

Datenanalyse und Einführung in Maschinelles Lernen WS 2025/26

Einführungsveranstaltung



Dozentin: Grit Behrens
mailto: grit.behrens@hsbi.de

Studiengang Informatik Fachbereich Campus Minden

Einführungsveranstaltung

Inhalt für heute:

- 1. Motivation für Datenanalyse: Maschinelles Lernen & Nachhaltigkeit**
- 2. Organisatorisches (Termine, Bewertung)**
- 3. Lehrinhalte der Vorlesungen und Praktika**
- 4. Einstieg**

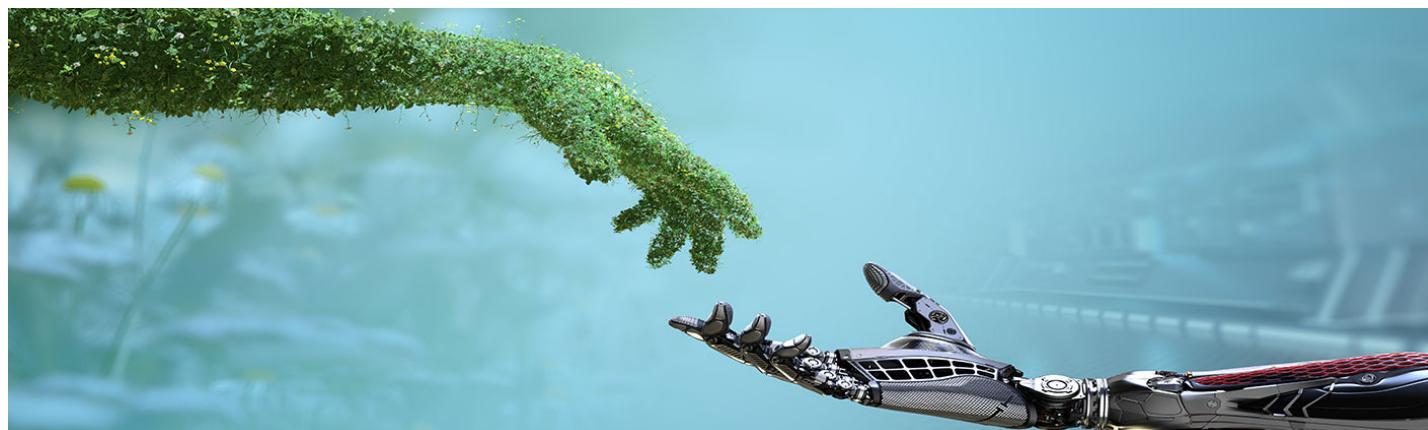
Einführungsveranstaltung

Inhalt für heute:

1. Motivation für Datenanalyse: ML & Nachhaltigkeit
2. Organisatorisches (Termine, Bewertung)
3. Lehrinhalte der Vorlesungen und Praktika
4. Einstieg

Motivation: Maschinelles Lernen und Nachhaltigkeit

Fokus: KI und Umwelt



Motivation: Maschinelles Lernen und Nachhaltigkeit

- Klimawandel regional und weltweit
- Nachteile und Vorteile von ML für den Klimawandel
- Beispielprojekte



Klimawandel regional

- 24.12.2023 Werre-Hochwasser-Rückhaltebecken in Löhne wird geflutet
- ▶ Regelmäßige Hochwasser in der Werre-Region bereiten Gefahrenlage
- ▶ 26.12. 2023 Überschwemmungen in flussnahen Gebieten

eitag, 05.04.2024

WESTFALEN-BLATT

OWL ÜBERREGIONAL ARMINIA SC PADERBORN 07 FOTOS

Dezember ein Rückhaltebecken in Löhne geflutet worden. Auch im benachbarten Bad Oeynhausen führt da Hochwasser vielerorts zu Problemen.

 Von Malte Samtenschnieder

sonntag, 24.12.2023, 15:40 Uhr aktualisiert: 25.12.2023, 18:27 Uhr



Werre zu senken, ist an Heiligabend (24. Dezember) das Hochwasser-Rückhaltebecken in Löhne geflutet worden. Foto:

WESTFALEN-BLATT

OWL ÜBERREGIONAL ARMINIA SC PADERBORN 07 FOTOS

hoher Werre-Pegel bereitet Sorgen in Bad Oeynhausen.

Bad Oeynhausen - Als „angespannt, aber vergleichsweise noch entspannt“, hat Stadtsprecher Volker Müller-Ulrich am zweiten Weihnachtstag (26. Dezember) die Hochwasserlage in Bad Oeynhausen bezeichnet. Besonders der hohe Werre-Pegel sei besorgniserregend.

 Von Malte Samtenschnieder

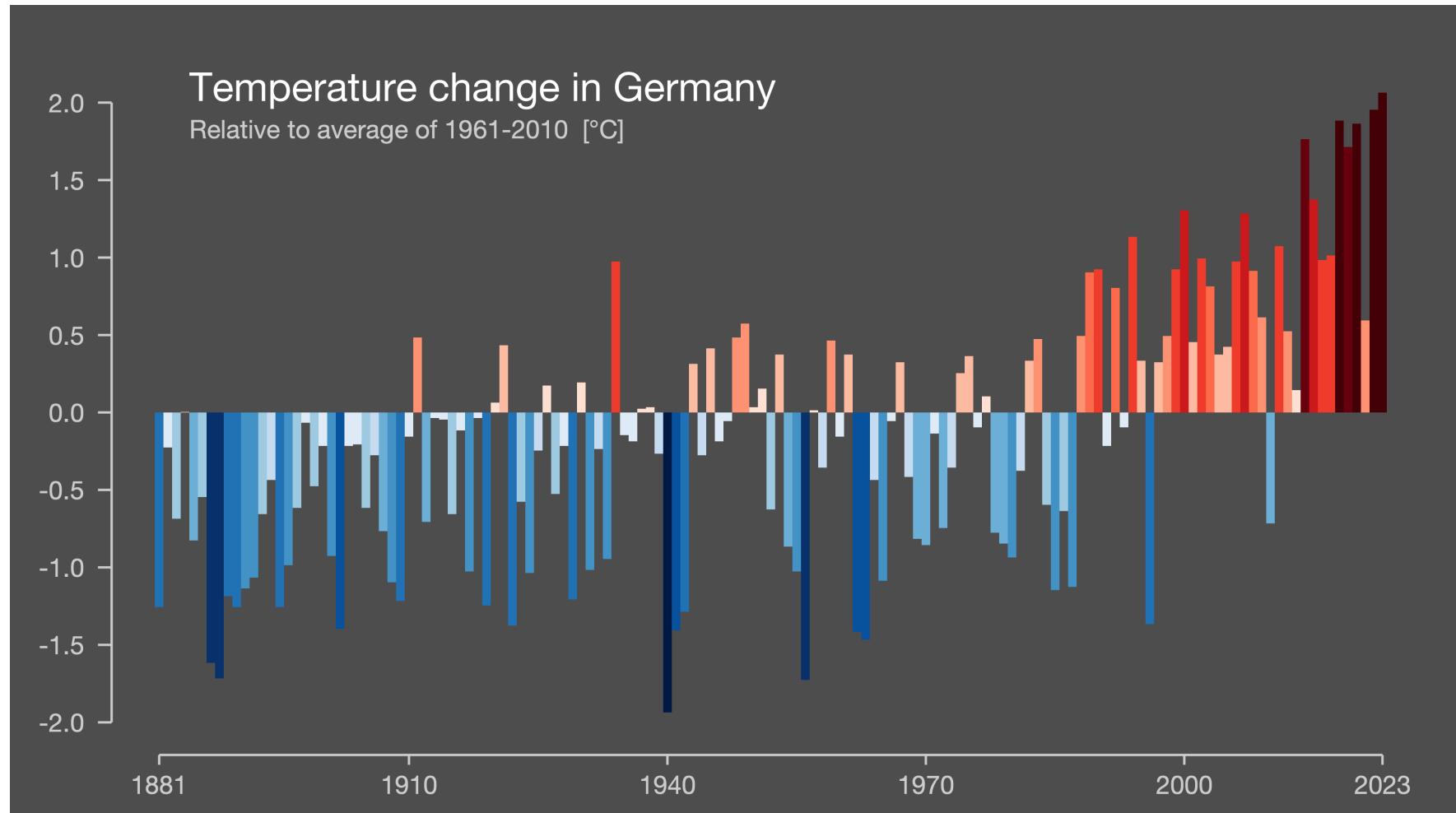
sonntag, 26.12.2023, 17:16 Uhr aktualisiert: 26.12.2023, 20:25 Uhr



ist die Weser über die Ufer getreten. Der Amanda-Anleger (Foto), die Sportplätze von Rot-Weiß Rehme u. Weserhütte sind unter anderem betroffen. Foto: Malte Samtenschnieder



