

Sonderwochentag «Data Science»



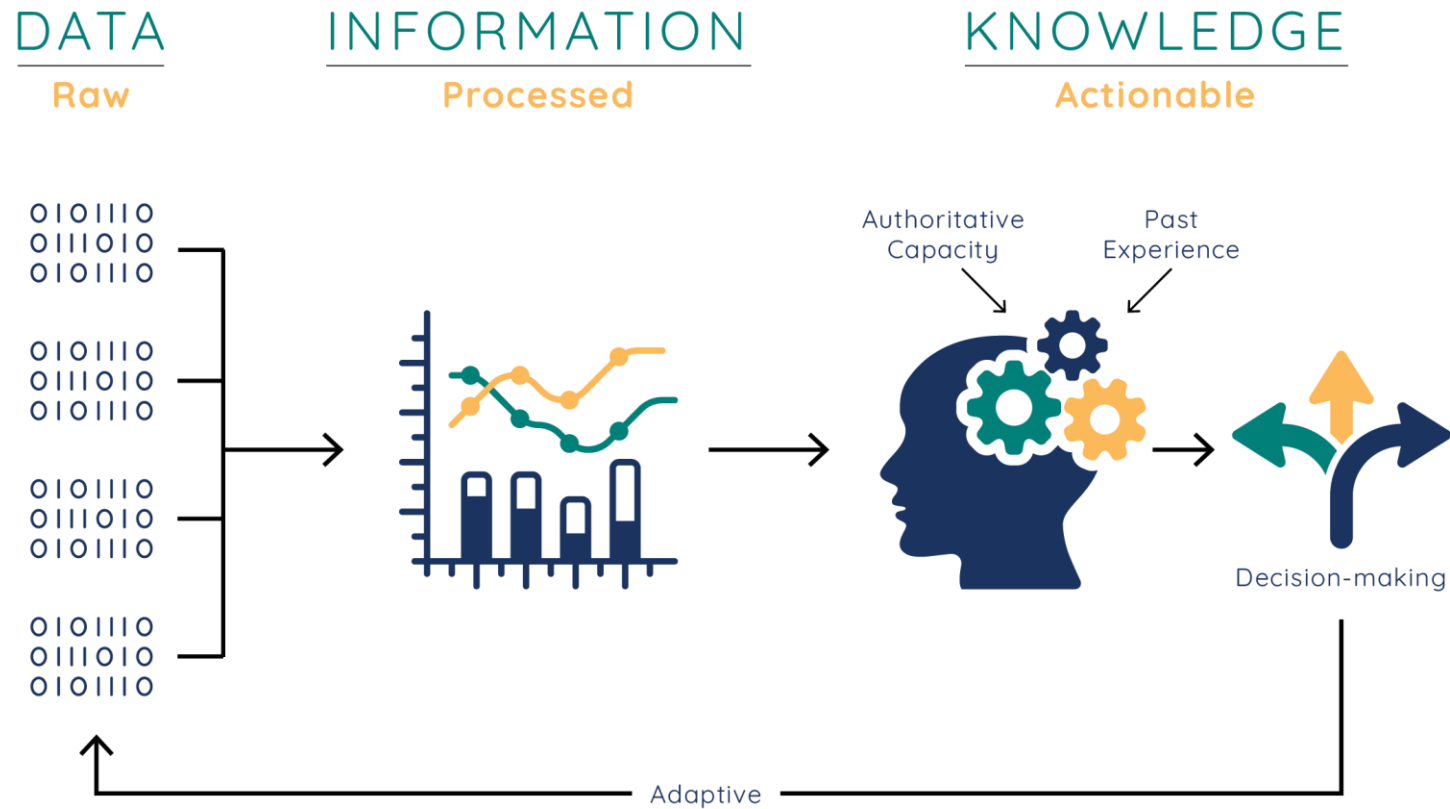
Dr. Philipp Hurni, Kantonsschule Sursee

Agenda

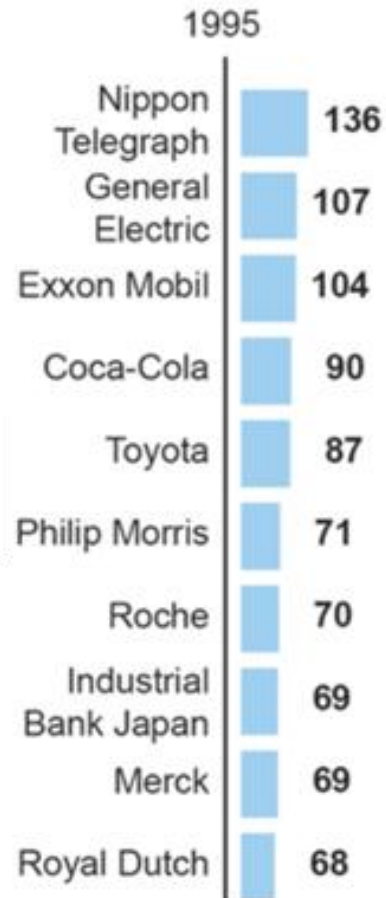
- Daten – das neue Öl
- Was ist Data Science?
 - Hackathons
 - OpenData
- Ausgewählte Methoden
 - Daten visualisieren (Matplotlib)
 - Schnittstellen (APIs)
 - Natural Language Processing (NLP)
 - Sentiment Analyse
 - Speech-to-Text
- Wettbewerb



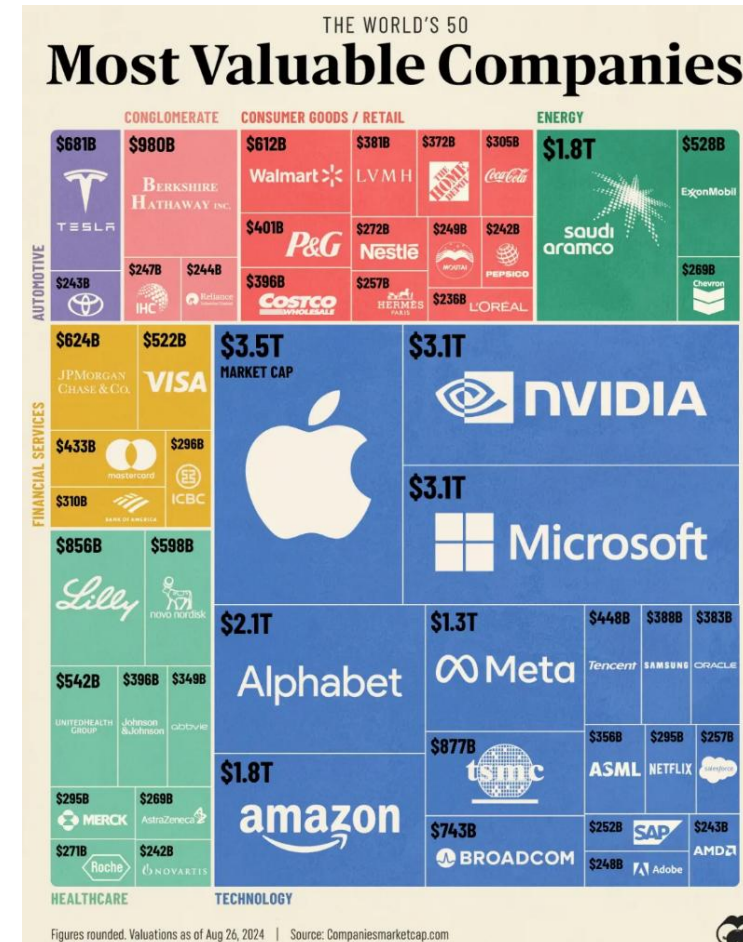
From Data to Knowledge – ein weiter Weg!



