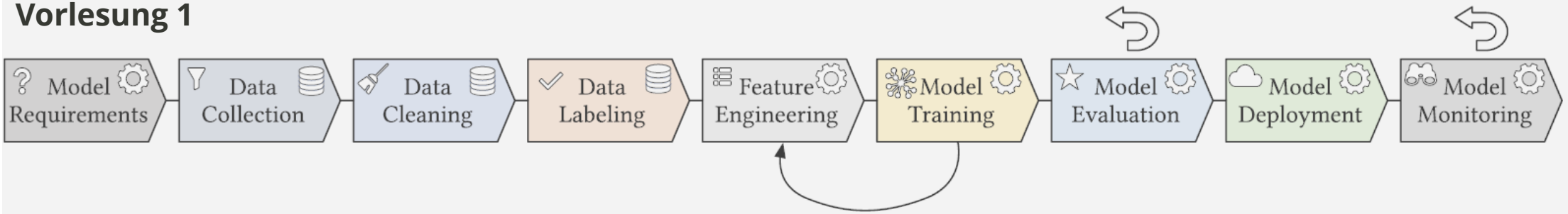
Dr. rer. nat. Valentin Khaydarov
Professur für Prozessleittechnik & Arbeitsgruppe Systemverfahrenstechnik

Daten: Gewinnung, Bereinigung, Labeling, Exploration und Feature Engineering

Vorlesung 2, Lehrveranstaltung Experimentelle Prozessanalyse

Einordnung der Vorlesung

Vorlesung 1



Vorlesung 2

Vorlesung 3 – Regr.

Vorlesung 4 – Class.

Vorlesung 5 – Clust.

Vorlesung 6 - Zeitreihenanalyse

Vorlesung 7 – Neuronale Netze

S. Amershi *et al.*, "Software Engineering for Machine Learning: A Case Study," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

Agenda

- Wiederholung
- Schritt 1: Modell requirements
- Schritt 2: Data collection und Data exploration
- Schritt 3: Data cleaning
- Schritt 4: Data labelling
- Schritt 5: Feature engineering
- Zusammenfassung und Ausblick

} **Data preprocessing**

Wiederholung der letzten Vorlesung

Wiederholung

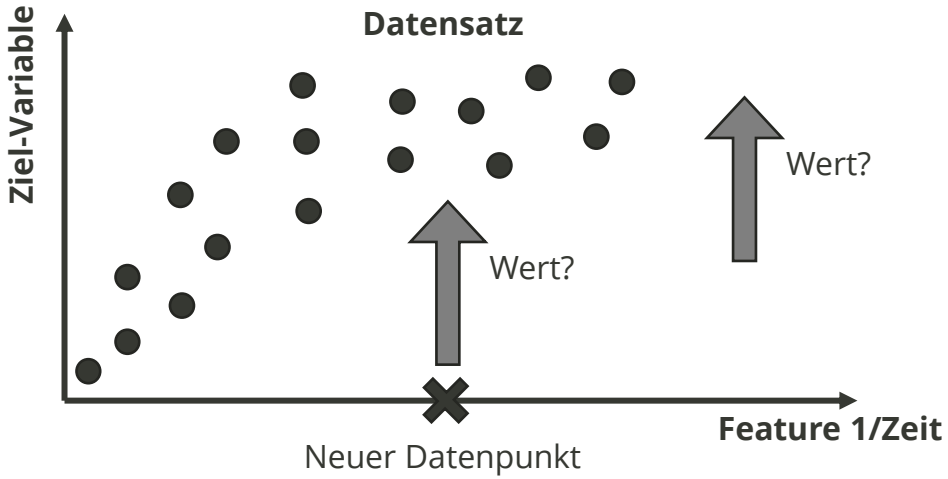
A computer program is said to learn from ??? with respect to ??? and ??? ???, if its performance on T, as measured by P, improves with experience E.

Je nach Ziel-Variable: ??? (Kardinalskala), ??? oder ??? (Ordinal- oder Nominalskala)

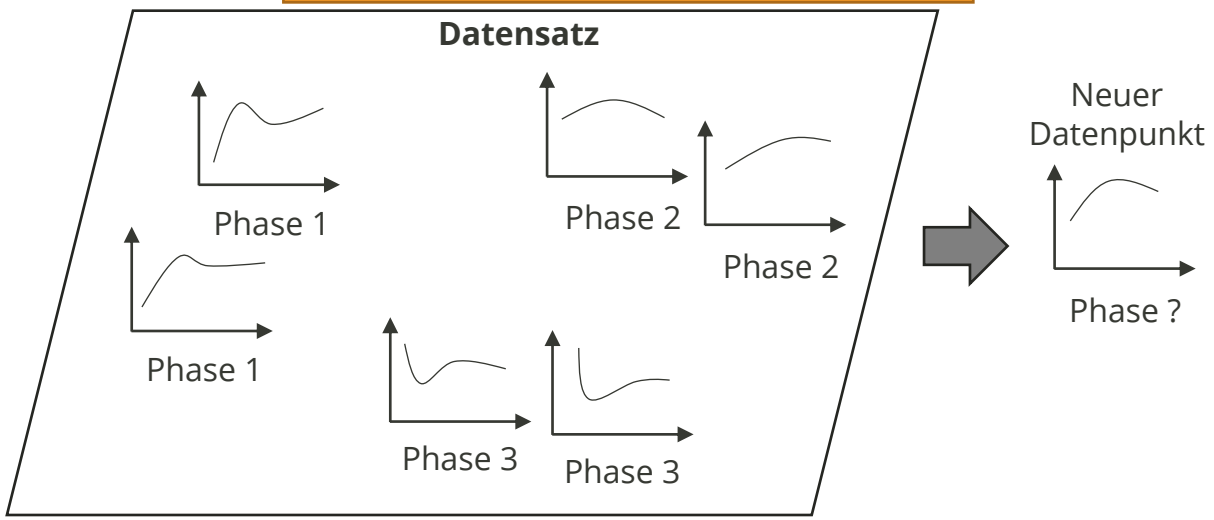
Aufgaben:

	Erfahrung	Aufgabe
Überwachtes Lernen	???	???
Unüberwachtes Lernen	???	???
Semiüberwachtes Lernen	???	???
Bestärkendes Lernen	???	???