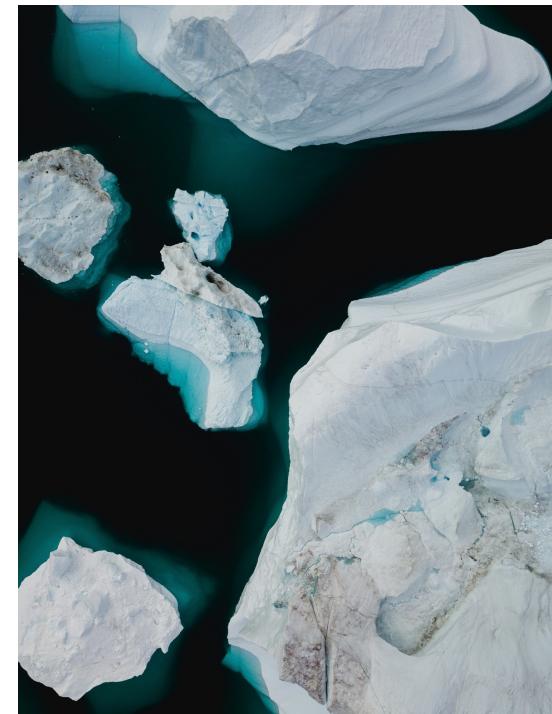
Klimawandel regional



<https://showyourstripes.info/c/europe/germany/all>

Klimawandel weltweit

- Temperaturanstieg und Eisschmelze bedroht alle küstennahen Städte und Gebiete weltweit, ganze Inselnationen
- Extremwetterlagen verstärken sich, Dürre, Hitze, Überschwemmung, Hurrikane
- Lebensmittelknappheit verursachen Migration weltweit
- Tier- und Pflanzensterben – bereits über eine Million Arten ausgestorben
- ...



Klimawandel weltweit



- Hirnkoralle 2014



- ▶ Hirnkoralle 2024

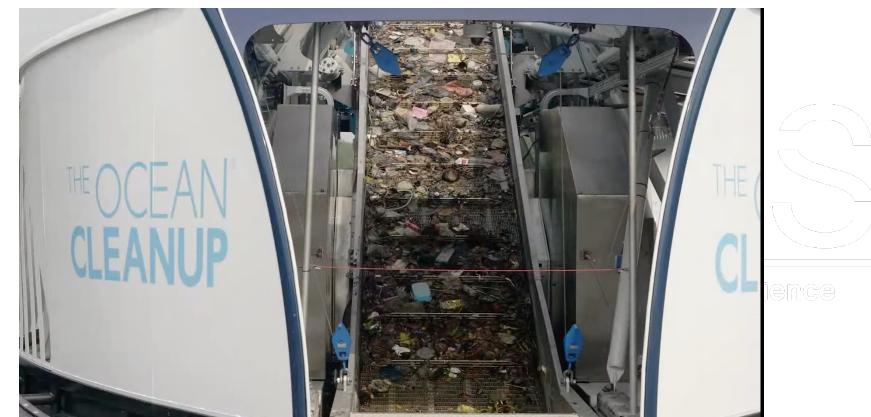
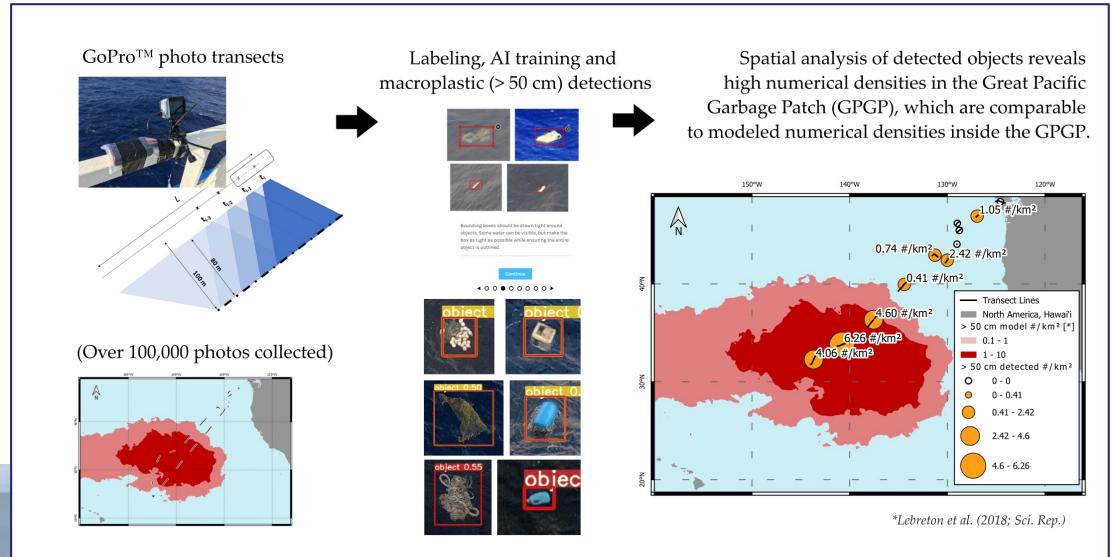
Vorteile von ML für die Umwelt

Vielfältige Beispiele

- **Effizienzsteigerung durch optimierte Prozesse**, dadurch gesenkter Energie- und Ressourcenverbrauch in Produktion, Landwirtschaft und Verkehr
- **Entscheidungsunterstützung** durch Analyse großer Datenmengen und Handlungsempfehlungen für umweltfreundliche Entscheidungen
- **Kreislaufwirtschaft** – verbessertes Recycling und Abfallsortierung in automatisierten Prozessen
- **Umweltmonitoring** – Sattelitenbilder auswerten und Sensordaten auswerten, um z.B. Waldschäden und Plastikmüll zu identifizieren
- **Energiewende** – Steuerung erneuerbarer Energien und Energienetze - > Energieeffizienz erhöhen
- **Präzisionslandwirtschaft** : Optimierung von Düngemittel- und Wasserverbrauch für höhere Erträge bei geringerem Ressourceneinsatz
- ...

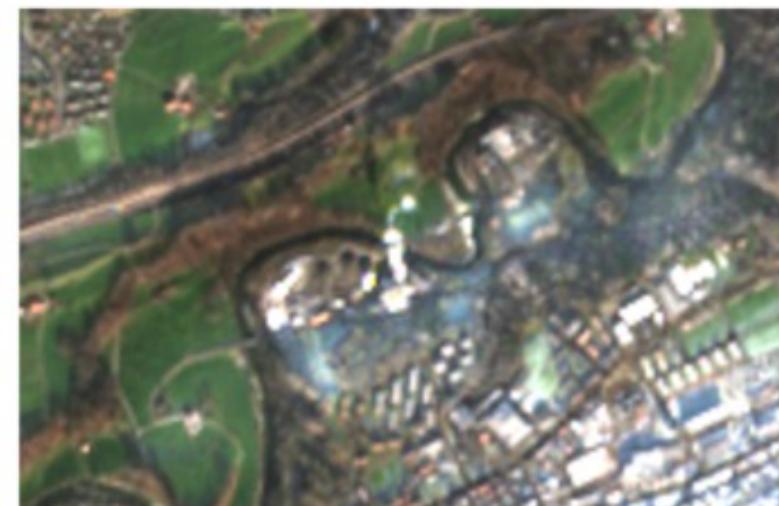
Ocean Cleanup

- Niederländische Umwelt-Organisation verwendet u.a. ML-Methoden auf Satellitenbildern, um Plastic-Müll in den Ozeanen aufzuspüren (River-AI-Model, Imaging System, Remote Sensing - Hyperspektralsensoren)



Erkennung von Rauchwolken in Satellitenbildern

- Charakterisierung von Schadstoffemissionen aus Satellitenbildern mit Deep Learning
- Ermöglicht bessere Überwachung der Schadstoffgrenzen
- Ein Projekt von Copernicus Sentinel 2021



Institute for Data Science
Solutions of HSBI

Risiken von ML für die Umwelt

- **Hoher Energiebedarf –** Training und Nutzung der Modelle energieintensiv, CO² – Fußabdruck zu hoch
- **Ressorcenverbrauch –** Metalle und seltene Erden werden in großen Mengen benötigt,
- **Rebound-Effekte –** Mehrkonsum macht Effizienzgewinne zunichtet, ML fördert auch Reklame von Konsumgütern
- **Intransparenz –** fehlende Nachvollziehbarkeit von ML-Entscheidungen erschwert Kontrolle auf Nachhaltigkeit



*„Die gesamte Infrastruktur der Rechenzentren und die Netzwerke zur Datenübermittlung sind weltweit für **zwei bis vier Prozent der weltweiten CO²-Emissionen** verantwortlich, das ist soviel wie die Emissionen der gesamten Luftfahrtindustrie“, Benedetta Brevini – Prof. für politische Ökonomie in „**Is AI good for the Planet?**“, ISBN-13 978-1509547951“*

KI-Leuchttürme für Umwelt, Klima, Natur und Ressourcen



Bundesministerium
für Umwelt, Naturschutz, nukleare Sicherheit
und Verbraucherschutz