"IT/Data/Tech" hat die Wirtschaft "übernommen"



in Mrd. USD



Daten-basierte Geschäftsmodelle sind entstanden

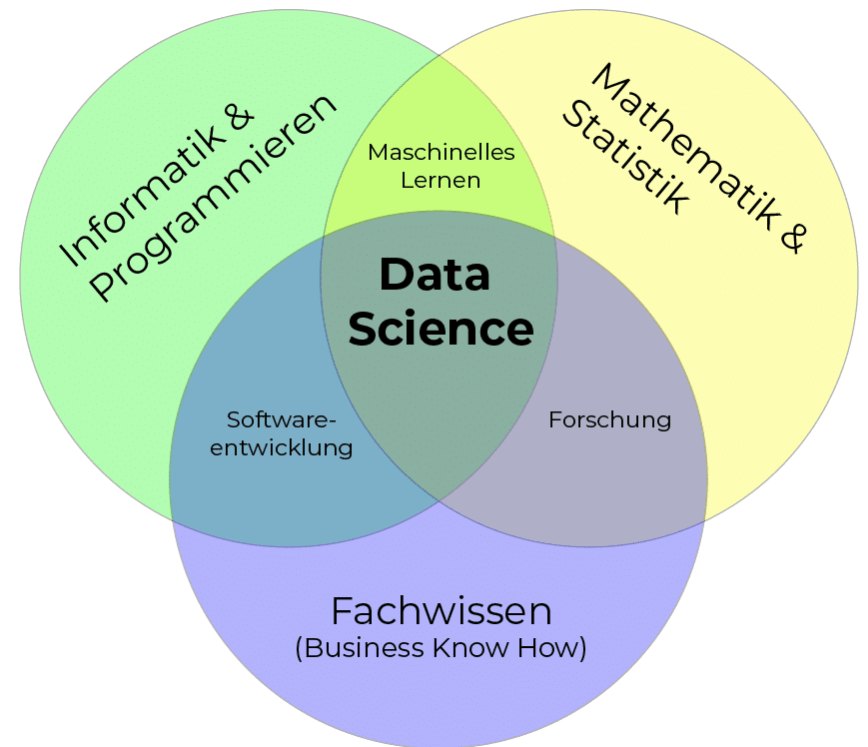
The Amazon logo, featuring the word "amazon" in a bold, black, sans-serif font. A curved orange arrow starts under the letter 'a' and points towards the letter 'z'.The Booking.com logo, featuring the word "Booking" in a white, sans-serif font inside a dark blue rounded square.The Netflix logo, featuring the word "NETFLIX" in a white, bold, sans-serif font with a black outline, set against a red background.

Was ist Data Science?

Data Science (Datenwissenschaft) ist eine interdisziplinäre Wissenschaft, die sich mit der Gewinnung von Wissen aus Daten befasst.

Die so genannte Datenwissenschaft generiert Informationen aus grossen Datenmengen, um daraus Handlungsempfehlungen abzuleiten.

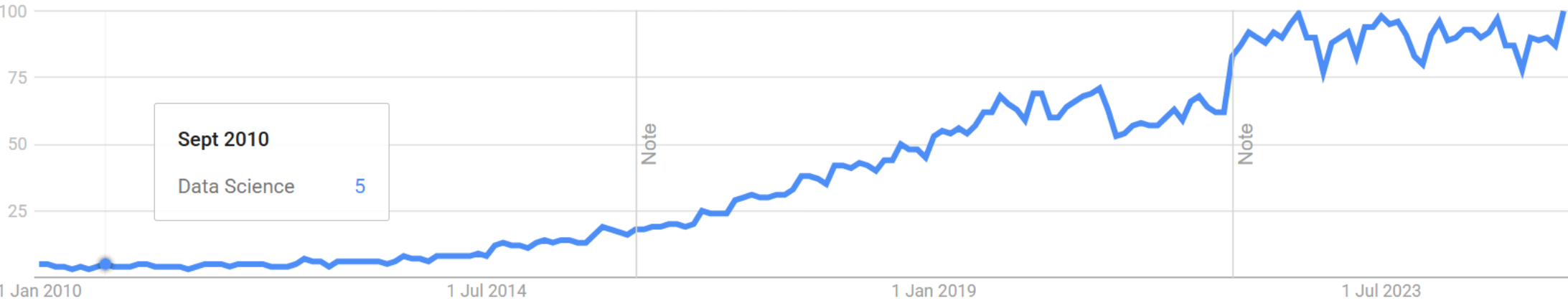
[[Wikipedia](#)]



Data Science / Data Scientist

data science
Search term

Interest over time



Universitäten



Universität
Zürich UZH

Wirtschaftswissenschaftliche Fakultät

u^b

b
UNIVERSITÄT
BERN

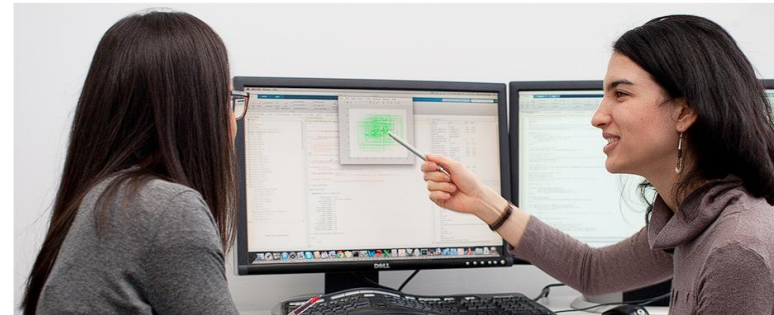
Master Statistik und Data Science

Data Science

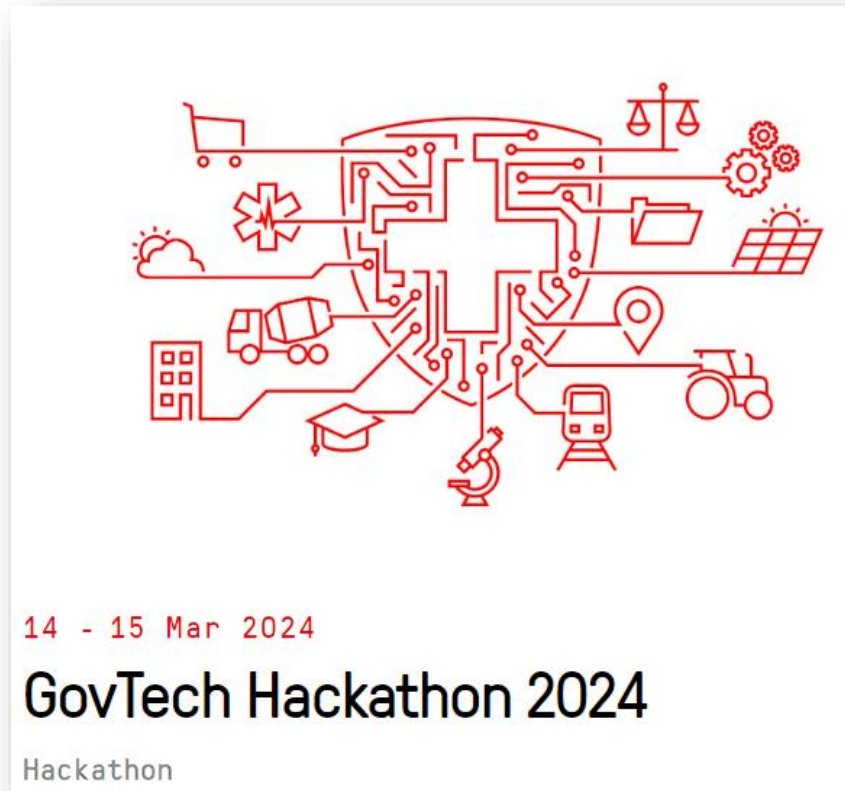


ETH zürich

Master Data Science



Hackathons



<https://swiss-ai-weeks.ch/>



https://www.youtube.com/watch?v=6_0PpEDE4jg

Öffentliche Datenquellen



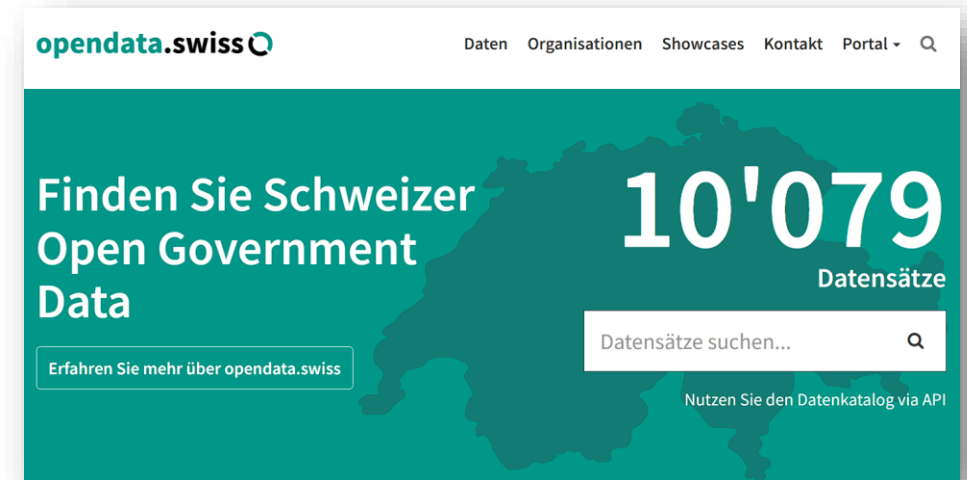
Philipp Hurni, Kantonsschule Sursee

OpenData

Der Staat erhebt bereits heute viele Daten, unter Verwendung von Steuergeldern

Diese Daten sollen von privaten und kommerziellen Akteuren in der Schweiz genutzt werden können

Übersicht: www.opendata.swiss
<https://github.com/rnckp/awesome-ogd-switzerland>



Parlament

Parlamentarische Beratungen,
Abstimmungen, etc.