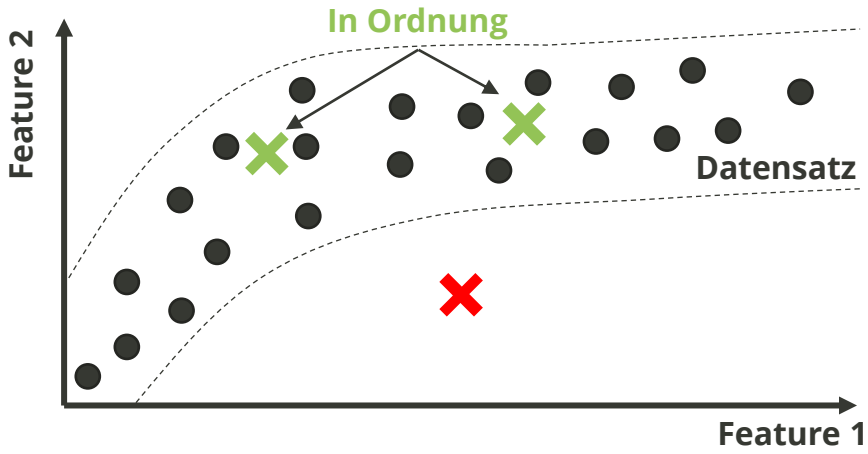
???



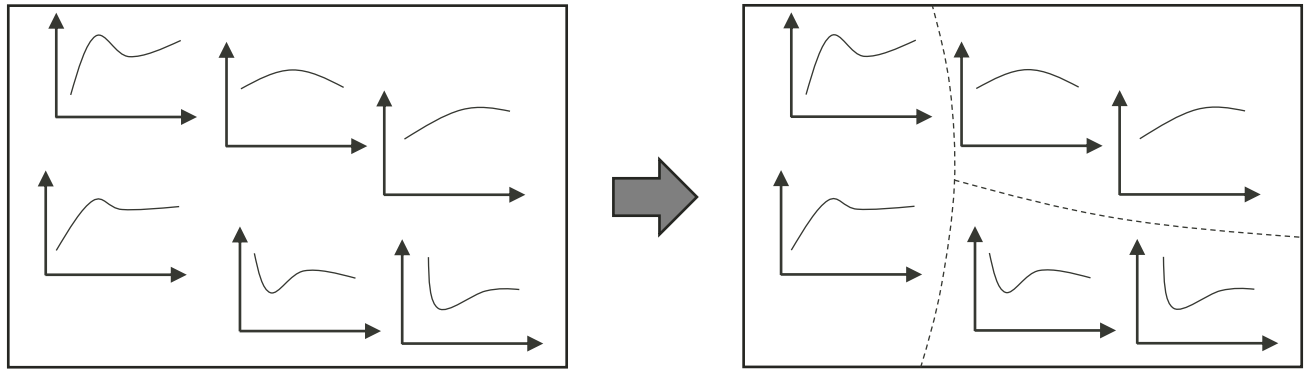
???



???

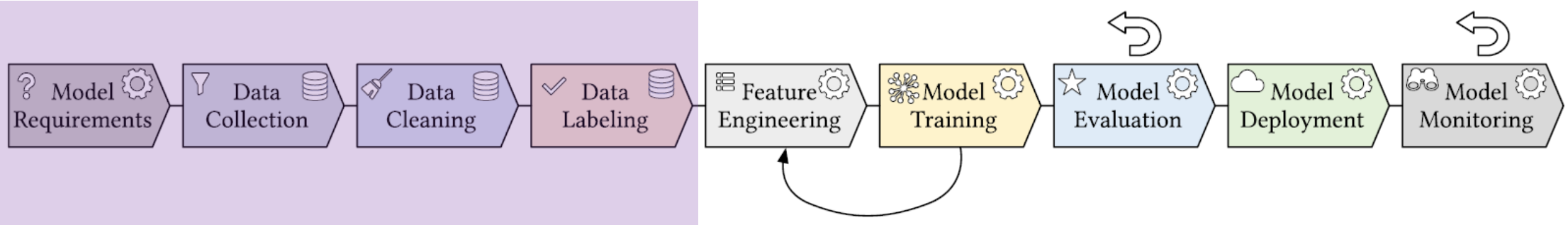


???



Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Wiederholung



Model requirement analysis (Anforderungsanalyse) – Identifikation von Rahmenbedingungen inkl. relevante Features in Daten und Vorauswahl passender Modelltypen

Input: **Anforderungen**

Arbeit mit Daten:

- Data collection – Bereitstellung des Datensatzes: Datenimport oder –gewinnung
- Data clearning – Aufbereitung des Datensatzes
- Data labeling – Markierung von Daten (Überwachstes Lernen)

Output: **Datensatz**

S. Amershi *et al.*, "Software Engineering for Machine Learning: A Case Study," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

Schritt 1:
Model requirements (inkl. Business understanding)

Formulierung der Problemstellung

Nach diesem Schritt sollen Antworten auf folgende Fragen gegeben werden:

- **Zielstellung** der zu entwickelnden Anwendung
- **Verfügbarkeit von Daten? Was genau enthalten die Daten? Wie wurden sie aufgenommen?**
  - **Daten verstehen**
- **Anforderungen** und relevante **Metriken**: Performance, Robustheit, Skalierbarkeit, Erweiterbarkeit, Echtzeitfähigkeit, Sicherheit, Ressourcenaufwand usw.
- **Minimale Anforderungen an die ML-Anwendung**
- Falls sie existieren: Wie funktionieren aktuell bestehende Lösungen? (**Baseline**)
- **Annahmen** und deren Validierung (falls möglich)
- **Wie** wird die Anwendung eingesetzt (online, offline)?
- Ist eine Modelldegradation zu erwarten und deshalb eine **Modellwartung** erforderlich?