Forschungs-Projekt EKI (2023-2025)

Energieeffiziente Künstliche Intelligenz im Rechenzentrum durch Approximation von tiefen neuronalen Netzen für Field-Programmable Gate-Arrays (FPGA's)

Ausgangssituation: Hoher Energiebedarf und einhergehender CO₂ – Ausstoß in Rechenzentren durch DNN

Idee:

- Aufbau von FPGA-Systemen mit hoher Energieeffizienz anstelle von Graphikprozessoren (GPU's) oder Zentralprozessoren (CPU's)
- Einsatz von Approximationsverfahren, statt Berechnungen
- Experimentelle Evaluation mit Test-DNN's

Ziel: Energieeinsparungen bei der Nutzung von großen DNN-Modellen auf Hochleistungs-FPGA's in Rechenzentren



EDI
Institute for Data Science
Solutions of HSBI

Uni Paderborn, FH Südwestfalen, HS Hamm-Lippstadt

KI-Leuchttürme für Umwelt, Klima, Natur und Ressourcen



Bundesministerium
für Umwelt, Naturschutz, nukleare Sicherheit
und Verbraucherschutz

Forschungs-Projekt KIRA(2023-2025)

KI-Referenzmodell für Energie- und Ressourceneffizienz
und dessen industrielle Anwendung

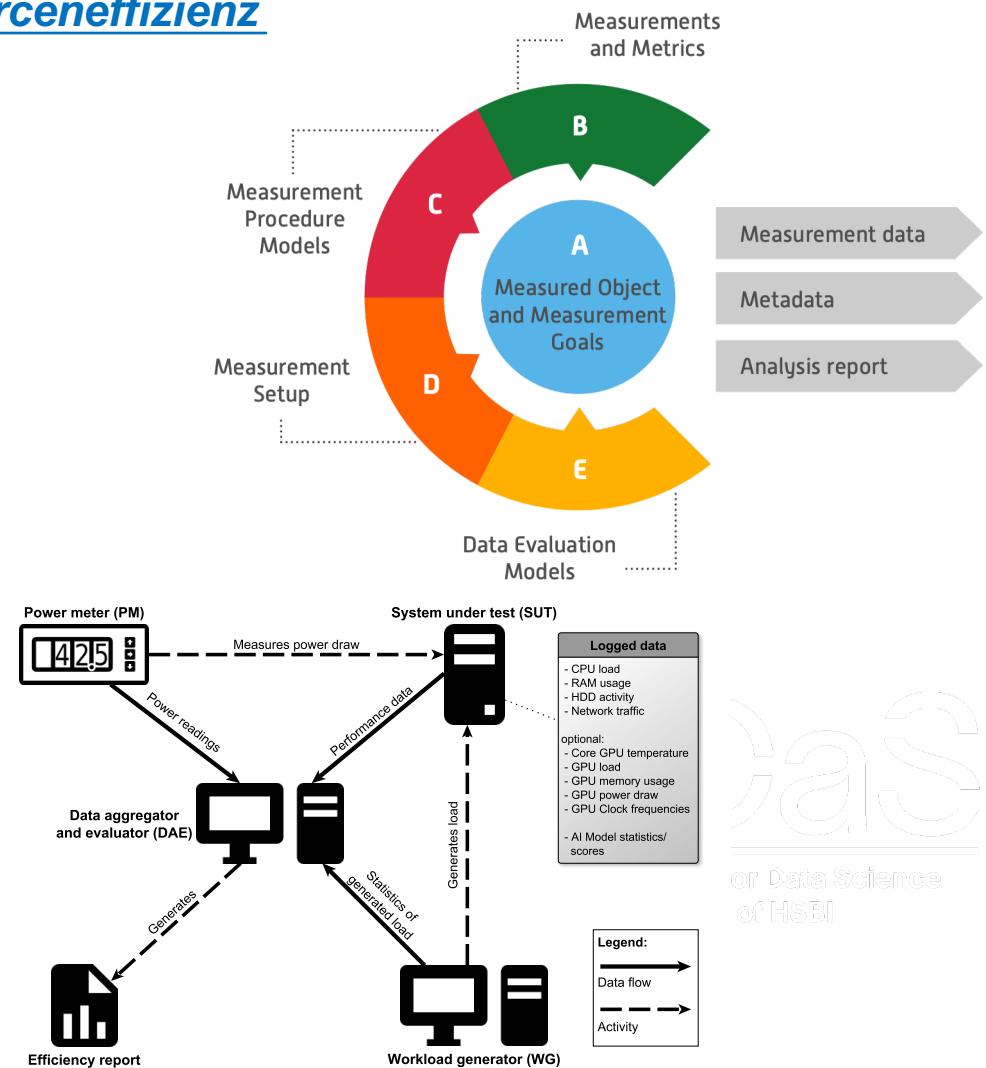
Ausgangssituation: Hoher Energie- und Ressourcenbedarf von KI-Anwendungen

Idee:

- Entwickeln und Prüfen von **Kriterien und Metriken** für Ressourcen- und Energiebedarf von KI-Systemen entlang des Lebenszyklus
- KI-Referenzmodell strukturiert **Zusammenhänge und Abhängigkeiten**, macht sie **transparent** und optimierbar
- Evaluation anhand von **Fallstudien**

Ziel: Entwicklung von ressourceneffizienter KI ermöglichen

Hochschule Trier, Umwelt-Campus Birkenfeld,
Öko-Institut, BITO Campus GmbH



Fazit : ML für die Umwelt braucht

- Gute vorbereitende Datenanalyse
- Effiziente Algorithmen mit wenig Energiebedarf
- Datenreduktion auf aussagekräftige Features
- Weniger Datentransport und ggf. Servernutzung
- Energieeffiziente Server nutzen
- Ggf. mehr Edge-Computing
- Erklärbare Algorithmen für mehr Transparenz



➤ Nutzt dieses ML-Projekt der Umwelt oder einem der 17 SDG's der UNO?

Einführungsveranstaltung

Inhalt für heute:

1. Motivation für Datenanalyse: ML & Nachhaltigkeit
2. **Organisatorisches (Termine, Bewertung)**
3. Lehrinhalte der Vorlesungen und Praktika
4. Einstieg

WBA-Organisatorisches I

- **Vorlesungen alle in Präsenz (ca.12 Termine + Klausurvorbereitung)**
Mi 14:00-15:30 Uhr – B70
- **Praktikatermine alle in Präsenz (ca. 10 Blätter)**
Mi 11:30 – 13:00 Uhr C2 Gr3 (immer eine Woche nach der VL)
Do 9:45 - 11:15 Uhr C2 Gr1 (mit M. sc. Sarah Flohr)
Do 11:30 – 13:00 Uhr C2 Gr.2
Do 14:00 – 15:30 Uhr C2 Gr.4
- **erste Praktika nach der 2. Vorlesung**
Aufgabenblätter **zu Hause mit Python vorbereiten** und im Praktikum die Lösungen vorführen. Es gibt für jedes Blatt Punkte.
- **Aufgabenblätter und Vorlesungsfolien in ILIAS;**
Beitritt mit Kurspasswort „Datenanalyse202526“
- Konsultationen nach Vereinbarung (z.B. per Mail)

WBA-Organisatorisches II

Besondere Termine mit Unterrichtsausfall:

- HTG-Kongress (1 Tag, Flohr + Behrens), Münster
Do 6.11.2024 keine Praktika in den Gruppen 1, 2 und 4
- Konferenzwoche Asia PVSEC, Bangkok
Mi 12.11. 2024 keine Vorlesung und kein Praktikum
Do 13.11.2024 keine Praktika , Gr 1 mit Sarah Flohr kann den Termin ggf. nutzen

Organisatorisches: Bewertung

Leistungsbewertung des Moduls zu 5 CP's in 2 extra Teilnoten:

1. Teilnote (Praktikum 2,5 Credit Points):

- Aufgaben werden zu jedem Blatt bewertet (je nach Punkteangaben auf den Blättern)
- Termine der Abgaben müssen unbedingt eingehalten werden.
Terminverzug um eine Woche - die Hälfte der Punkte Abzug. Zwei Wochen später Null Punkte.
- **Abgabe zu zweit**, jeder muss alles erklären können, unterschiedliche Punktevergabe im Team ist möglich.
- Gesamtnote wird zum Schluss mit einheitlicher Notenskala berechnet, ab 50% bestanden.
- 75% der Anwesenheit obligatorisch-> ca. 3 mal darf man max. fehlen (Punkte fehlen dann aber auch)