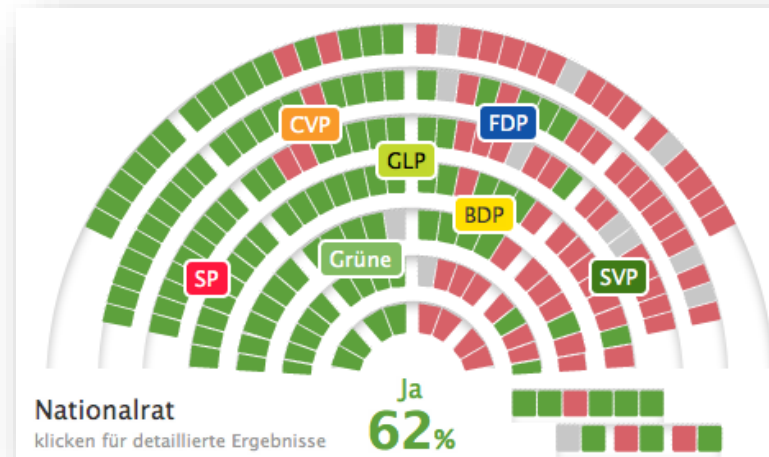
<http://ws-old.parlament.ch>

Parlamentsdienste
Services du Parlement
Servizi del Parlamento
Servetschs dal parlament



Projektmanagement
CH-3003 Bern
www.parlament.ch

Informationen über die WebServices der
Parlamentsdienste



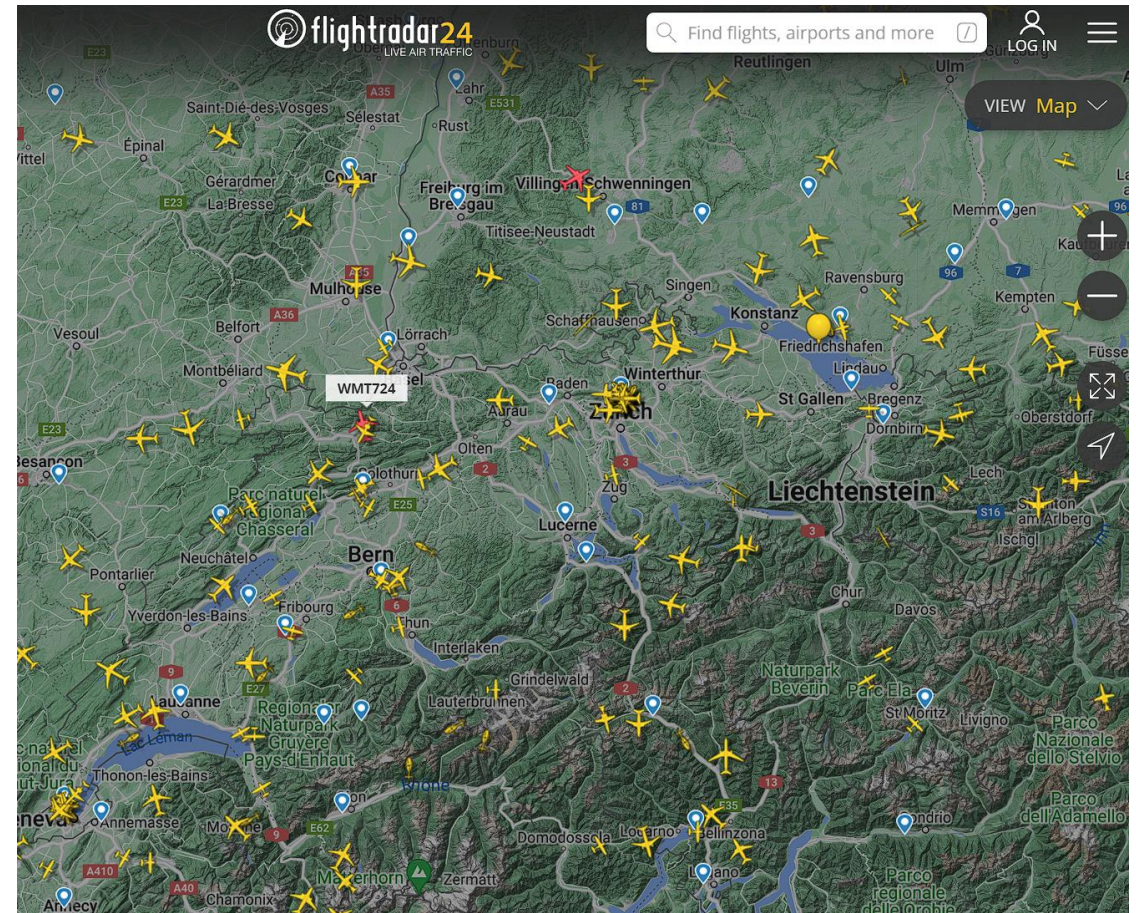
Flightradar 24

Near-realtime Information von
Flugbewegungen in aller Welt

<https://www.flightradar24.com/>

Schnittstelle für den Zugriff

<https://github.com/JeanExtreme002/FlightRadarAPI>



Flightradar 24

```
SucheFlugzeugeUeberPosition.py x
1 from FlightRadar24 import FlightRadar24API
2 from FlightRadar24.entities import Entity
3 import time
4 import os
5 import math
6
7 # ABFRAGEBEREICH
8 BREITENGRAD = 47.1786 # Breitengrad Sursee
9 LAENGENGRAD = 8.1064 # Längengrad Sursee
10
11 #BREITENGRAD = 47.4496 # Breitengrad Kloten
12 #LAENGENGRAD = 8.5822 # Längengrad Kloten
13
14 RADIUS_KILOMETER = 20 # Radius in KM
15 AKTUALISIERUNGS_INTERVALL_SEKUNDEN = 3 # Wie oft wird geprüft welche Flugzeuge da sind?
16
17 def main():
18     # Zugang zur FlightRadar-Schnittstelle
19     flugradar_api = FlightRadar24API()
20     # Begrenzung berechnen mit der API
21     begrenzung_str = flugradar_api.get_bounds_by_point(BREITENGRAD, LAENGENGRAD, int(RADIUS_KILOMETER * 1000))
22
23     # Prüfe intervallweise
24     while True:
25
26         # hier holen wir die Flüge und speichern sie in der Liste "fluege"
27         fluege = flugradar_api.get_flights(bounds=begrenzung_str)
28         # wir gehen nun durch alle Flüge durch und drucken diese als Tabelle
29         if len(fluege) > 0:
30
31             # Tabellentitel
32             print(f"Flugnr | Flugzeugtyp | Flug Start | Flug Ziel | Breite | Länge | Höhe (ft) | Distanz (km)")
33             print("-----|-----|-----|-----|-----|-----|-----|-----")
34             for fluege in fluege:
35                 print(f"{fluege.callsign} | {fluege.type} | {fluege.position.timestamp} | {fluege.position.destination} | {fluege.position.latitude} | {fluege.position.longitude} | {fluege.position.altitude} | {fluege.distance}")
36
37             print(f"4 Flugzeuge gefunden")
38
39 if __name__ == '__main__':
40     main()
```

```
Shell x
>>> %Run SucheFlugzeugeUeberPosition.py
Flugnr | Flugzeugtyp | Flug Start | Flug Ziel | Breite | Länge | Höhe (ft) | Distanz (km)
-----|-----|-----|-----|-----|-----|-----|-----
HBZVK | G2CA | unknown | unknown | 47.14550 | 7.87960 | 4842 | 17.54
UAL416 | B788 | Chicago | Mailand | 47.17010 | 7.91320 | 36975 | 14.63
HBZP2 | B06 | Emmen | unknown | 47.15940 | 8.13910 | 3675 | 3.27
IFA6536 | GLEX | Samarkand | Luxemburg | 47.06070 | 8.18070 | 39975 | 14.26
4 Flugzeuge gefunden
```

NewsAPI

API für Zugriff auf
unterschiedliche News-Quellen

<https://newsapi.org/>

Beispiel: BBC UK

```
1 # importing requests package
2 import requests
3
4 def NewsFromBBC():
5
6     # BBC news api
7     # following query parameters are used
8     # source, sortBy and apiKey
9     query_params = {
10         "source": "bbc-news",
11         "sortBy": "top",
12         "apiKey": "4dbc17e007ab436fb66416009dfb59a8"
13     }
14     main_url = " https://newsapi.org/v1/articles"
15
16     # fetching data in json format
17     res = requests.get(main_url, params=query_params)
18     open_bbc_page = res.json()
19
20     # getting all articles in a string article
21     article = open_bbc_page["articles"]
22
23     # empty list which will
24     results = []
25
26     for ar in article:
27         print("Author: ", ar["author"], "-", " Title: ", ar["title"], "-", " URL: ", ar["url"])
28
29
30 # Driver Code
31 if __name__ == '__main__':
32
33     # function call
34     NewsFromBBC() |
```