Formulierung der Problemstellung

Das Modell ist in der Regel nur ein Teil der Anwendung und dessen Performance ist nicht immer entscheidend

Für die Modullauswahl relevante Anforderungen:

- Gehören Eingangsvariablen Kardinal-, Ordinal- oder Nominalskala? Sind die Daten eine Sequenz (Zeitreihe) oder unabhängige Messungen?
- Zielvariablen: Vorhanden? Kardinal-, Ordinal- oder Nominalskala? Vollständig und zuverlässig annotiert?
- Umfang vorhandener Daten
- Ist Expertenwissen vorhanden?
- Modellgüte bzw. Zielfunktion sowie weitere relevanten Metriken

Output: **vorläufiger ML-Versuchsplan**

Typische Problemstellungen in der Prozessindustrie

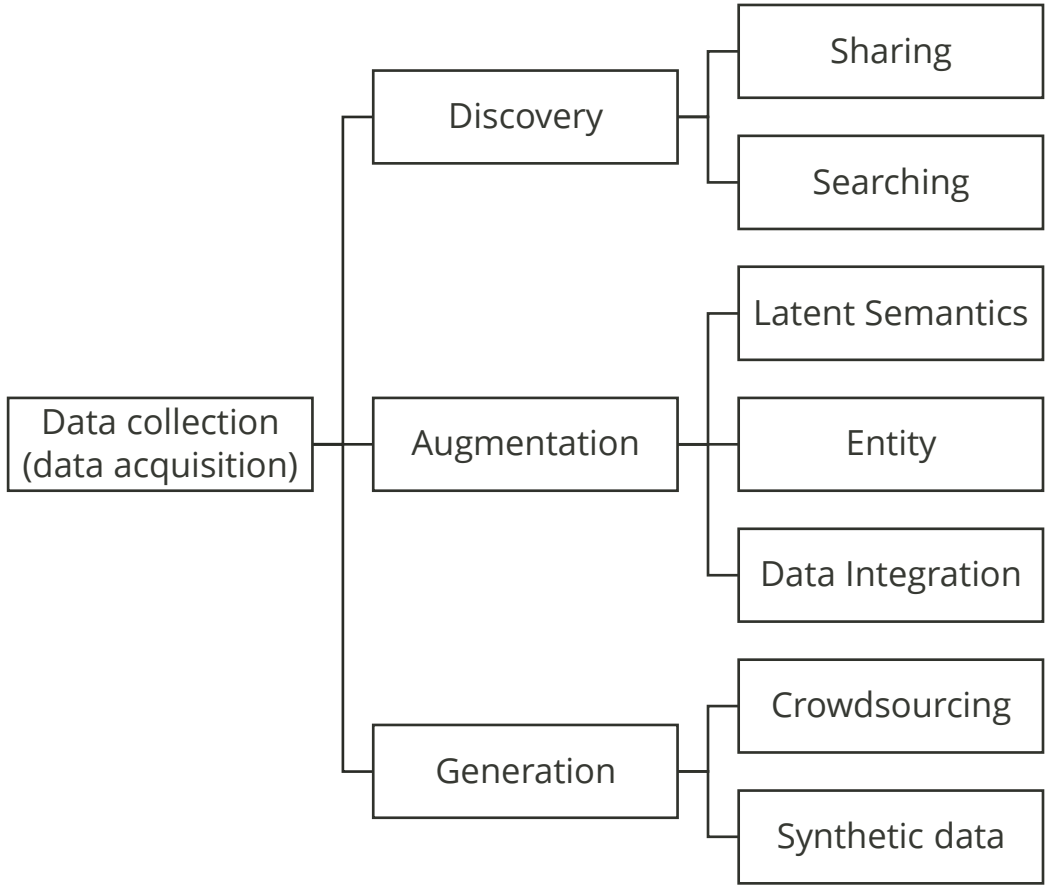
- Bewertung einer Prozessvariablen, die sich nicht direkt messen lässt → Softsensor
  - Regression oder Klassifikation
- Prädiktion des weiteren Zeitverlaufs → Prädiktive Wartung
  - Regression
- Erkennung eines Prozesszustandes oder einer bestimmten Störung
  - Klassifikation
- Anomalie-Erkennung
  - Clustering, Klassifikation
- Prozessregelung
  - Reinforcement-Learning

Schritt 2: Data collection

Data collection

Datengewinnung umfasst eine Vielfalt an Methoden zur **Beschaffung** von Daten, die weiterhin bei dem **Modelltraining** verwendet werden.

- Bis zu 90% aller Zeitaufwände bei der Entwicklung von ML-Anwendungen ist die Arbeit mit Daten
- Modellqualität hängt unmittelbar von der Datenqualität ab
- Neue Modellierungsansätze zur Abbildung komplexen Verhaltens (z.B. DeepNN) erfordern proportional mehr Daten für das Training



Y. Roh, G. Heo, and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2019, doi: 10.1109/TKDE.2019.2946162.

Industrielle Daten

Datentypen:

- **Prozessdaten von Sensoren sowie Signalgebern**, die primär für Prozessüberwachung, - optimierung und Regelung eingesetzt werden
  - Temperatur, Druck, Füllstand, Stoffeigenschaften, Sollwerte usw. in der Regel Daten der Kardinal- oder Ordinalskala (analoge und binäre Signale) als Zeitreihen
  - Verfügbar über das Prozessleitsystem und archiviert in einem Prozesshistorian oder offline als Protokoll
- **Diskrete Ereignisse** wie Prozessphase oder Rezeptschritt, Operatoreingriffe, Alarme der Nominalskala
  - In der Regel unstrukturiert und teilweise manuell aufgenommen
- **Engineeringdaten zur Anlage**
  - Meistens statische und **unstrukturierte** Daten
  - In der Regel in der Form von Dokumentation zur Produktionsanlage wie R&I-Fließbild, Konstruktionsdokumente, aktuelle Anlagekonfiguration, Regelungskreise, Rezepte usw.
  - Liefert sehr nützliche Kontextinformation für besseres Verständnis von Daten
- **Rohdaten von Analysegeräte (z.B. Spektroskopie, Chromatographie, Laserbeugung), Bild- und Videodaten**
  - Integriert mit relativ komplexen Preprocessing- und Datenauswertung-Algorithmen
  - Gekapselt in von dem SPS entkoppelten Systemen (z.B. Edge-Node-Konzept)

Besonderheiten und Herausforderungen industrieller Daten

Datenvarianz je nach Betriebsweise der Anlage:

- Batch-Betrieb: Varianz zwischen Chargen durch Änderungen im Startbedingungen und Wiederholungen
- Konti-Betrieb: Varianz ist in der Regel nur bedingt aussagekräftig → Data-rich-but-information-poor (DRIP)-Daten
- Daten sind in der Regel nicht nach Klassen bilanziert

Fluch der Dimensionalität:

- Enorme Anzahl diverser und möglicher Datenquellen: Dutzende oder Hunderte Sensoren und Signalgeber

Zeit:

- Je nach der Prozessskala können Messwerte fein ausgelöst sein (Millisekunden-Bereich)
- Datenaufnahme verschiedener Sensoren in der Regel nicht synchronisiert (s. OPC UA)
- Time-Sampling kann im Laufe der Zeit variieren (ms-Bereich im Betrieb und Min-Bereich im Prozesshistorian)

Besonderheiten und Herausforderungen industrieller Daten

Big-Data:

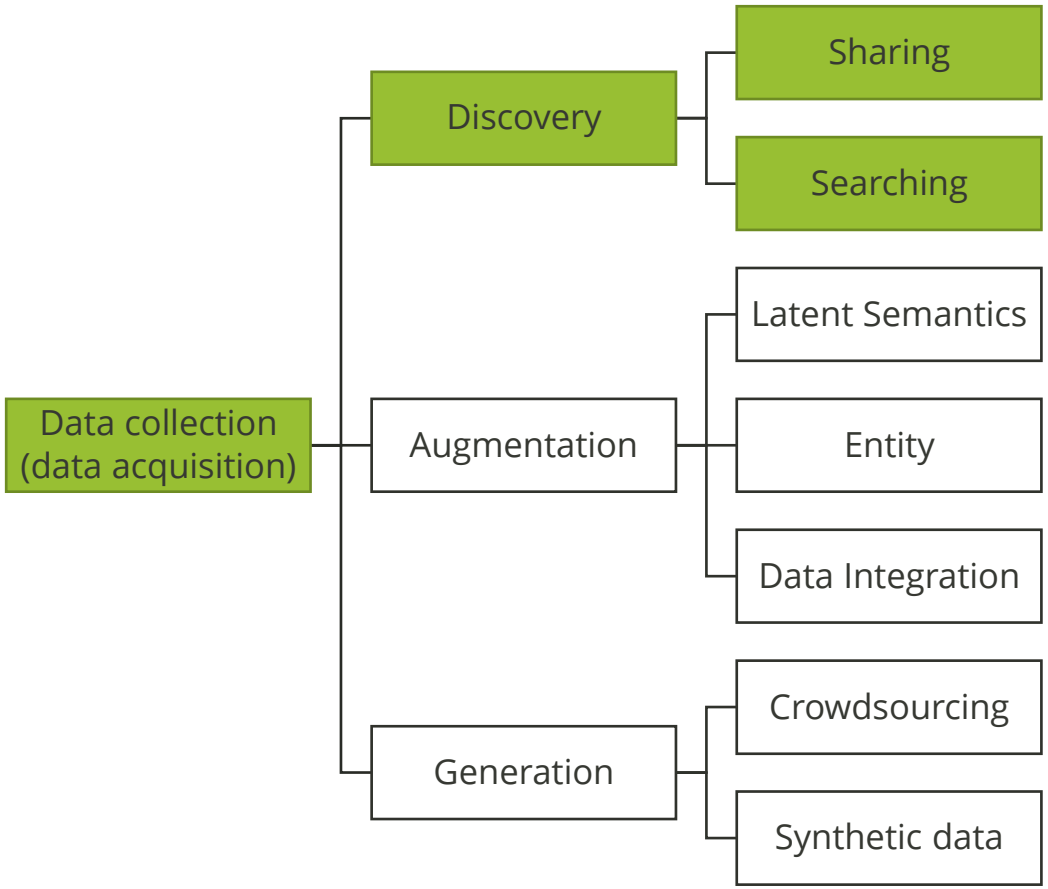
- Prozesse können bis zu vielen Tagen dauern:
  - ein Parameter, eine Messung pro Sekunde, eine Woche lang: 60x60x24x7 = 604800 Messpunkte

Einbindung von Kontextinformation:

- Normen schreiben Anlagenbauern und –betreibern die Durchführung einer sehr umfangreichen Anlagendokumentation vor
- Verwendung dieser Information zur ausführlichen Meta-Beschreibung von Datensätzen

Weitere spezifische datenbezogene Herausforderungen werden weiter in den Abschnitten „Data cleaning“ und „Data labeling“ betrachtet

Data discovery



Y. Roh, G. Heo, and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2019, doi: 10.1109/TKDE.2019.2946162.

Data discovery

Methoden für das **Teilen** und die **Wiederverwendung** von bestehenden Datensätzen.