2. Teilnote (Klausur - 2,5 Credit Points):

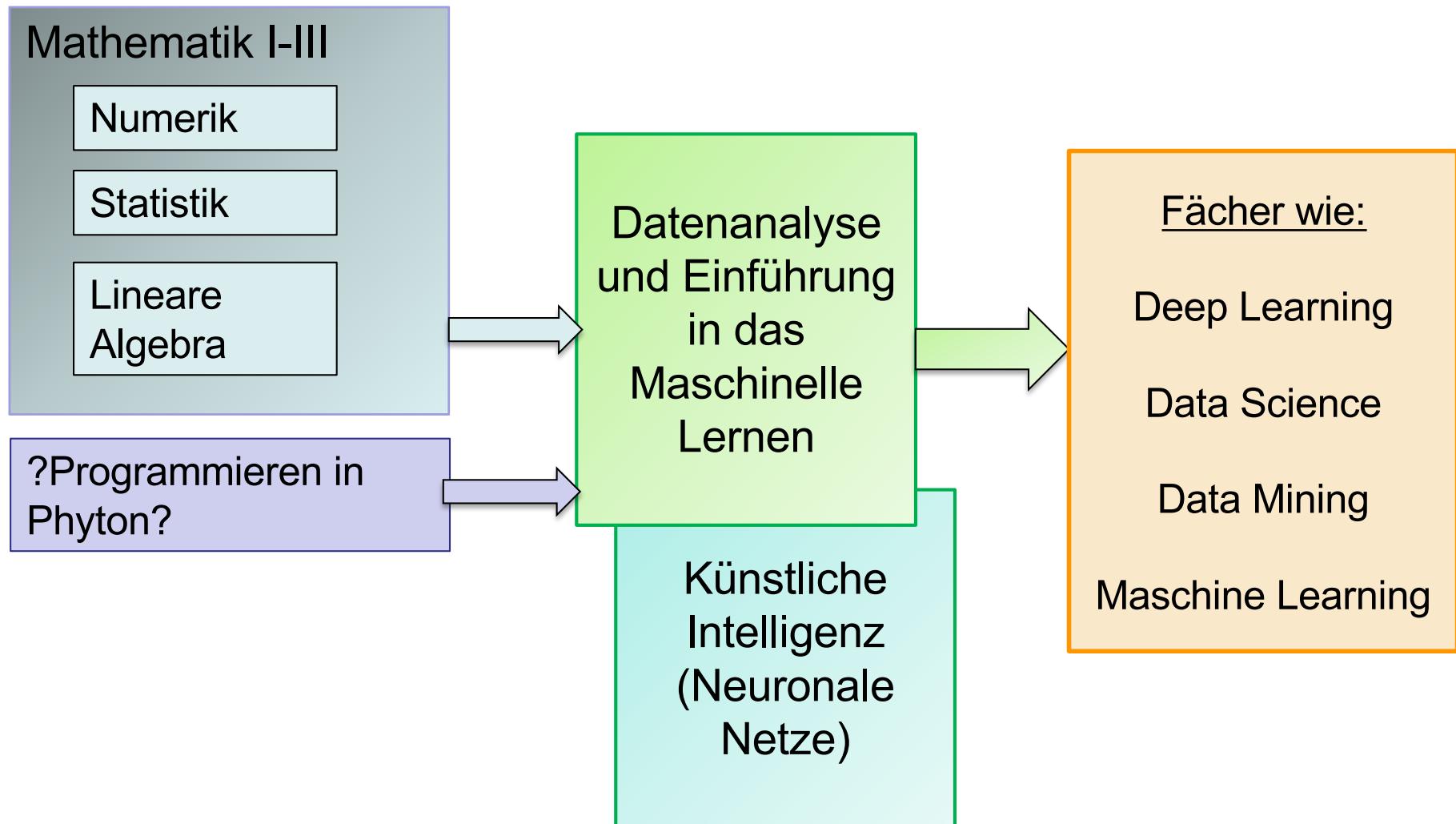
- Schreiben der Klausur; max. 3 Versuche
- Keine Hilfsmittel
- Stoff aus den Vorlesungen mit Fragen und kleineren Aufgaben inhaltlich ähnlich wie in den Praktika oder den VL-Beispielen

Einführungsveranstaltung

Inhalt für heute:

1. Motivation für Datenanalyse: ML & Nachhaltigkeit
2. Organisatorisches (Termine, Bewertung)
3. **Lehrinhalte der Vorlesungen und Praktika**
4. Einstieg

Ökosystem der Teilgebiete/Module



Lehrinhalte der Veranstaltung

- 1. Einstieg zu Datenanalyse und Einführung in Maschinelles Lernen**
2. Datenanalyse mit Python
3. Zeitreihen
4. Maschinelles Lernen: Neuronale Netze
5. Lineare Regression
6. Logistische Regression und Klassifikation

Einführungsveranstaltung

Inhalt für heute:

1. Motivation für Datenanalyse: KI & Nachhaltigkeit
2. Organisatorisches (Termine, Bewertung)
3. Lehrinhalte der Vorlesungen und Praktika
- 4. Einstieg**

Was sind Daten?

Daten sind Informationen.

...Daten kann man nie genug haben.

...Daten werden vermarktet.

...Daten werden als die **Quelle der Wahrheit** angesehen,

Stimmt das immer so? Was ist tatsächlich wichtig in Bezug auf Daten?

Wichtig ist die **Analyse** der Daten!

Wichtig ist der **Erzeugungsprozess!** Wann sind sie **wie** entstanden.

Wichtig ist zu wissen: **wo** die Daten **her** kommen!

Def: Daten sind Momentaufnahmen der Realität. Sie alleine beinhalten selbst keine kontinuierliche Realität.

- **Die Kontexte fehlen häufig.**
- **Sie können verzerrt sein.**
- **Es treten Lücken auf.**
- **Es fehlen häufig relevante Variablen/Features.**

→ Es gibt bei Daten immer einen eingeschränkten Betrachtungsbereich.

Für die Analyse benötigen wir eine definierte Zielstellung.

→ Daten liefern Hinweise, nicht unbedingt die Wahrheit, schon gar nicht die Grundwahrheit (**ground truth**).

Die Grundwahrheit

Def: Ground truth

- **korrekte fehlerfreie** Daten oder Informationen, die bei **Beobachtungen oder Messungen in der realen Welt** gesammelt wurden
- werden oft als **Benchmark** genutzt, um die Genauigkeit des Modells zu testen.
- beim maschinellen Lernen zum Lernen als **Trainings- und Testdatensatz** genutzt.

Warum ist es häufig so schwierig, die Grundwahrheit zu erhalten?

- Sensoren! -> Verfügbarkeit und Kosten bei der Datenaufnahme
- Manuelles Daten sammeln kann sehr aufwendig sein ; fehlerträchtig
- Veränderlichkeit der Ereignisse aufgrund von Umwelteinflüssen
- Voreingenommenheit und Fehler bei der Erhebung der Daten (repräsentative Daten?)
- Daten labeln (kennzeichnen) kann Herausforderung sein
- Ethische und Datenschutzbedenken

Arten von tabellarischen Daten

Unterscheidung nach den Skalen

1. **Nominal** skalierte Daten: kategoriall, keine feste Reihenfolge, z.B. Geschlecht , es gilt lediglich die **Gleichheit** oder die **Ungleichheit**,
2. **Ordinal** skalierte Daten: Werte können geordnet werden, aber ohne Abstandsmaß, definieren eine **Ordnungsrelation** „<“ oder „>“, z.B. „Wochentage“
3. **Intervall** skalierte Daten: Werte können geordnet und Abstände angegeben werden, Beispiele sind Skalen mit Nullpunkt wie die „Temperatur mit absolutem Nullpunkt -273 K“, „Jahresangaben“)
4. **Proportional** skalierte Daten: Definieren eine klare Skala, so dass das Verhältnis zwischen zwei Werten eine sinnvolle Größe ist, (z.B. Alter, Einkommen)

→ **Ab welchem Skalentyp macht es Sinn einen Mittelwert zu berechnen?**
→ **Wie kann ich Variablen der verschiedenen Skalentypen für die Datenanalyse nutzen?**

Arten von großen Datenmengen ?

Welche Arten von großen Datenmengen kennen Sie?

- ..
- ..
- ..
- ..
- ..

Arten von großen Datenmengen ?

Welche Arten von großen Datenmengen kennen Sie (SS 2024)?

1. **Social Media (z.B. persönliche Daten und Verlaufsdaten für Werbeanzeigen)**
2. **Videokameras aus der Überwachung oder aus ZOOM zur Gesichtserkennung**
3. **Anfragen auf Suchmaschinen -> Antworten auswerten für Optimierung**
4. **Umgebungsdaten beim autonomen Fahren**
5. **Satellitenbilder, Nachrichten aus der Medien-Kommunikation -> Terrorprognose**
6. **Daten für den Katastrophenschutz zur Prognose von Umweltkatastrophen**
7. **Daten aus der Produktion zur Qualitätskontrolle**
8. **Daten aus der Produktion zur Wartung der Produktionsstraßen**
9. **Energiedaten zur Optimierung**
10. **Medizinische Daten zur Vorhersage von Grippewellen**

Begriff Maschinelles Lernen

Wir betrachten das „**Maschinelle Lernen**“ als Teilgebiet der **KI**, das in seiner Anwendung bei strukturierten Daten häufig in den **DataMining – Prozess eingebettet** ist.

