new_house)

print(f"The predicted price for the new house is: {predicted_price[0]}")
```

BASELINE MODELL

- **Sollte einfach zu implementieren sein, mit einer vernünftigen Chance, anständige Ergebnisse zu liefern und einer sehr geringen Wahrscheinlichkeit des Overfitting.**

- **Sollte interpretierbar sein, um das Verständnis für die Daten zu verbessern und ein besseres „Feature Engineering“ zu ermöglichen.**

Ameisen, E. (2018, March 6). *Always start with a stupid model, no exceptions.* Medium. https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa

LERNMATERIAL

- Schaut die Videos des Abschnitts „The problem of overfitting“ von Woche 3 des Kurses Supervised Machine Learning: Regression and Classification auf Coursera.

AUFGABEN

- Datensatz weiter um zusätzliche Variablen ergänzen, die für die Schätzung des Umsatzes relevant sein könnten.
- Euren Datensatz teilen in einen Trainingsdatensatz vom 01.07.2013 bis 31.07.2017 und einen Validierungsdatensatz vom 01.08.2017 bis 31.07.2018.
- Eine lineare Modellgleichung aufstellen, die das adjustierte R^2 für Euren Trainingsdatensatz maximiert.
- Im Verzeichnis „Baseline Model“ Eures Team Repositories die Berechnung für die lineare Regression dokumentieren.
- Einen Account bei Kaggle erstellen (mit der E-Mail, die ihr auch für EduHub genutzt habt).

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

View Active Events

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 pra
problem

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

kaggle

Search competitions

Filters

Create

Home

Competitions

Datasets

Models

Code

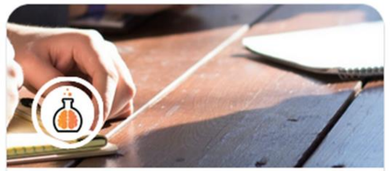
Discussions

Learn

More

View Active Events

All competitions | Featured | Getting Started | Research | Community | Playground | Simulations | Analytics



LLM - Detect AI Generated Text

Identify which essay was written by ...

Featured · Code Competition

1740 Teams

$110,000 2 months to go

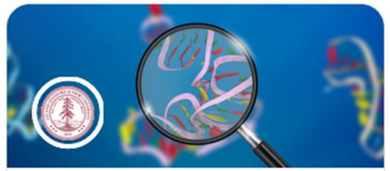
Open Problems – Single-Cell...

Predict how small molecules change...

Featured

1120 Teams

$100,000 3 days to go



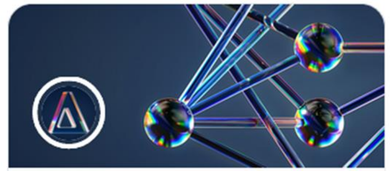
Stanford Ribonanza RNA Folding

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

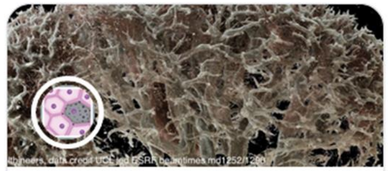
$100,000 23 days to go

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

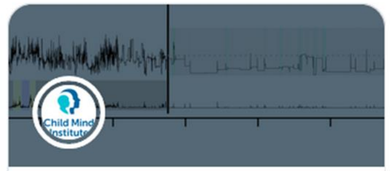
Linking Writing Processes to Writing...

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

kaggle

Search

Sign In | Register

Create

Home

Competitions

Datasets

Models

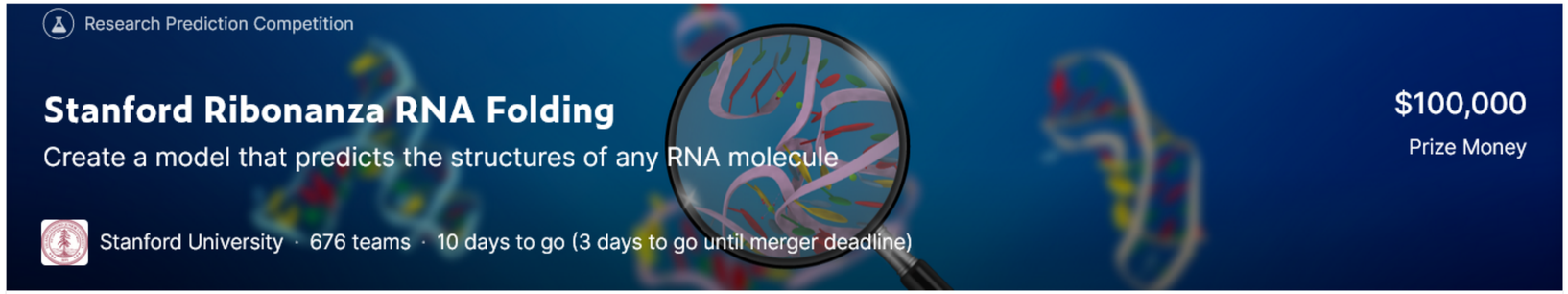
Code

Discussions

Learn

More

View Active Events



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	
2	DI		0.13773	108	3h	