Daten verarbeiten und visualisieren

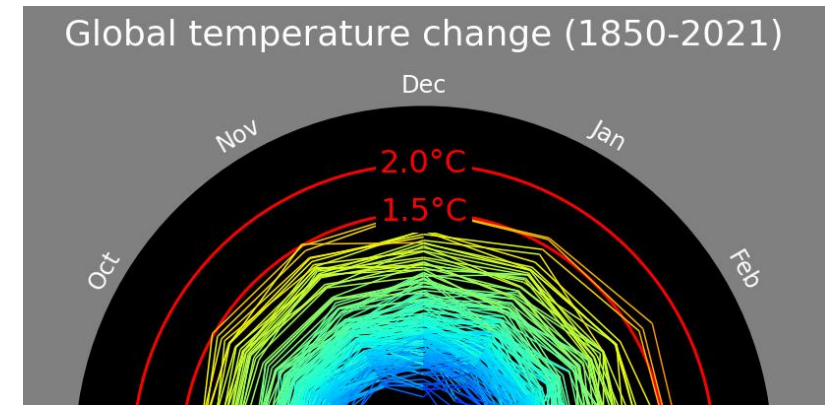


Philipp Hurni, Kantonsschule Sursee

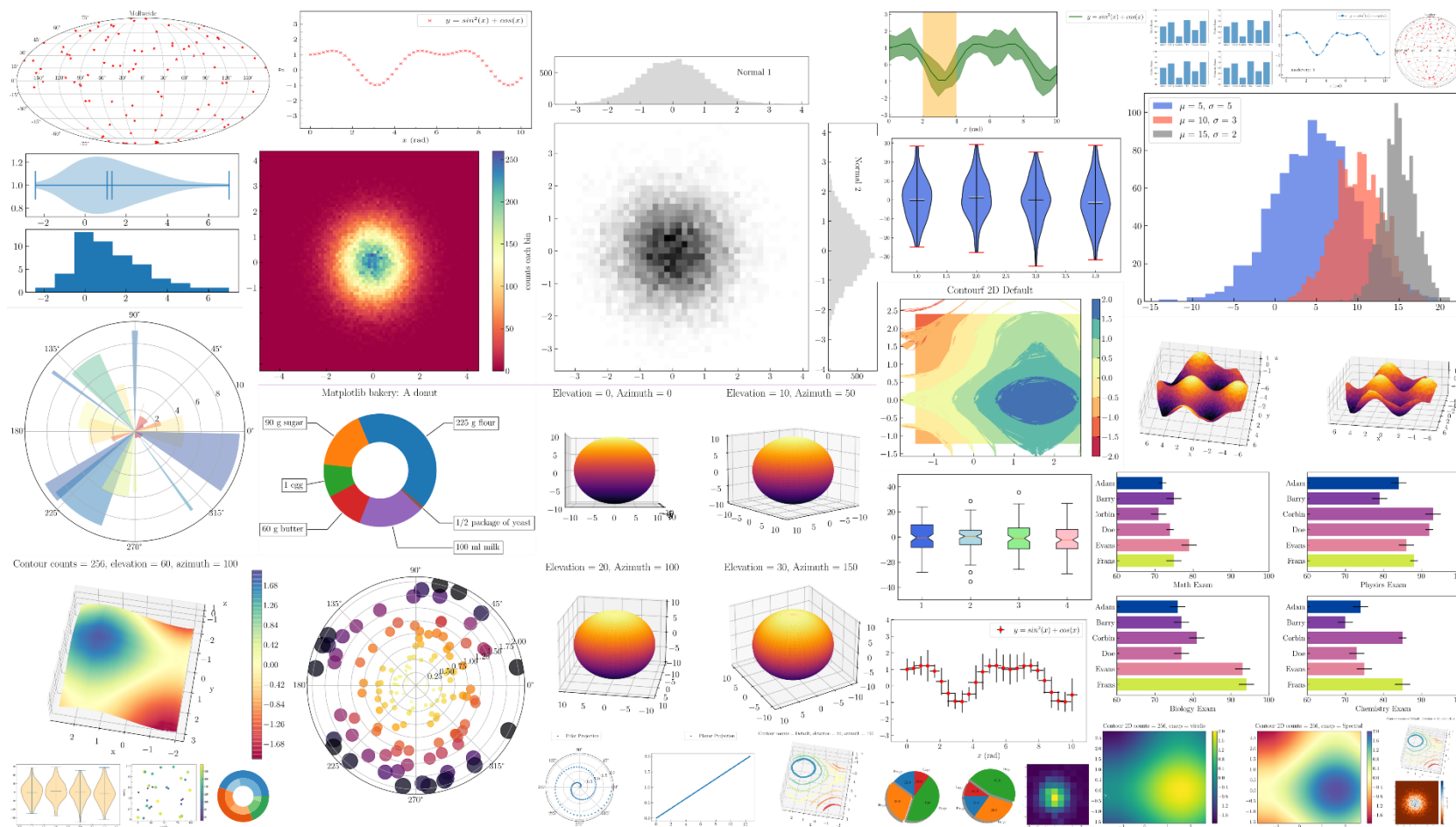


<https://vorlesungsverzeichnis.unibas.ch/de/vorlesungsverzeichnis?id=282426>
<https://ksl.unibe.ch/KSL/detailsansicht?2>
<https://ksl.unibe.ch/KSL/kurzansicht?3&stammNr=420844&semester=HS2025&lfidNr=0>

- Bibliothek in Python zum visuellen darstellen von Daten
- Sehr grosses Framework mit vielen Möglichkeiten
- Heute unverzichtbar in naturwissenschaftlichen und oft auch sozialwissenschaftlichen Disziplinen
- 1./2. Semester von vielen Studiengängen heute führen in Matplotlib ein