Der englische Begriff : „**Machine Learning**“ wird ebenso häufig verwendet.

Wir beschäftigen uns mit **strukturierten Datensätzen**, die in Tabellenform darstellbar sind und in **Merkmalsvektoren** umgewandelt werden können.

Unüberwachtes Lernen

Agent **lernt Muster in der Eingabe**,
die Daten sind **nicht gelabelt**,
die Ausgaben sind nicht erkannt. (z.B. Cluster von ähnlichen nützlichen Mustern)

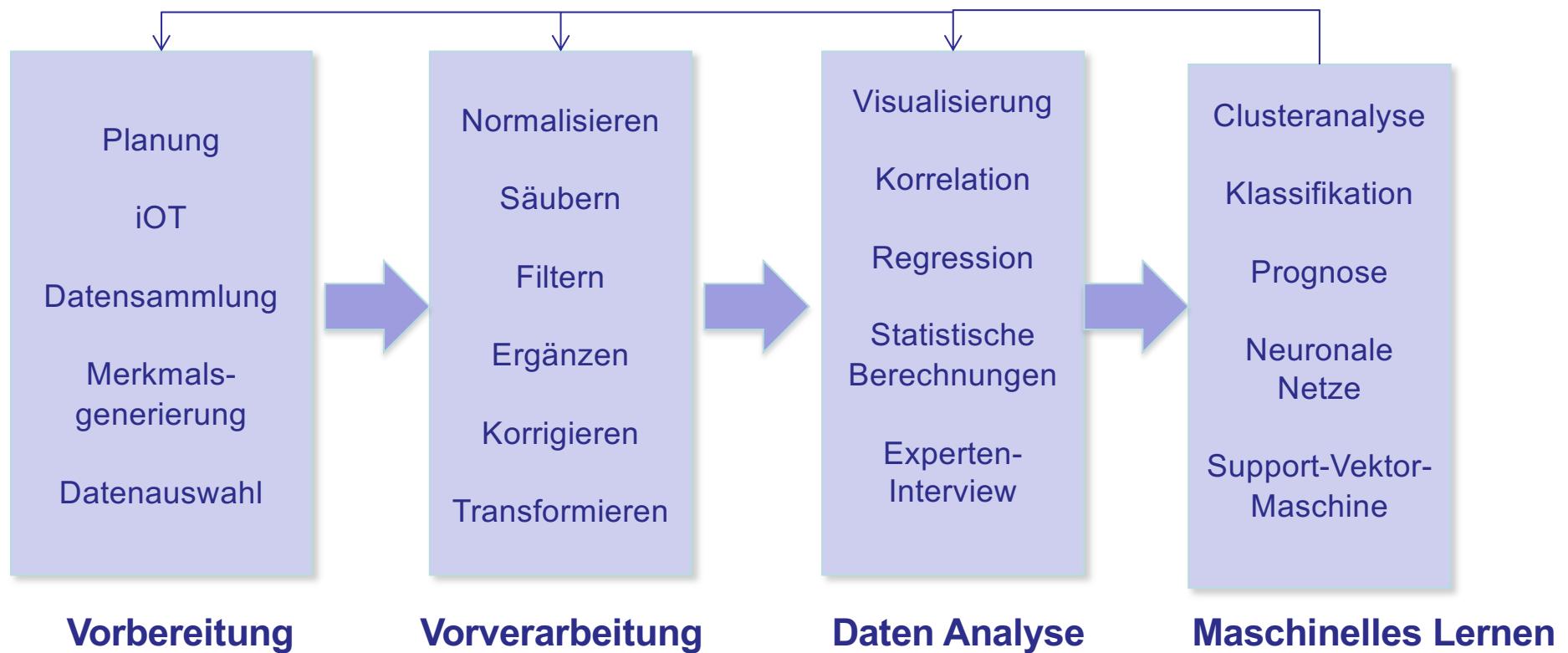
Überwachtes Lernen

Agent **lernt Funktion durch die Abbildung der Paare von Eingabedaten auf die Ausgabedaten**.

Die **Ausgabedaten müssen gelabelt sein**. (z.B. Objekterkennung)

Maschinelles Lernen arbeitet häufig mit **sehr großen Datenmengen**.

„Datenanalyse“ und „Maschinelles Lernen“ sind eingebettet in den Data-Mining-Prozess:



Begriff: Datenanalyse

- **Definition:** Die Datenanalyse arbeitet mit **statistischen Methoden**, gewinnt **zusammenfassende Informationen** (Kenngrößen) aus **numerischen** Daten und bereitet diese **grafisch** auf. Sie ist in den **Data-Mining-Prozess** eingebettet.
- **Univariate Kenngrößen** beziehen sich auf eine einzelne Variable, nicht auf Abhängigkeiten der Variablen untereinander z.B.:
 - arithmetisches Mittel
 - Gewichtetes arithmetisches Mittel
 - Median
 - Modus
- **Streumaße, (auch genannt: Spread, Dispersion)** charakterisieren, inwiefern Daten um ein Parameter herum streuen, also davon mehr oder weniger abweichen, z.B.:
 - Varianz
 - Standardabweichung
- **Multivariate Kenngrößen** beschreiben Abhängigkeiten der Variablen untereinander
 - Korrelationen

Arithmetisches Mittel, Mittelwert

- **Definition:** Der **Mittelwert** ist der Durchschnitt einer Reihe von Werten. Man berechnet die Summe der Werte und dividiert sie durch die Anzahl der Werte.
 - Der Mittelwert zeigt, wo der **Schwerpunkt einer Wertegruppe** liegt.

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} = \sum \frac{x_i}{n}$$

Σ – Summenzeichen

n – Anzahl der Elemente

#Mittelwert, Beispiel

#Anzahl der Haustiere, die jede Person einer Gruppe besitzt

```
sample = [1, 3, 2, 5, 7, 0, 2, 3]
mean = sum(sample) / len(sample)
print(mean) # Ausgabe: 2.87
```

Gewichteter Mittelwert I

- **Definition:** Beim **gewichteten Mittelwert** kann jedem Wert ein Gewicht zugeordnet werden. Jeder Wert wird dann mit dem jeweiligen Gewicht multipliziert und durch die Summe der Gewichte dividiert.
- Der gewichtete Mittelwert kann **nach Bedeutung der Werte manipuliert** werden.

$$\text{gewichteter Mittelwert} = \frac{(x_1 \cdot w_1) + (x_2 \cdot w_2) + (x_3 \cdot w_3) + \dots (x_n \cdot w_n)}{w_1 + w_2 + w_3 + \dots + w_n}$$

w_i – Gewichte der Elemente

```
# gewichteter Mittelwert
# drei Teilnoten mit einem Gewicht von je 20% und
# eine schriftliche Abschlussprüfung mit einem Gewicht von 40%
sample = [1.3, 1.0, 2.0, 3.0]
weights = [.20, .20, .20, .40]
weighted_mean = sum(s * w for s,w in zip(sample, weights)) /
sum(weights)
print(weighted_mean) # Ausgabe:2.06
```

Gewichteter Mittelwert II

```
# gewichteter Mittelwert  
# drei Teilnoten mit einem Gewicht von je 20% und  
# eine schriftliche Abschlussprüfung mit einem Gewicht von 40%  
sample = [1.3, 1.0, 2.0, 3.0]  
weights = [1.0, 1.0, 1.0, 2.0 ]  
weighted_mean = sum(s * w for s,w in zip(sample,  
weights)) / sum(weights)  
print(weighted_mean) # Ausgabe:2.06
```

Wir gewichten jede Prüfung mit »1«, die Abschlussprüfung aber mit »2«, sodass sie das doppelte Gewicht einer einfachen Teilnote erhält. Die Antwort bleibt die gleiche mit 2.06. Die Gewichte sind proportionalisiert (stehen im gleichen Verhältnis).

```
#Verwendung der zip-Funktion zum Paaren von Listen in Python:  
gezippte_listen = zip(sample, weights)  
  
#Umwandeln in eine Liste  
ergebnis_liste=list(gezippte_listen)  
  
print(ergebnis_liste) # Ausgabe: [(1.3, 1.0), (1.0, 1.0), (2.0, 1.0), (3.0, 2.0)]
```

Median

- **Definition:** Der **Median** ist der mittlere Wert in einer Reihe von **geordneten** Werten. Bei einer **ungeraden Anzahl** der sortierten Werte ist der Median genau in der **Mitte**. Bei einer **geraden Anzahl** von Werten wird der **Mittelwert der beiden Werte gebildet, die der Anzahl-Mitte am nächsten liegen.**

```
#Median, Beispiel
#Anzahl der Haustiere, die jede Person einer Gruppe besitzt
sample=[0,1,5,9,7,11,10]
def median(values):
    ordered = sorted(values)
    print(ordered)
    n = len(ordered)
    mid = int(n / 2) - 1 if n% 2 == 0 else int(n/2) # „linke“ Mittelposition
    if n% 2 == 0:
        return (ordered[mid] + ordered[mid+1]) / 2.0
    else:      # gerade Anzahl -> Mittelwert von linker und rechter Mittelposition
        return ordered[mid] # ungerade Anzahl -> mittlere Zahl
print(median(sample)) #output [0, 1, 5, 7, 9, 10, 14] # geordnete Liste
                                         # Median
```