Allererster Schritt - Suche nach relevanten Datensätzen (lokal und global).

Voraussetzung ist die Erfüllung der FAIR-Prinizipen:

- **Findable**: Eindeutiger und dauerhafter Identifikator vorhanden und Metadaten erlauben die Findung
- **Accessible**: Standardisierte Protokolle für Datenzugriff
- **Interoperable**: Daten sind in einer (Computer-) lesbaren und interpretierbaren Form
- **Reusable**: Daten sind zusammen mit ausführlicher Beschreibung und Lizenzbedingungen zur Verfügung gestellt

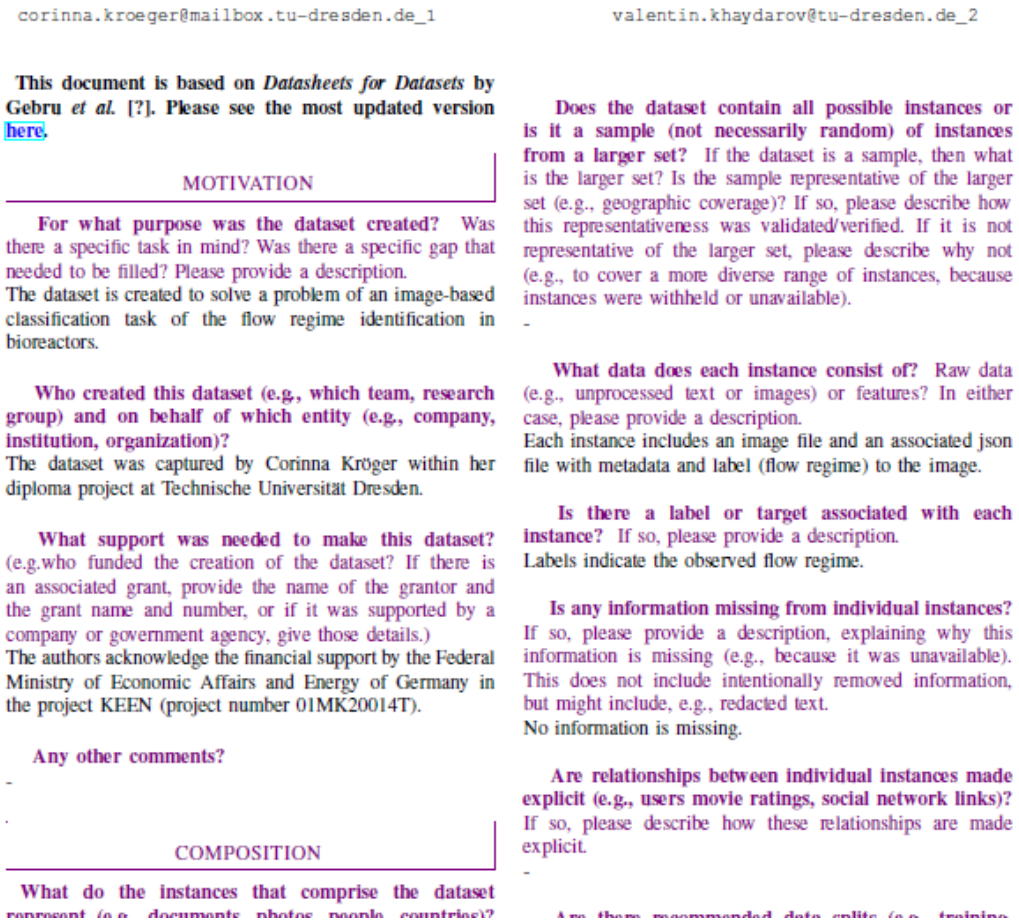
Mehr auf https://www.go-fair.org/fair-principles/

Data discovery: Datasheet for Dataset

Abschnitte:

- Motivation
- Composition
- Collection
- Preprocessing/Cleaning/Labeling
- Uses
- Distribution
- Maintenance

Quelle: Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., & Crawford, K. (2018). Datasheets for datasets. ArXiv.



corinna.kroeger@mailbox.tu-dresden.de_1 valentin.khaydarov@tu-dresden.de_2

This document is based on *Datasheets for Datasets* by Gebru *et al.* [?]. Please see the most updated version here.

MOTIVATION

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.
The dataset is created to solve a problem of an image-based classification task of the flow regime identification in bioreactors.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?
The dataset was captured by Corinna Kröger within her diploma project at Technische Universität Dresden.

What support was needed to make this dataset? (e.g.who funded the creation of the dataset? If there is an associated grant, provide the name of the grantor and the grant name and number, or if it was supported by a company or government agency, give those details.)
The authors acknowledge the financial support by the Federal Ministry of Economic Affairs and Energy of Germany in the project KEEN (project number 01MK20014T).

Any other comments?
-

COMPOSITION

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).
-

What data does each instance consist of? Raw data (e.g., unprocessed text or images) or features? In either case, please provide a description.
Each instance includes an image file and an associated json file with metadata and label (flow regime) to the image.

Is there a label or target associated with each instance? If so, please provide a description.
Labels indicate the observed flow regime.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.
No information is missing.