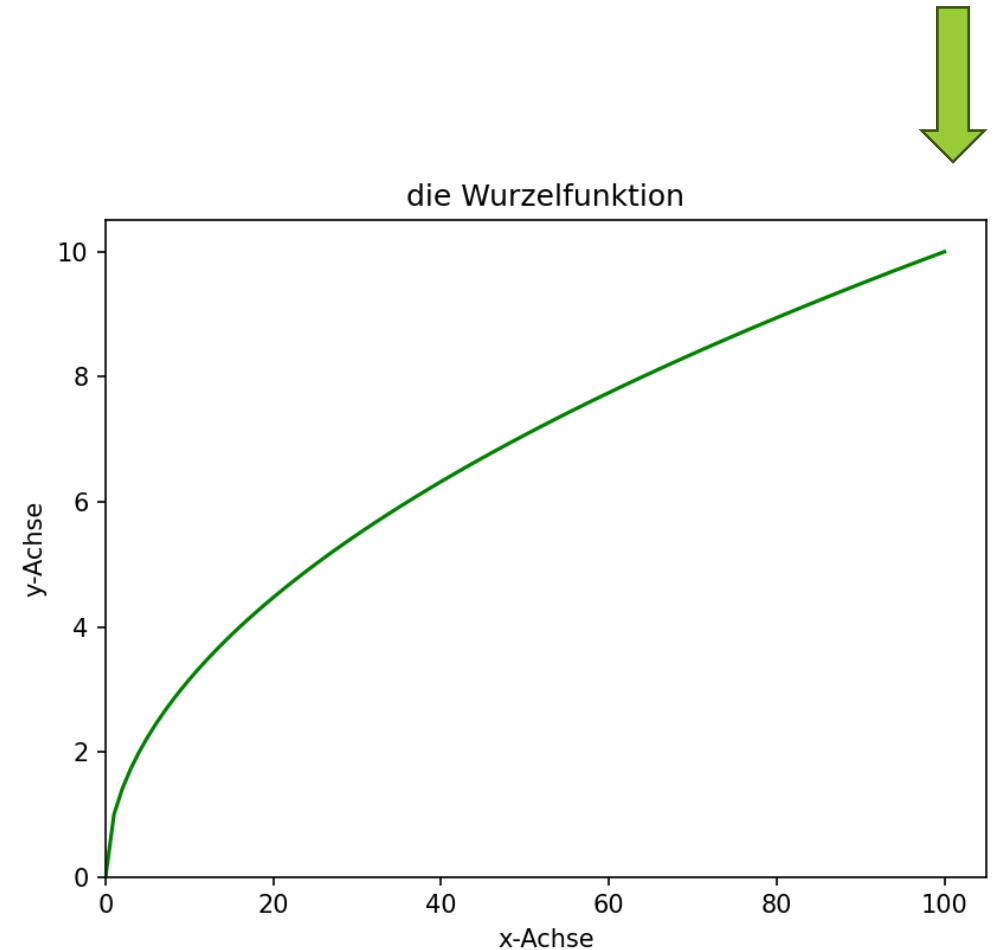


Matplotlib



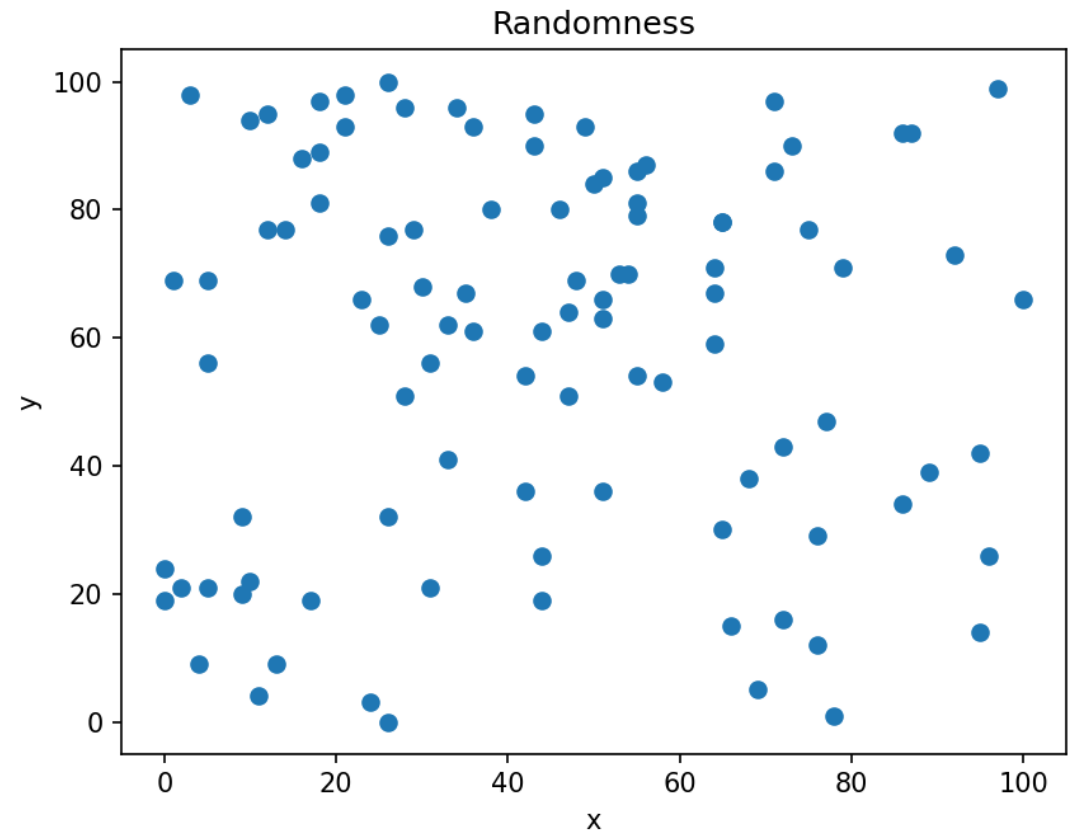
Übungen mit Matplotlib

- Du hast Matplotlib A.py
- Zeichne zusätzlich die Logarithmusfunktion $\text{Log}_{10}(x)$
- Spiele mit der y-Achse (log-view)
- Stelle sicher, dass nur der Ausschnitt x in $[0,100]$ und y in $[0,10]$ dargestellt wird



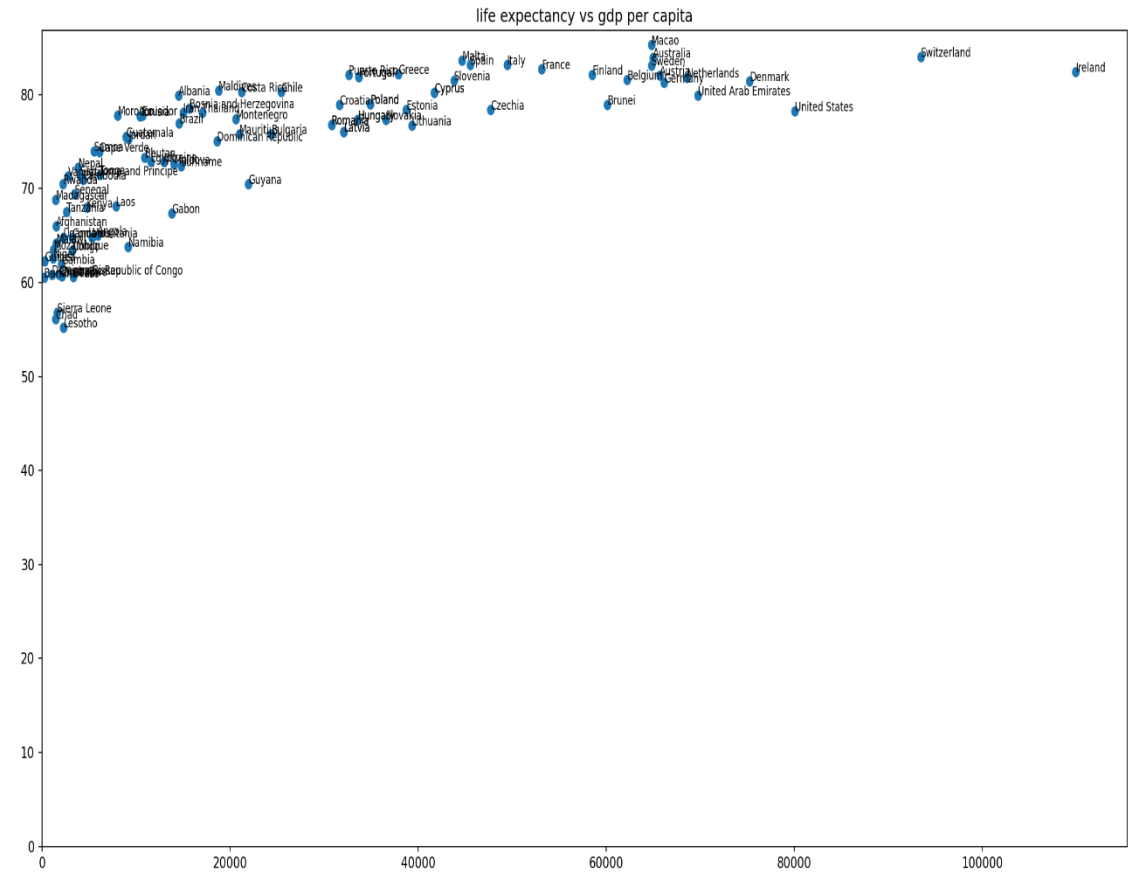
Übungen mit Matplotlib

- Du hast Matplotlib B.py – hier wird ein ScatterPlot angewendet
- Schaue dir die Daten in countries.csv an. Als Wissenschaftler – wie würdest du die Daten darstellen?
- Versuche das in Matplotlib zu bewerkstelligen.



Übungen mit Matplotlib

- In Matplotlib C wird auf eine zweite CSV-Datei zugegriffen
- Mach eine Hypothese, und versuche diese grafisch zu stützen, oder zu verwerfen
- Beispiel: je reicher ein Land, umso höher die Lebenserwartung



Beispiel-Anwendungen von Data Science in der Finanzwelt



Data Science Prozess

1. Collect (Sammeln): Daten aus verschiedenen Quellen erfassen.

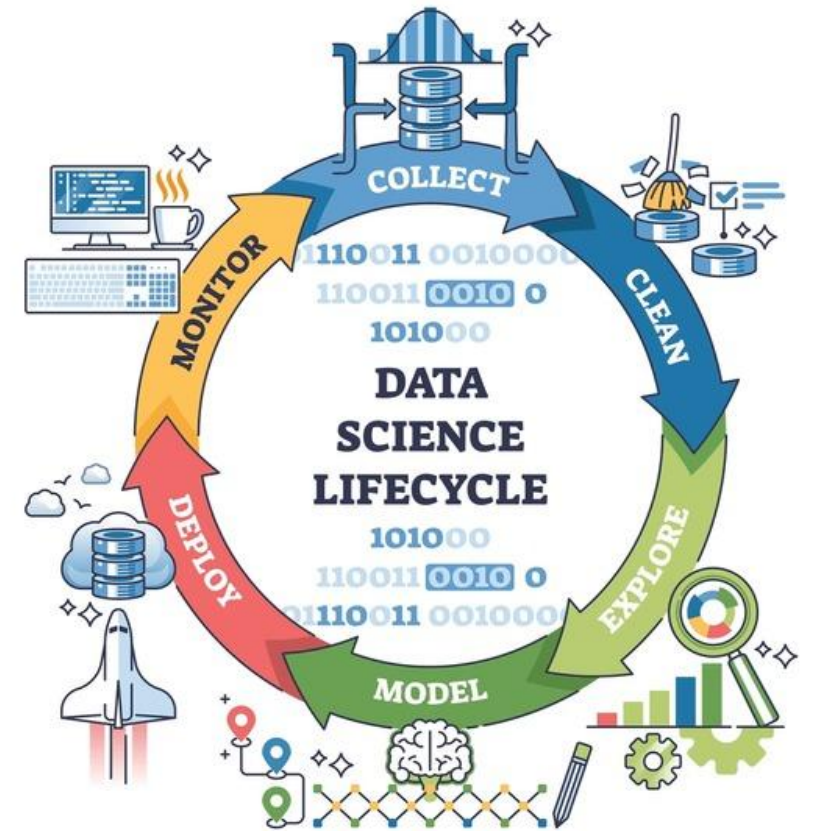
Hier werden Rohdaten in die Umgebung geholt, sei es aus Datenbanken, APIs, Protokolldateien oder externen Quellen.