Modus

- **Definition:** Der **Modus** ist der **am häufigsten vorkommende Wert** in einer Stichprobe.
Kommt kein Wert mehr als zweimal vor, gibt es keinen Modus. Treten die **zwei Maximal-Werte mit gleicher Häufigkeit** auf, wird der Datensatz als **bimodal** bezeichnet.

```
#Modus, Beispiel
```

```
# Anzahl der Haustiere, die jede Person besitzt
```

```
from collections import defaultdict      # Unterklasse von dictionary, nutzt Defaultwerte
```

```
sample = [1, 3, 2, 4, 7, 0, 1, 3]
```

```
def mode(values):
```

```
    counts = defaultdict(lambda: 0)
```

#Überschreiben der missing keys

```
    for s in values: #Hochzählen der Anzahl für Werte aus sample
```

```
        counts[s] += 1
```

```
        print(counts)
```

```
{1: 1}  
{1: 1, 3: 1}  
{1: 1, 3: 1, 2: 1}  
{1: 1, 3: 1, 2: 1, 4: 1}  
{1: 1, 3: 1, 2: 1, 4: 1, 7: 1}  
{1: 1, 3: 1, 2: 1, 4: 1, 7: 1, 0: 1}  
{1: 2, 3: 1, 2: 1, 4: 1, 7: 1, 0: 1}  
{1: 2, 3: 2, 2: 1, 4: 1, 7: 1, 0: 1}
```

```
max_count = max(counts.values())
```

```
modes = [v for v in set(values) if counts[v] == max_count]
```

```
return modes
```

```
print(mode(sample)) #Ausgabe [1, 3] # 1 und 3 kommen am häufigsten und  
# je zweimal vor
```

Varianz

Varianz und Standardabweichung beschreiben die Unterschiede zwischen Mittelwert und jedem Datenpunkt und **wie dicht** die Daten um den Mittelwert liegen – **die Streuung**.

$$\text{Varianz der Grundgesamtheit} = \frac{(x_1 - \text{Mittelwert})^2 + (x_2 - \text{Mittelwert})^2 + \dots + (x_n - \text{Mittelwert})^2}{N}$$

Die **Varianz** ist der **mittlere quadratische Abstand** aller Werte vom **Mittelwert**.

Die **Grundgesamtheit** stellt den idealen kompletten Datenraum mit allen Werten dar. Teilmengen davon nennt man **Stichprobe**.

```
#Varianz, Beispiel für Grundgesamtheit
# Anzahl der Haustiere, die jede Person besitzt
data = [0, 1, 5, 7, 9, 10, 14]

def variance(values):
    mean = sum(values) / len(values)    # mean ist rund 6.5
    _variance = sum((v - mean) ** 2 for v in values) / len(values)
    return _variance

print(variance(data)) # Ausgabe: 21.387755102040813
```

Frage: Warum quadrieren wir?

Standardabweichung

... Die Standardabweichung ist die Wurzel aus der Varianz. Sie liegt wieder auf der gleichen Skala wie die Werte des Datensatzes und kann ggf. die Maßeinheit übernehmen (z.B. Anzahl der Haustiere)

```
#Standardabweichung, Beispiel für Grundgesamtheit
from math import sqrt

# Anzahl der Haustiere, die jede Person besitzt
data = [0, 1, 5, 7, 9, 10, 14]

def variance(values):
    mean = sum(values) / len(values)
    _variance = sum((v - mean) ** 2 for v in values) / len(values)
    return _variance

def std_dev(values):
    return sqrt(variance(values))

print(std_dev(data)) # Ausgabe: 4.624689730353898
```

Varianz und Standardabweichung einer Stichprobe

Bei einer **Stichprobe** wird durch $n-1$ (**Anzahl der Werte – 1**) **geteilt**, nicht durch die Anzahl aller Werte:

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1} \quad \text{Varianz der Stichprobe}$$

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}} \quad \text{Standardabweichung der Stichprobe}$$

Frage: Was ist der Grund für dieses Vorgehen? Betrachten Sie den Unterschied zwischen Grundgesamtheit und Stichprobe!

Die Korrektur des Nenners um minus 1 erzielt einen **größeren Wert für die Streuung**. Die Stichprobe **kann verzerrt** sein, die Streuung wird so nicht unterschätzt.

Berechnung der Standardabweichung einer Stichprobe

```
#Standardabweichung, Beispiel für Grundgesamtheit
from math import sqrt

# Anzahl der Haustiere, die jede Person besitzt
data = [0, 1, 5, 7, 9, 10, 14]

def variance(values, is_sample: bool = False):    #Boolesche Variable für Stichprobe (sample)
    mean = sum(values) / len(values)
    _variance=sum((v-mean)**2 for v in values)/(len(values)-(1 if is_sample else 0))
    return _variance

def std_dev(values, is_sample: bool = False):
    return sqrt(variance(values, is_sample))

print("VARIANCE = {}".format(variance(data, is_sample=True))) # für die Stichprobe
print("STD DEV = {}".format(std_dev(data, is_sample=True)))    # für die Stichprobe

# Ausgabe:
VARIANCE = 24.95238095238095 (war 21.38 für die Grundgesamtheit)
STD DEV = 4.99523582550223 (war 4.62 für Grundgesamtheit)
```

Visualisierungen

Welche Visualisierungen für numerische Daten kennen Sie?

➤ ...

Visualisierungen

Welche Visualisierungen für numerische Daten kennen Sie?

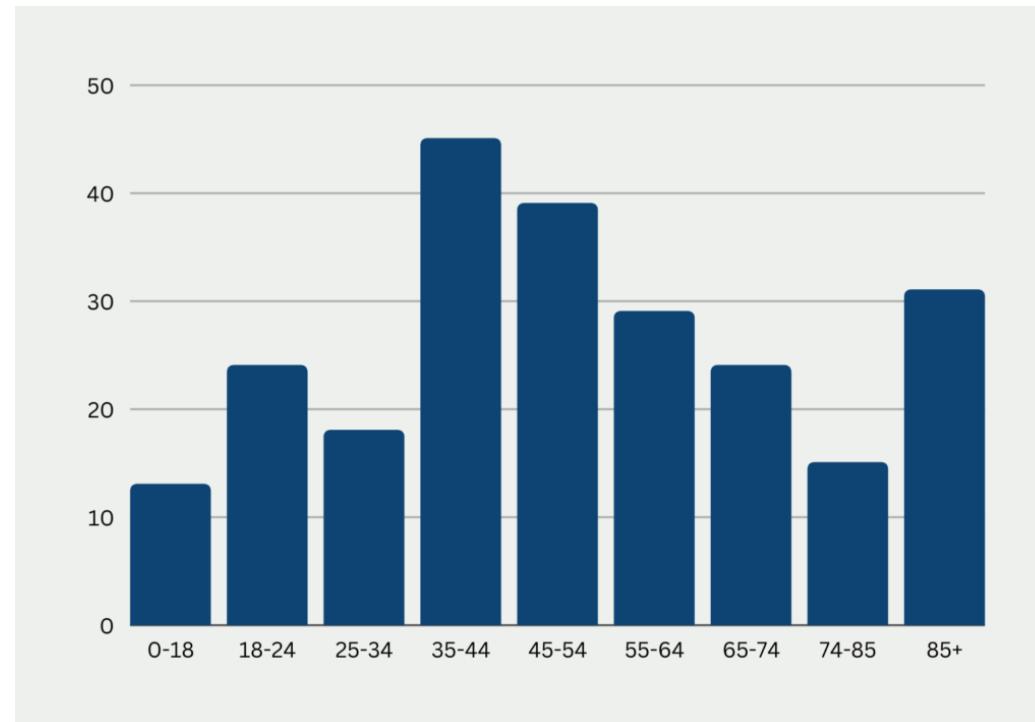
➤ ...