Are relationships between individual instances made explicit (e.g., users movie ratings, social network links)? If so, please describe how these relationships are made explicit.
-

Are there recommended data splits (e.g., training

Data discovery: Datenrepositories

Datenrepositories (Daten intern gespeichert):

Kaggle.com, Amazons opendata.aws, UCI ML Repository, Microsofts msropendata.com, EU Open Data Portal, OpenML

Die meistverwendeten sind direkt in Tools integriert: Matlab (s. Sample Data Sets), sklearn (s. Toy datasets) und tensorflow (s. Tensorflow-Datasets)

Suchportale (Nur Verweise auf externe Quellen):

Google Datasets Search Engine

Listen von Datenrepositories:

https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research

https://github.com/awesomedata/awesome-public-datasets

In der Regel nur Daten für allgemeine Zwecke (viel Bilddaten). **Prozessdaten sind kaum öffentlich verfügbar.**

Prominente industrielle Datensätze

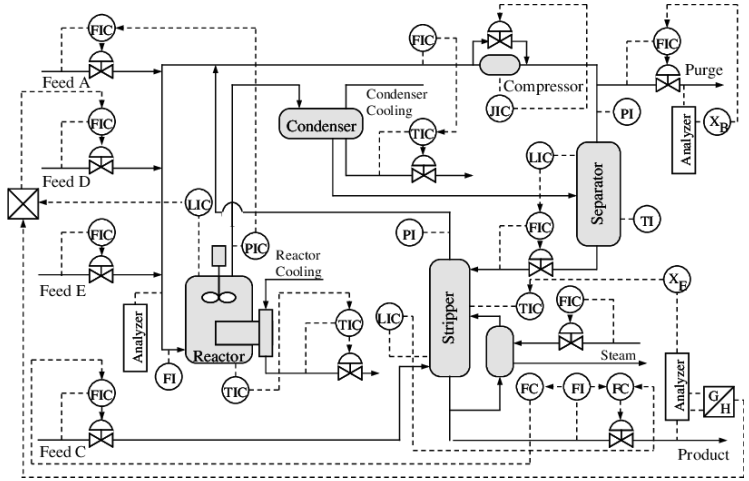
Tennessee Eastman Process (TEP)

Down und Vogel, 1993

Open-Loop-Prozess mit 12 Regelventilen und 41 Messstellen

Als Challenge für Störungserkennung und dynamische Optimierung

Simulationsmodell und reale Datensätze auffindbar



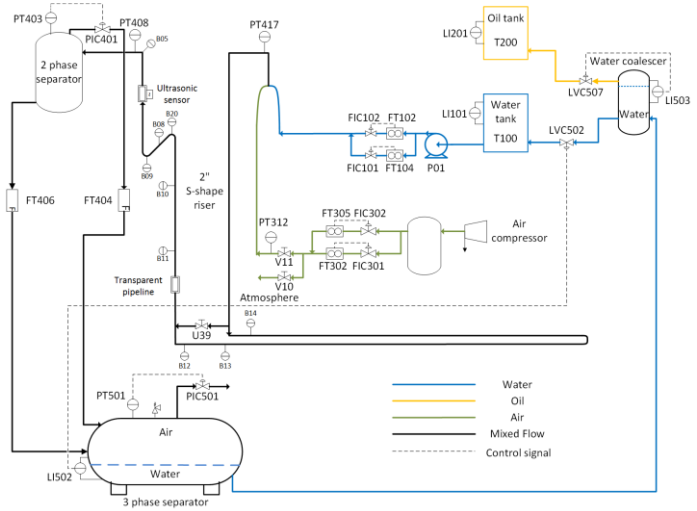
Kano u. a., „Contribution Plots for Fault Identification Based on the Dissimilarity of Process Data“.

Multiphase flow facility (PRONTO)

Stief, Tan, Cao und Ottewill, 2019

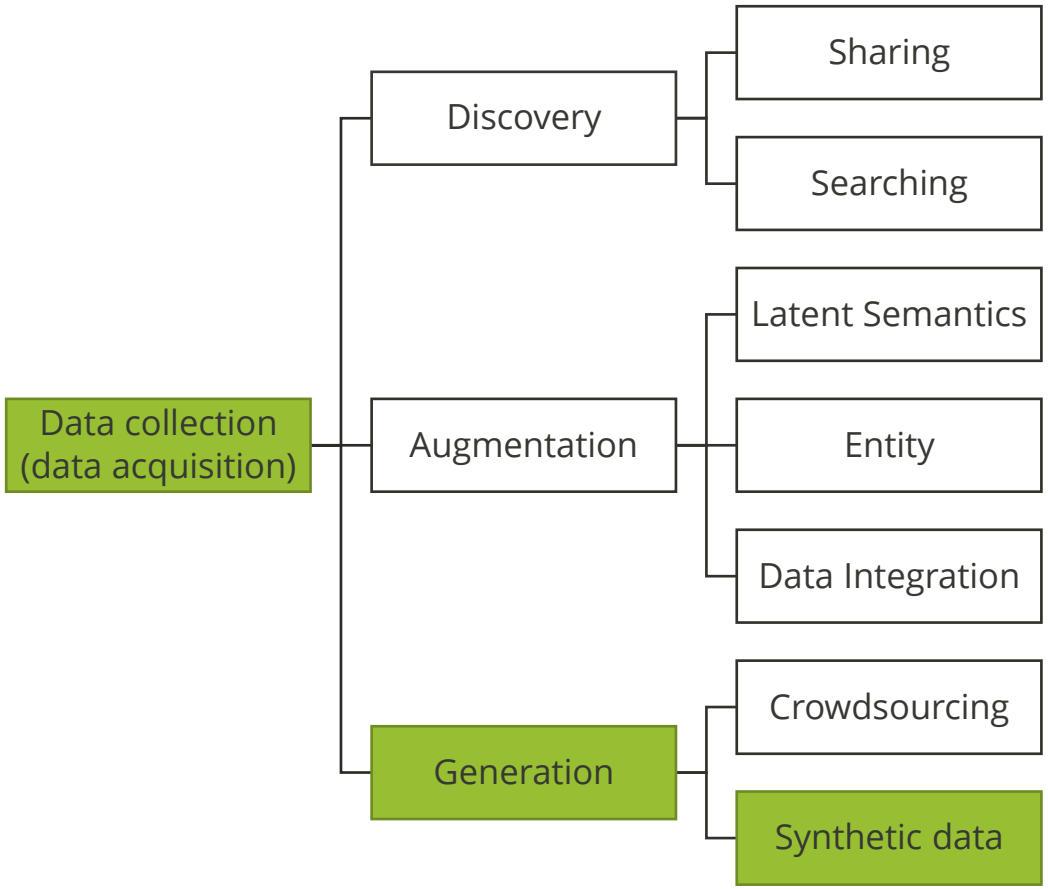
Heterogene Datenquellen (Prozesswerte, Alarme, Logs, Hochfrequente Ultraschall- und Druckmessungen und Videodaten)