2. Clean (Bereinigen): Die gesammelten Daten vorbereiten, bereinigen und transformieren.

Rohdaten sind oft unvollständig, fehlerhaft oder inkonsistent. Fehlende Werte müssen "gefüllt" werden, Duplikate entfernt, etc. Das kann sehr zeitaufwändig sein

3. Explore (Erforschen): Daten müssen analysiert werden, um Muster, Trends und Anomalien zu verstehen.

Mithilfe von statistischen Methoden und Visualisierungen werden Einblicke in die Daten gewonnen, Hypothesen formuliert und die Relevanz der Erkenntnisse für das Problem bewertet.

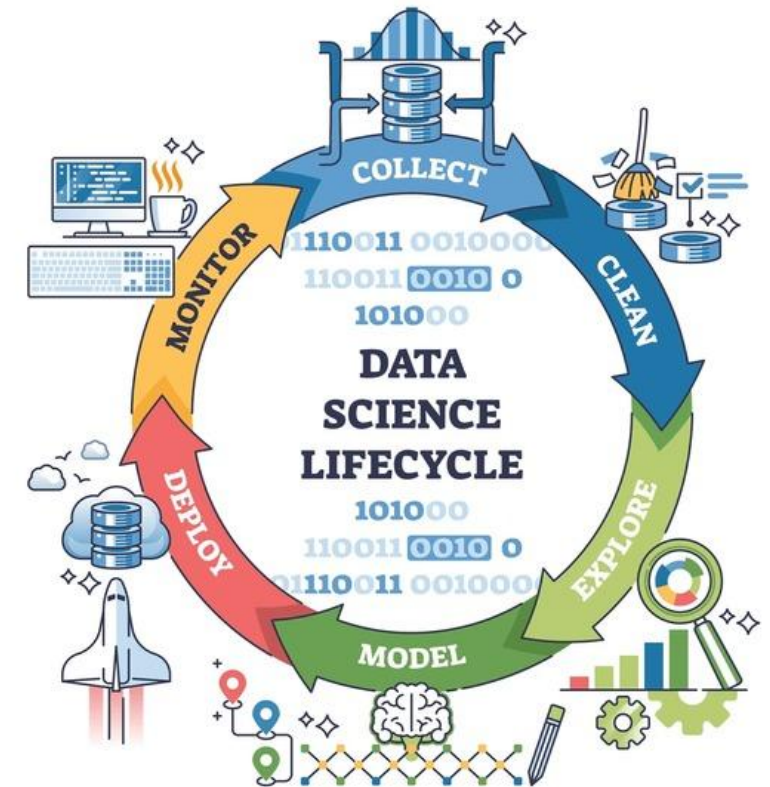


Data Science Prozess

4. Model (Modellieren): Algorithmen und Techniken anwenden, um ein Vorhersagemodell oder ein beschreibendes Modell zu erstellen. Dies umfasst die Auswahl des geeigneten Algorithmus (z. B. Regression, Klassifikation, Clustering), das Training des Modells mit den vorbereiteten Daten und die Abstimmung der Modellparameter,

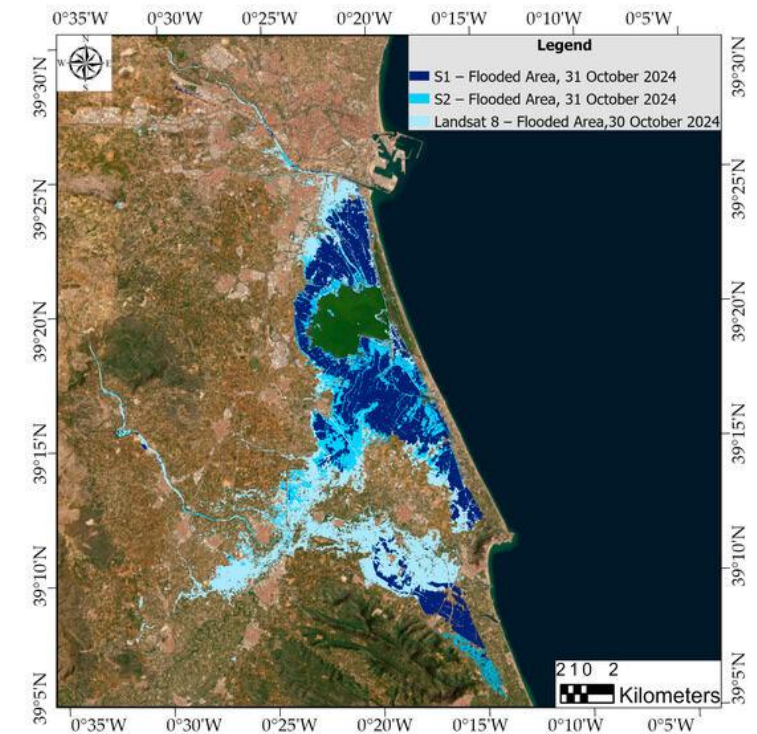
5. Deploy (Implementieren): Das fertige Modell in einer Produktionsumgebung einsatzbereit machen. Das Modell wird in die reale Welt gebracht, oft integriert in eine Anwendung, ein Dashboard oder ein automatisiertes System.

6. Monitor (Überwachen): Die Leistung des im Einsatz befindlichen Modells kontinuierlich verfolgen und bewerten. Modelle können im Laufe der Zeit durch sich ändernde Datenmuster an Genauigkeit verlieren.



Data Science Use Cases: Satellitenbilder

Parametrische Versicherung:
Versicherungsschäden rasch beziffern und
begleichen – ohne menschliches Zutun



Data Science Use Cases: High Frequency Trading

Börsenhandel

```
stockPriceFetcher.py x
1 import yfinance as yf
2 import time
3 from datetime import datetime
4
5
6 UPDATE_INTERVAL_SECONDS = 5 # Aktualisierungsintervall in Sekunden (alle 5s)
7 STOCK_TICKER="LMND"
8
9 while True:
10     # 1. Daten abrufen
11     # hole die Aktie
12     stock = yf.Ticker(STOCK_TICKER)
13     info = stock.info
14
15     # Preis extrahieren
16     current_price = info.get('currentPrice', 'N/A')
17
18     # Zeitstempel
19     current_time = datetime.now().strftime("%H:%M:%S")
20     print(current_time,"| Aktueller Preis: ", current_price)
21     time.sleep(UPDATE_INTERVAL_SECONDS)
22
23
```



Data Science Use Cases: High Frequency Trading

Krypto

```
cryptoFetcher.py x
1 import ccxt
2 import time
3 from datetime import datetime
4
5
6 UPDATE_INTERVAL_SECONDS = 1 # Aktualisierungsintervall in Sekunden
7 CRYPTO_PAIR = 'BTC/USDT'
8 EXCHANGE_ID = 'binance'
9
10 while True:
11     # Daten abrufen
12     exchange = getattr(ccxt, EXCHANGE_ID)()
13     exchange.load_markets()
14
15     ticker = exchange.fetch_ticker(CRYPTO_PAIR)
16     current_price = ticker['last']
17
18     # Zeitstempel
19     current_time = datetime.now().strftime("%H:%M:%S")
20     print(current_time, "| Aktueller Preis: ", current_price)
21     time.sleep(UPDATE_INTERVAL_SECONDS)
```



Data Science Use Cases: Satellitenbilder



Geospatial Detection Features



Counting car groupings



Supply chain performance and cargo loading



Weather & climate Impact



Damage assessment

Natural Language Processing



Philipp Hurni, Kantonsschule Sursee

Natural Language Processing

Die Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) ist ein Teilbereich der Künstlichen Intelligenz. Sie soll Computer in die Lage versetzen, menschliche Sprache zu verstehen, zu interpretieren und zu manipulieren.

NLP will die Lücke zwischen menschlicher Kommunikation und den Sprachverarbeitungsfähigkeiten von Computern schliessen und greift dazu auf eine Vielzahl von Disziplinen zurück, unter anderem die Informatik und die Computerlinguistik.