- Histogramme
- Scatterplots
- Boxplot
- Linienplot
- Tortendiagramm
- Balkendiagramm
- Violinenplot

Histogramme

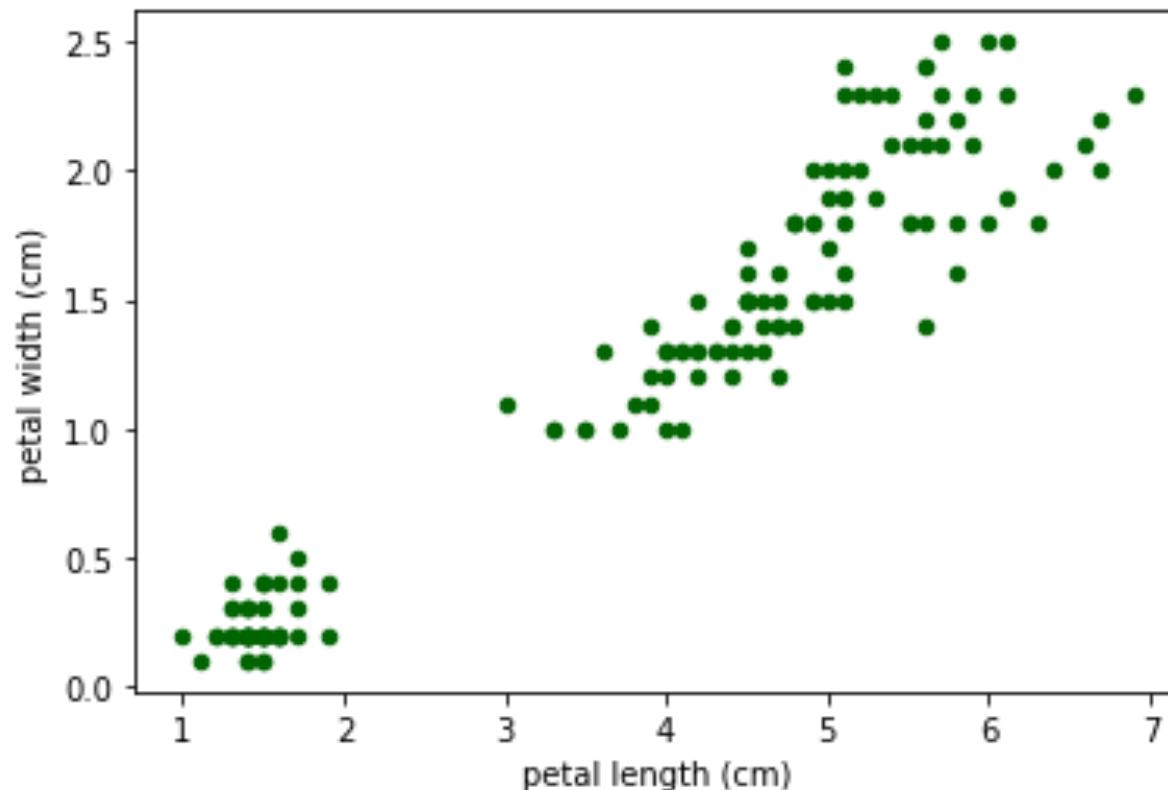
Begriff **Histogramm** kommt von „*historical diagramm*“. Sie visualisieren die **Häufigkeitsverteilung der Werte**.

Ein Histogramm besteht eigentlich aus einer statistischen Transformation von stetigen Daten in diskrete Daten, die dann gebarchartet werden.



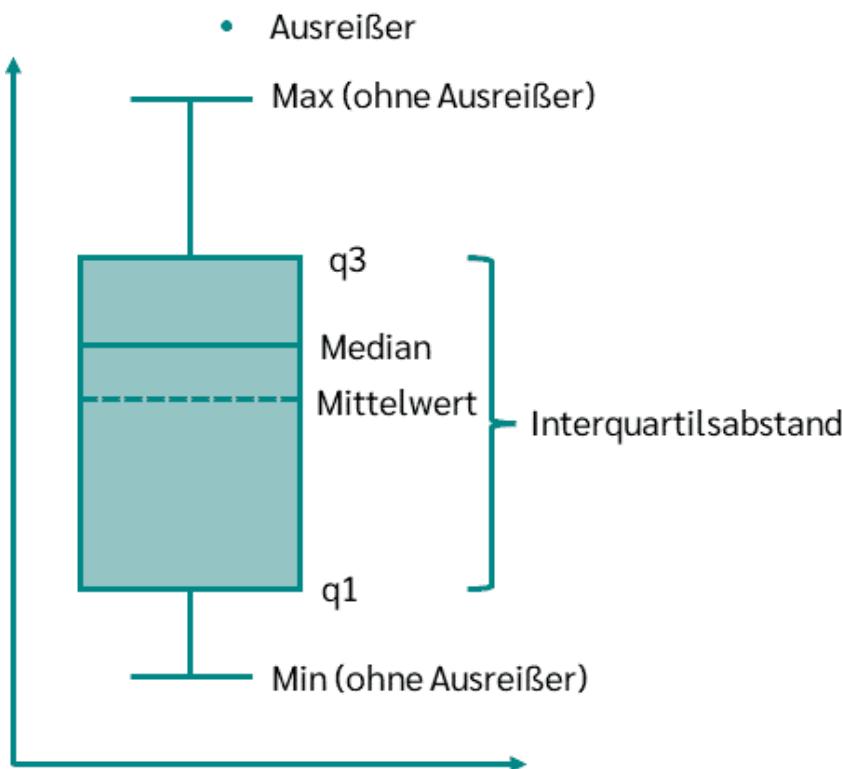
Scatterplots (Streudiagramme)

In einem **Scatterplot** lässt sich eine Menge von Vektoren darstellen, indem an den Koordinaten ein Symbol dargestellt wird, etwa ein gefüllter Kreis. Kreuz etc. Typischerweise werden bei tabellarischen Daten **zwei Variablen** ausgewählt, je eine für **die x- und die y-Achse**, das Scatterplot stellt dann die **Abhängigkeit zwischen zwei Merkmalen** in diesem Datensatz dar.



Boxplots

Ein **Box-Plot** (oder deutsch **Kastengrafik**) ist ein **Diagramm, das zur grafischen Darstellung der Verteilung eines Merkmals verwendet wird**. Es fasst dabei verschiedene robuste Streuungs- und Lagemaße in einer Darstellung zusammen.



Die Box gibt den Bereich an, in dem die mittleren 50% aller Daten liegen.

Das untere Ende der Box ist demnach das 1. Quartil und das obere Ende das 3. Quartil.

Zwischen q1 und q3 liegt damit der Interquartilsabstand

Im Boxplot gibt die durchgezogene Linie den Median an und die gestrichelte Linie den Mittelwert.

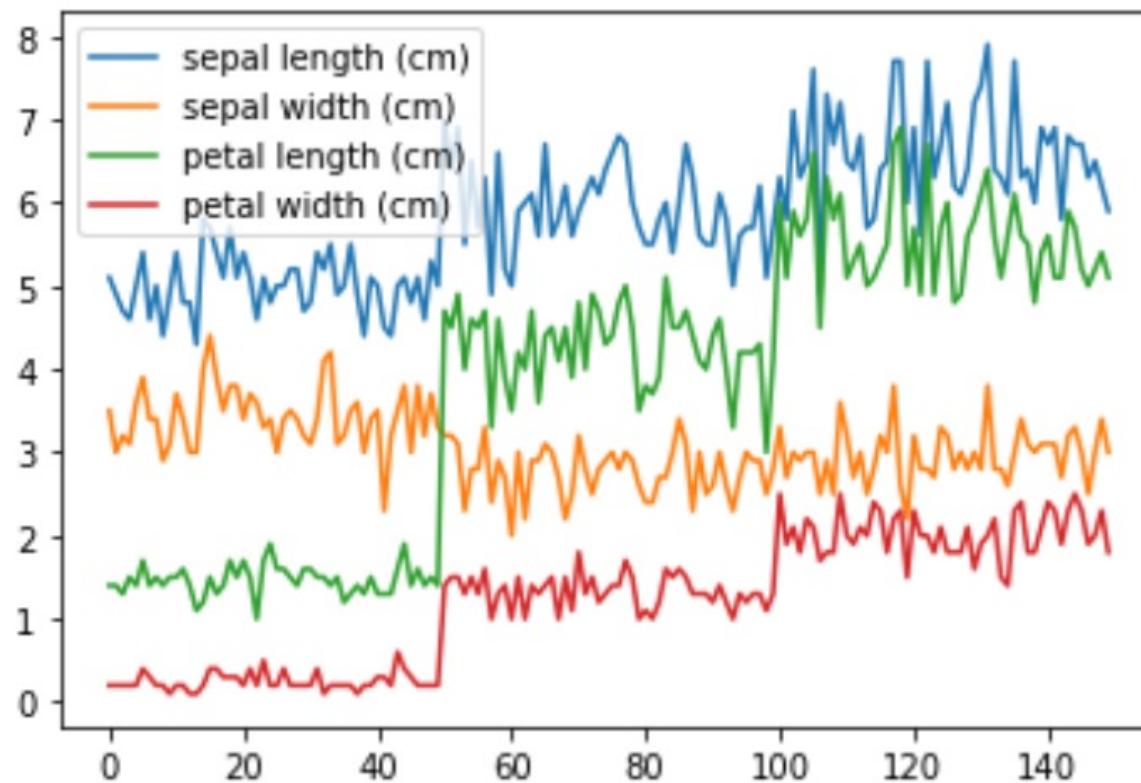
Die T-förmigen Whisker gehen bis zu dem letzten Punkt, der noch in dem 1,5-fache des Interquartilsabstands liegt.

Punkte, die weiter entfernt liegen, werden als Ausreißer betrachtet.