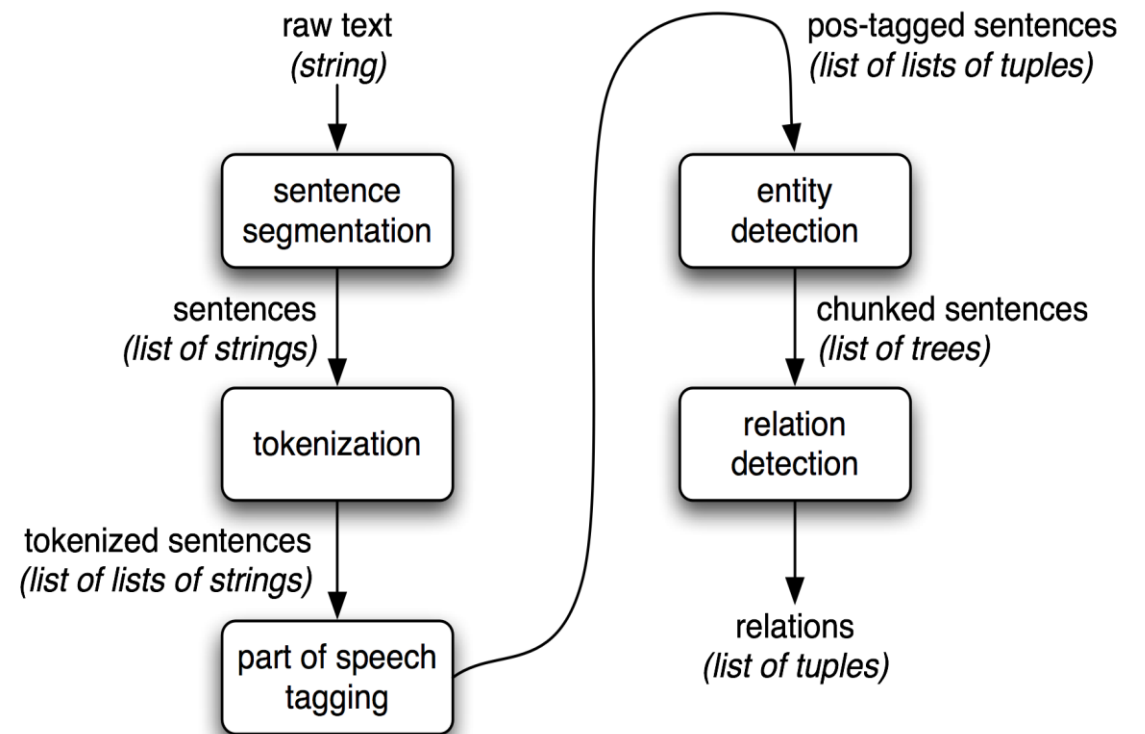
[Quelle: [SAS.com](https://sas.com)]

Natural Language Processing

Bekannte Techniken von NLP sind

- Sentence Segmentation/Tokenization
- POS Tagging
- Entity Detection
- Summarization
- Sentiment Analysis
- Keyword Extraction
- Text-to-Speech

Etc. etc. etc.



Sentence Segmentation / Tokenization

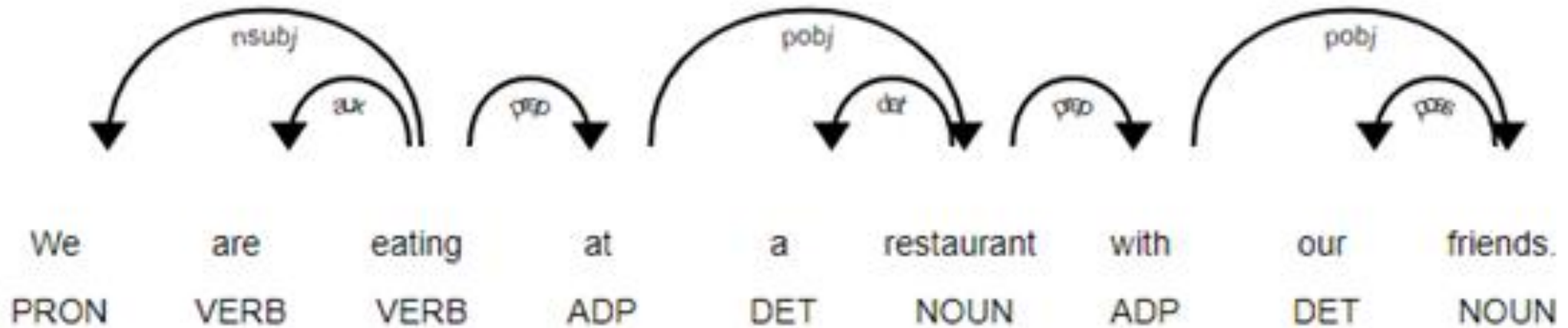
Sentence Segmentation

Hello world. This blog post is about sentence segmentation. It is not always easy to determine the end of a sentence. One difficulty of segmentation is periods that do not mark the end of a sentence. An ex. is abbreviations.



- **Hello world.**
- **This blog post is about sentence segmentation.**
- **It is not always easy to determine the end of a sentence.**
- **One difficulty of segmentation is periods that do not mark the end of a sentence.**
- **An ex. is abbreviations.**

Part of Speech Tagging (POS Tagging)



Part of Speech Tagging

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP\$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP\$	Possessive whpronoun
WRB	Whadverb

Tag	Meaning	English Examples
ADJ	adjective	new, good, high, special, big, local
ADP	adposition	on, of, at, with, by, into, under
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determiner	article the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, university

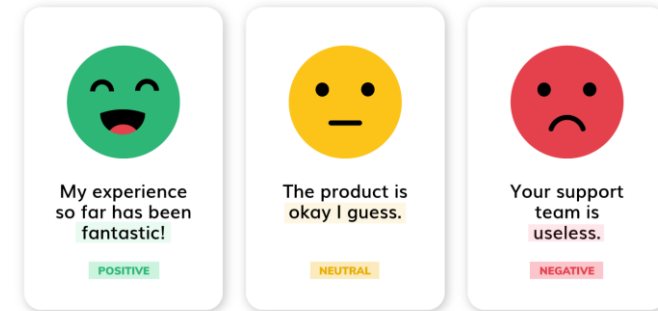
Auftrag D

- Lassen Sie zuerst runImports laufen
- Öffnen Sie TextAnalyse D.py
- Prüfen Sie ob Sie mit dem Part Of Speech (POS) Tagging einverstanden sind

```
40 from textblob_de import TextBlobDE as TextBlob #2
41 import nltk
42 nltk.download('punkt')
43
44 text = "Meine Berliner und Berlinerinnen, ich bin stolz, heute in Ihre Stadt " \
45 + "zu kommen als Gast Ihres hervorragenden Regierenden Bürgermeisters, der in allen " \
46 + "Teilen der Welt als Symbol für den Kampf- und Widerstandsgeist West-Berlins gilt. Ich " \
47 + "bin stolz, auf dieser Reise die Bundesrepublik Deutschland zusammen mit ihrem " \
48 + "hervorragenden Herrn Bundeskanzler besucht zu haben, der während so langer Jahre die Politik " \
49 + "der Bundesregierung bestimmt hat nach den Richtlinien der Demokratie, der Freiheit " \
50 + "und des Fortschritts. Ich bin stolz darauf, heute in Ihre Stadt in der " \
51 + "Gesellschaft eines amerikanischen Mitbürgers gekommen zu sein, General Clays, der hier in " \
52 + "der Zeit der schwersten Krise tätig war, durch die diese Stadt gegangen ist, und " \
53 + "der wieder nach Berlin kommen wird, wenn es notwendig werden sollte. Vor " \
54 + "zweitausend Jahren war der stolzeste Satz, den ein Mensch sagen konnte, der: Ich bin ein " \
55 + "Bürger Roms. Heute ist der stolzeste Satz, den jemand in der freien Welt sagen kann: " \
56 + "Ich bin ein Berliner."
57
58 print("\n\n\n")
59 print("Satz für Satz:")
60
61 blob = TextBlob(text)
62 sentences = blob.sentences
63 for i in range(len(sentences)):
64     print("Satz ", i, ":", sentences[i])
```


Sentiment Analysis

- Sentiment Analyse ist eine Anwendung des NLP, deren Hauptziel es ist, die emotionale Tonalität (positiv, negativ oder neutral) eines Textes zu bestimmen.
- Sie wird verwendet, um Meinungen, Einstellungen und Emotionen von Sprechern oder Schreibern zu identifizieren und zu extrahieren.