Liniendiagramm

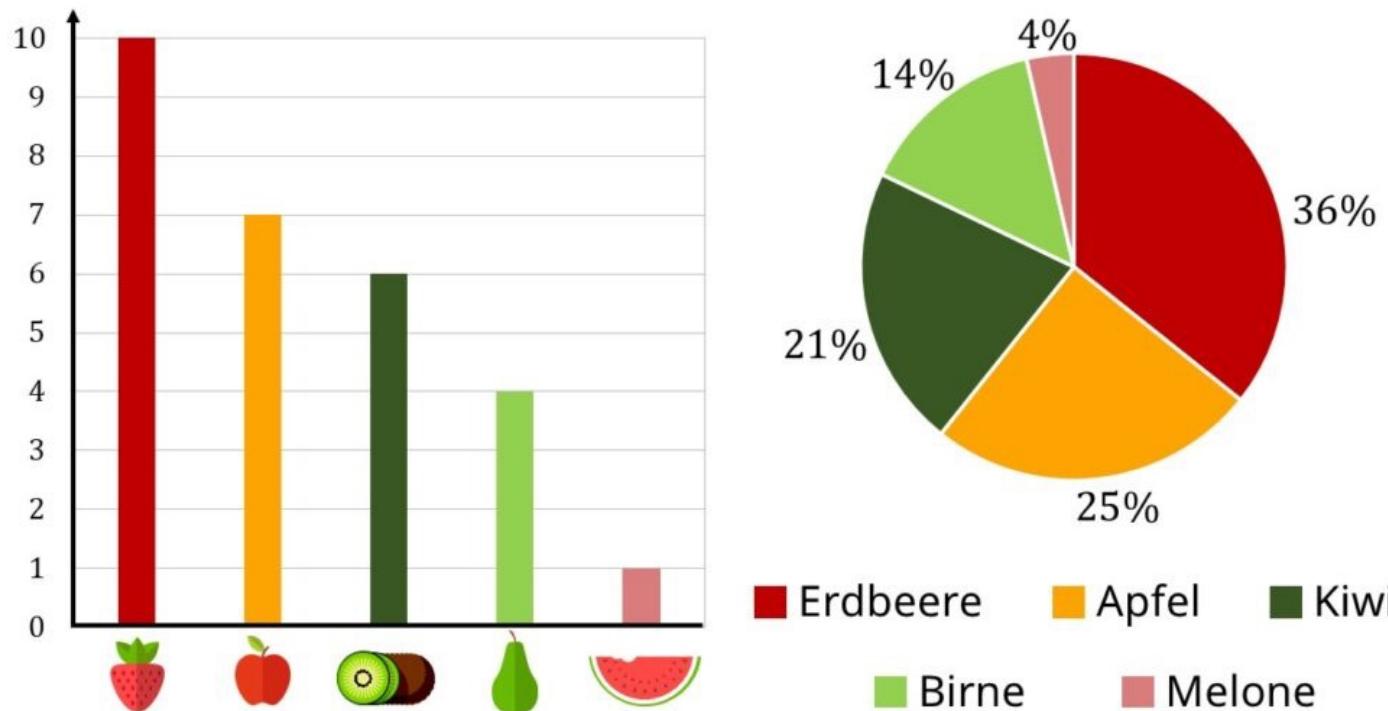
Ein **Liniendiagramm** zeichnet Daten in einer Linie (für 2 Variablen in 2D).

...Das Verbinden der Punkte suggeriert eine Datenfülle, die oft nicht vorliegt.



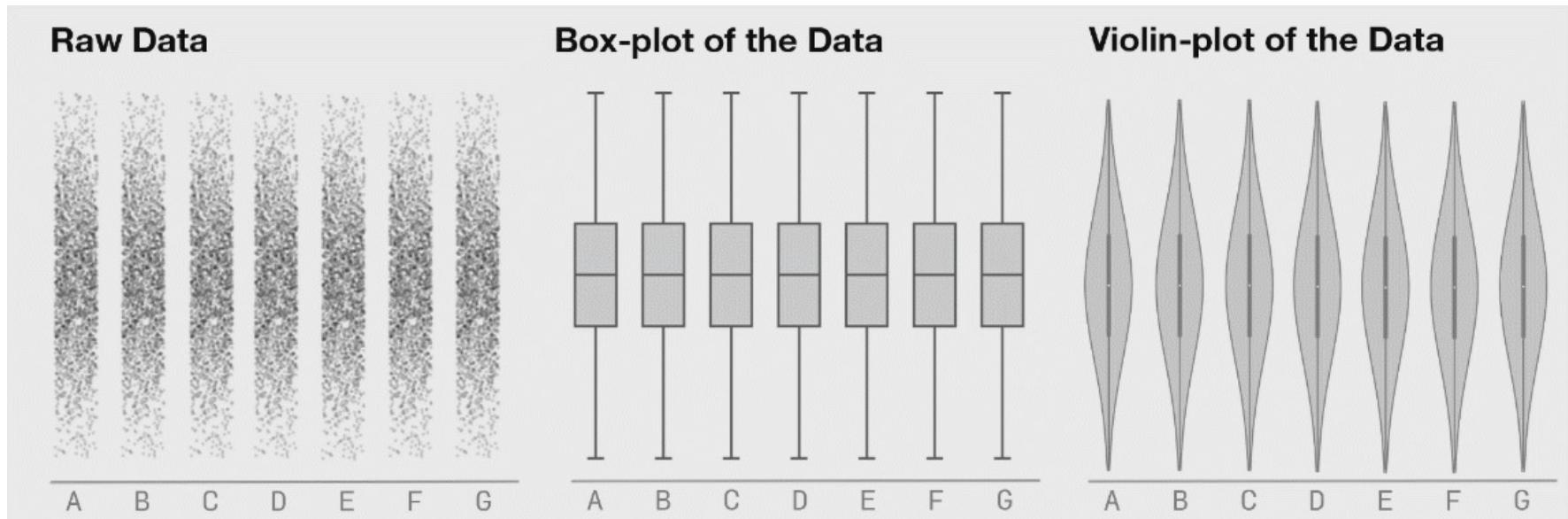
Tortendiagramm (Kreisdiagramm)

Ein **Kreisdiagramm** ist kreisförmig und in mehrere Sektoren eingeteilt. Die Gesamtsumme aller Werte wird zum ganzen Kreis (100% oder 360°) relativiert. Prinzipiell kann man auch ein **Barplot (Balkendiagramm)** anfertigen.



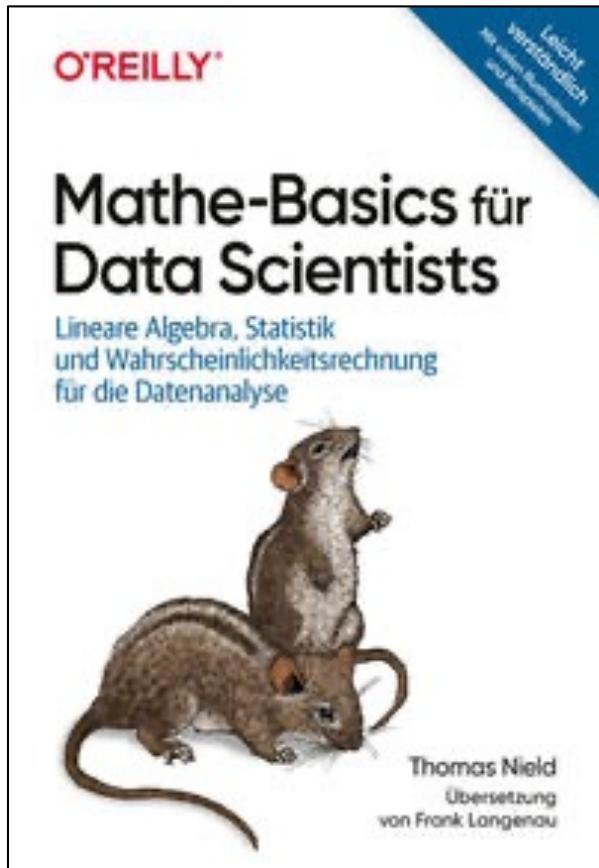
Violinenplot

Ein **Violinenplot** ist ähnlich einem Boxplot, aber anstelle eines Rechtecks wird die Dichteverteilung ausschnittsweise dargestellt.



<https://towardsdatascience.com/violin-plots-explained-fb1d115e023d>

Literaturangaben und Quellen



NIELD, Thomas und Frank LANGENAU,
2023. *Mathe-Basics für Data Scientists:
Lineare Algebra, Statistik und
Wahrscheinlichkeitsrechnung für die
Datenanalyse*. Heidelberg: O'Reilly Verlag.
ISBN 9783960107644

<https://voelkel.pages.cs.uni-duesseldorf.de/ds-skript/deskriptive-statistik.html>

Nächster Lehrinhalt

1. Einstieg zu Datenanalyse und Einführung in Maschinelles Lernen
2. **Datenanalyse mit Python**