Hotel Sursee •

3.7  (86 Bewertungen) | Nr. 1 von 2 Hotels in Sursee

 Bahnhofstrasse 15, Sursee 6210 Schweiz

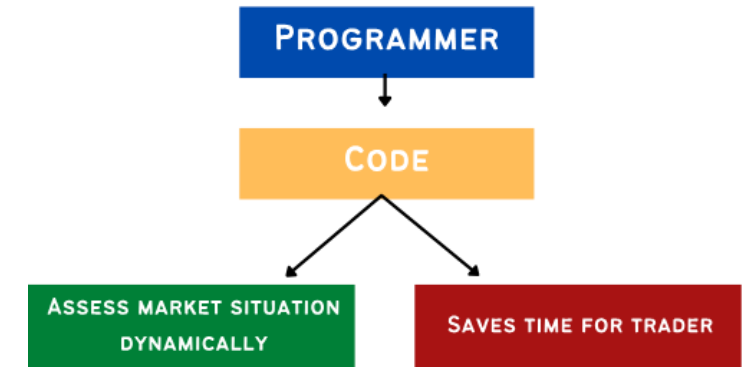
 Bewertung schreiben



Sentiment Analysis

- Sentimentanalyse wird in vielen Bereichen eingesetzt, darunter die Überwachung von Social-Media-Einträgen (Brand Monitoring), das Verständnis von Kundenrezensionen und Feedback, die Marktforschung und die Analyse von Finanznachrichten zur Vorhersage von Marktbewegungen.
- Herausforderungen: Zu den Hauptherausforderungen gehören das Verständnis von Ironie, Sarkasmus, Doppeldeutigkeiten und die korrekte Verarbeitung von negierten oder abgeschwächten Aussagen (z. B. "nicht schlecht" ist positiv, obwohl es das Wort "schlecht" enthält).

ALGORITHMIC TRADING




Sentiment Analyse

Ordner NLP / Sentiment Beispiel.py

```
from textblob_de import TextBlobDE as TextBlob
text = "Dieser Sonderwochentag ist grandios!"

blob = TextBlob(text)
sentences = blob.sentences

for i in range(len(sentences)):
    print("Satz ", i, ":", sentences[i])
    print("Sentiment: ", sentences[i].sentiment.polarity)
print("Gesamt-Sentiment :", blob.sentiment.polarity)
```



Satzweise Sentiment

Ein "geharnischter Brief"

Liebe G25s.

Ich muss euch dringendst auffordern, beim Betreten des Zimmers 010 mehr Vorsicht walten zu lassen. Es ist nun bereits zum x-ten mal passiert, dass Schüler aus eurer Klasse während meines Unterrichts (und notabene vor dem Klingeln) auf mitunter toplatschige Art und Weise die Türe geöffnet haben und ins Zimmer gelaufen sind, um dann festzustellen, dass ich am unterrichten bin.

Das ist seit Beginn dieses Semesters gefühlt jede 2. Woche passiert. Meine bisher in Grenzen gehaltene Verärgerung darüber habe ich jeweils kundgetan, leider mit begrenzter Wirkung.

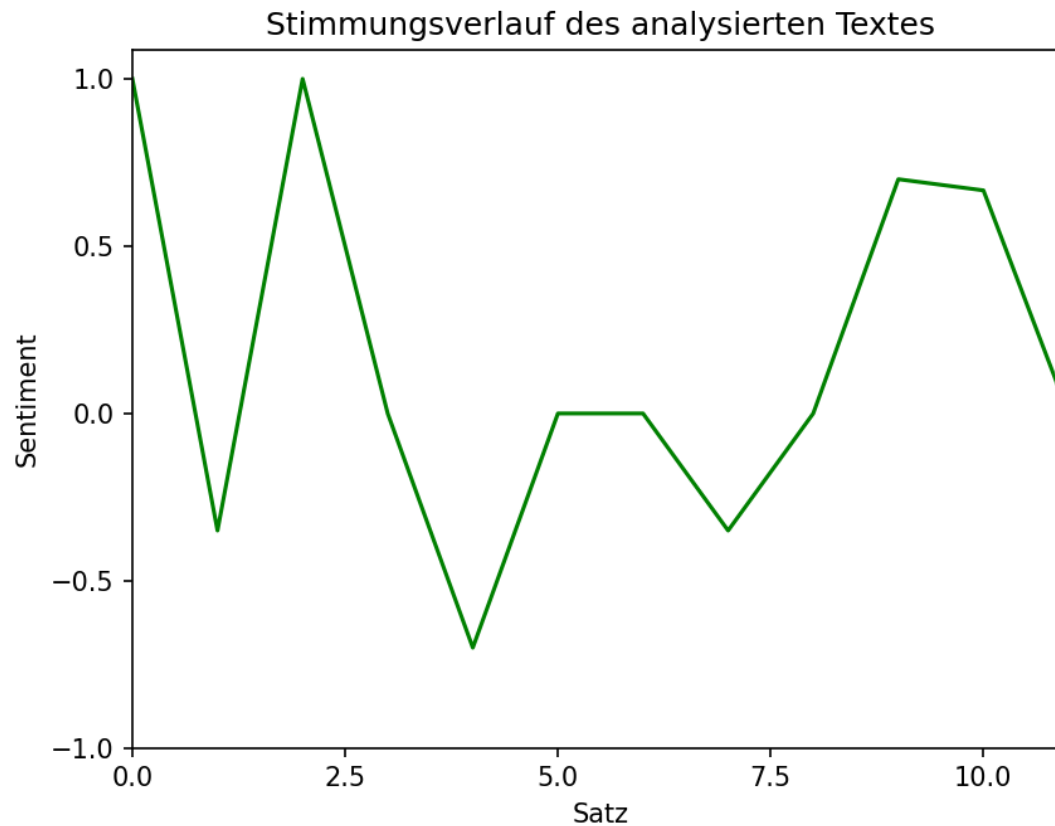
Ich muss hier protestieren, das geht so nicht. Ich will euch darauf hinweisen, dass ihr Schulzimmer nicht einfach aufreissen und reinplatzen dürft. Schon gar nicht, wenn die Schulstunden noch nicht zu Ende sind.

In dem Sinne, ich hoffe auf Einsicht. Ich schätze eure Klasse sehr, ihr habt viel Energie und seid begeisterungsfähig. Aber ihr müsst hier besser aufpassen und mehr Rücksicht walten lassen.

Aufgabe "geharnischter Brief"

- a) Wende die Sentiment Analyse auf den Brief an
- b) Zeige den Verlauf der Stimmung über den Brief hinweg mit Matplotlib auf

Aufgabe "Geharnischter Brief"



How To Sandwich Your Feedback

Step 1: **Praise Employees**



Step 2: **Negative Feedback**



Step 3: **Praise Again**



Speech to Text



Speech to text ist mittlerweile eine sehr breit eingesetzte Technologie. Die folgenden drei Schritte stehen dabei im Zentrum:

1. **Audio signal-Erfassung:** Das Mikrofon nimmt die akustischen Schwingungen der Stimme auf und wandelt sie in ein digitales Signal um.
2. **Akustische Modellierung:** Das System gleicht die extrahierten Audio-Features mit einem Akustischen Modell ab. Dieses Modell enthält Wahrscheinlichkeiten dafür, welche Klänge (Phoneme) des menschlichen Sprechens zu den gehörten Signalen passen könnten.
3. **Spracherkennung und Transkription:** Die wahrscheinlichsten Phonem-Sequenzen werden anschliessend durch ein Sprachmodell gefiltert. Das Sprachmodell bewertet, welche Wortkombinationen in einer bestimmten Sprache (z.B. Deutsch) am sinnvollsten und grammatikalisch korrekt sind, um die endgültige Textausgabe zu erzeugen.

Speech to Text

- Öffne die Datei Speech-To-Text.py und lass sie laufen.
- Du musst etwas in dein Mikrofon sprechen, und das sollte dann erkannt werden
- Nach 20s bricht das Tool ab und schreibt auf was es gehört hat

Übung: Live Sentiment Analyse

Schreibe ein kleines Programm, welches deine Sprache verarbeitet und dir einen Sentiment-Wert berechnet zwischen -1 und 1

**Voice Call
Sentiment
Analysis**



Übung: Live Sentiment Analyse – zweiter Teil

Schreibe ein kleines Programm, welches die "Stimmung im Raum" analysiert. Es soll alle Sprache aufschnappen, bewerten, und dann den Mittelwert aller letzten 10 Sätze berechnen und diesen anzeigen.

Wettbewerb



Mini-Projekt

Wählt eines oder mehrere der heute angeschauten Werkzeuge aus und setzt euch ein Ziel – Beispiele

- Ein "automatischer Krypto Trading" Bot
- Analyse von Songlyrics mittels NLP Methoden
- Eine Speech-to-Text Anwendung – z.B. Lieder abspielen auf Kommando
- Sentiment - Anwendung – z.B. mit News, Literatur, oder z.B. einen Chatverlauf auf Stimmung untersuchen

Wettbewerb

- Gruppen oder alleine – ihr wählt selbst
- Jede/r hat 1 Stimme
- Stimmen für sich selbst ist nicht erlaubt
- Der Gewinner bekommt einen kleinen Preis