Multimodale und Multirate Störungserkennung und Anlagediagnose



https://doi.org/10.5281/zenodo.1341583

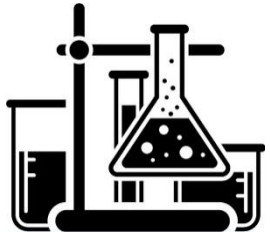
Data generation



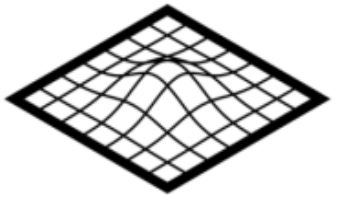
Y. Roh, G. Heo, and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2019, doi: 10.1109/TKDE.2019.2946162.

Data generation: Datenquellen

Produktion
- Vollständig repräsentativ
- Große Menge
- Kleine Varianz
- Datenkosten niedrig
- Nur intern verfügbar



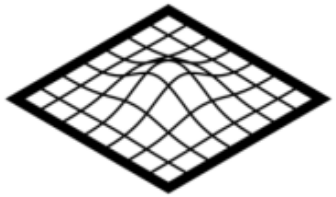
Test-Umgebung
- Wenig repräsentativ
- Kleinere Datenmenge
- Höhere Varianz
- Datenkosten höher
- Dürfen veröffentlicht werden



Simulationsmodelle
- Bilden Modell ab
- So viel wie man will
- Beliebige Varianz
- Niedrige Kosten
- Grundsätzlich frei teilbar

Transfer-Learning ist der Einsatz Erfahrung auf ein ähnliches aber anderes Problem zu übertragen. Sehr beliebt und effektiv in Computer-Vision-Anwendungen.

Data generation und Transfer learning



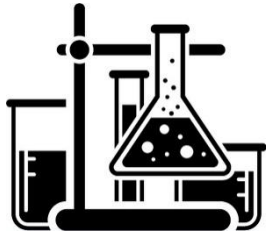
Synthetische Daten
(Simulation)

Realle Daten

Transfer-
Learning

Pre-trainiertes Modell (Source)

Modell (Target)



Daten
(Test-Umgebung)

Realle Daten

Transfer-
Learning

Pre-trainiertes Modell (Source)

Modell (Target)

Schritt 2a: Data Exploration

Data exploration

Datenexploration ist ein Verfahren zur vorläufigen Analyse von Daten.

Datenexploration ist der allererste Schritt nach der Erwerbung von Daten.

Teilziele:

- **Formalisierung** Information über Variablen: Beschriftung, Typ, Einheit, Grenzen/Mögliche Werte
- Verschaffung eines **Überblicks** über und **Verständnis** von Daten
- **Initiale Analyse** von Mustern und Charakteristika der Daten
- Formulierung **initialer Hypothesen** für Modellierung
- **Vorplanung nächster Schritte** für Datenbereinigung, Datenaufbereitung, Featureauswahl sowie Vorauswahl geeigneter Modelltypen

Beschreibende Statistik

Histogramme und empirische Verteilungsfunktion

Absolute Häufigkeit:

- Anzahl der Beobachtungswerte mit einer bestimmten Ausprägung

$$h(a_j) = h_j$$

Relative Häufigkeit:

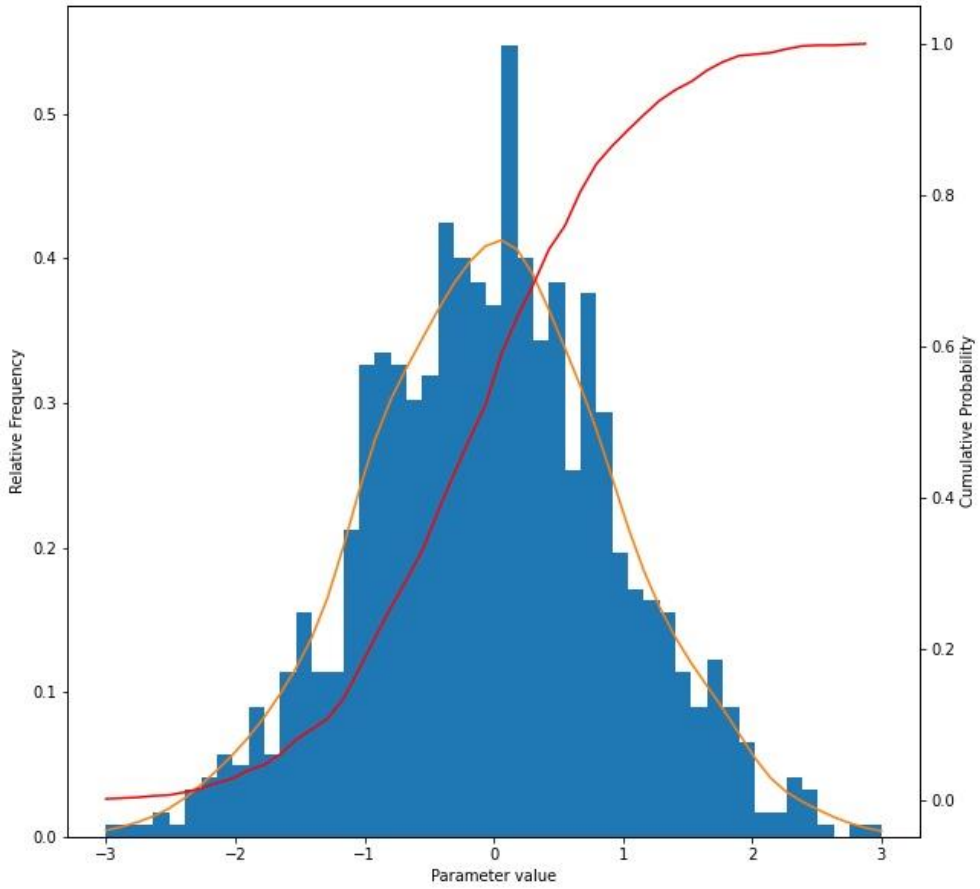
- Anzahl der Beobachtungswerte mit einer bestimmten Ausprägung durch die der Merkmale

$$f(a_j) = \frac{h(a_j)}{n}$$

Relative Summenhäufigkeit der Klasse:

- Aufsummierte relative Häufigkeiten der Klassen:

$$H_j = \sum_{l=1}^{j} \frac{h_j}{n}$$



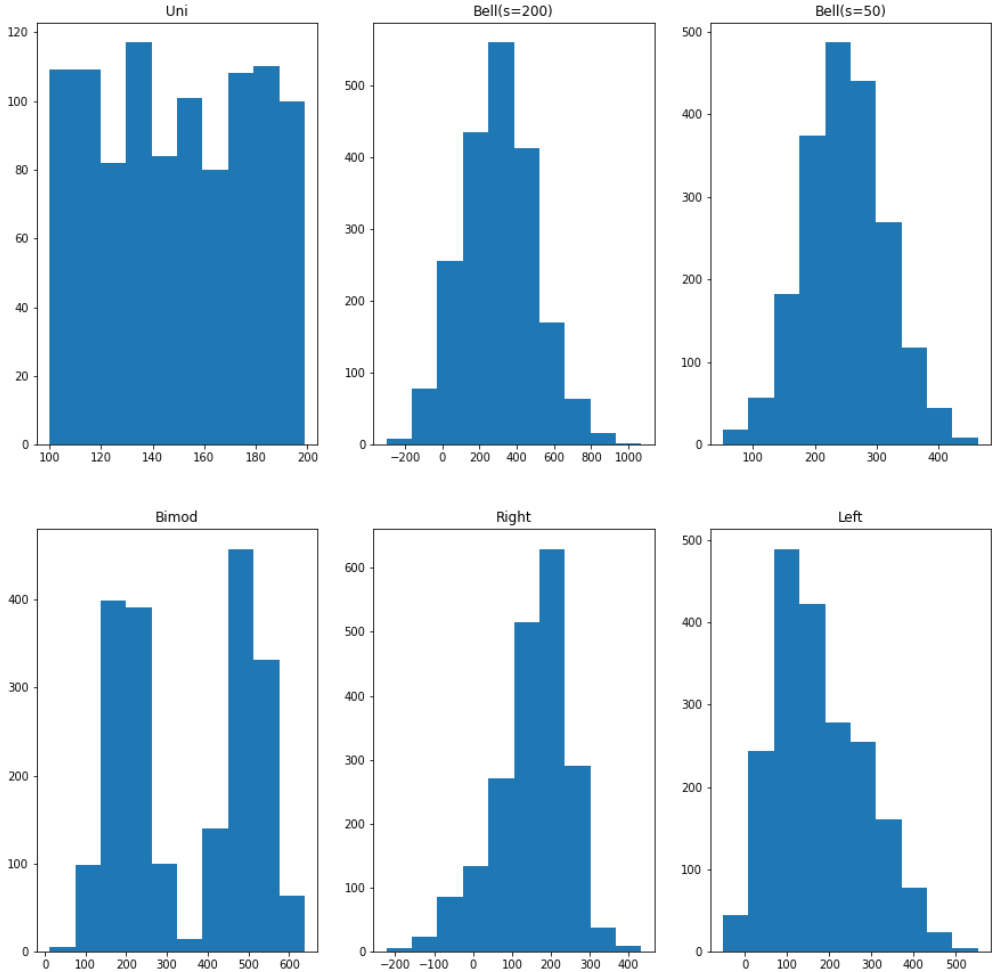
Beschreibende Statistik
Empirische Verteilungsfunktion

Symmetrie
- Symmetrisch
- Asymmetrisch

Modalität
- Unimodal
- Bimodal
- Multimodal

Breite
- Schmalgipflig
- Breitgipflig

Schiefe
- Linkssteil (rechtsschief)
- Rechtssteil (linksschief)



Beschreibende Statistik
Kennzahlen für ein Merkmal

- Beschreibung der Datenmenge mit wenigen Größen
- Informationsverdichtung
- Vergleich von Messreihen

Unterteilung in:

- empirische Maße der Lage (Mittelwertmaße)
- empirische Streuungsmaße (Streuungsmaße)

$\underline{\boldsymbol{x}} = (x_1, \dots, x_n)^T$ - Messwertvektor eines beliebigen Merkmals (Spalte der Datenmatrix)

$\underline{\boldsymbol{x}}_s = \left(x_{(1)}, \dots, x_{(n)}\right)^T$ mit $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)},\ x_{(i)} \in \{x_1, \dots, x_n\}$ - der zugehörige der Größe nach sortierte Vektor

Beschreibende Statistik

Kennzahlen für ein Merkmal - empirische Maße der Lage

Minimum: $x_{min} = x_{(1)}$

Maximum: $x_{max} = x_{(n)}$

Mittelwert: $m = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Median:
$$m_e = \begin{cases} x_{\left(\frac{n+1}{2}\right)} & \text{für n ungerade} \\ \frac{x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}}{2} & \text{für n gerade} \end{cases}$$

Modalwert: $D = x_i$ wo x_i ist der häufigste Wert

Beschreibende Statistik

Kennzahlen für ein Merkmal - empirische Maße der Lage

Quantile: empirisches Quantil der Ordnung q $(0 < q < 1)$:

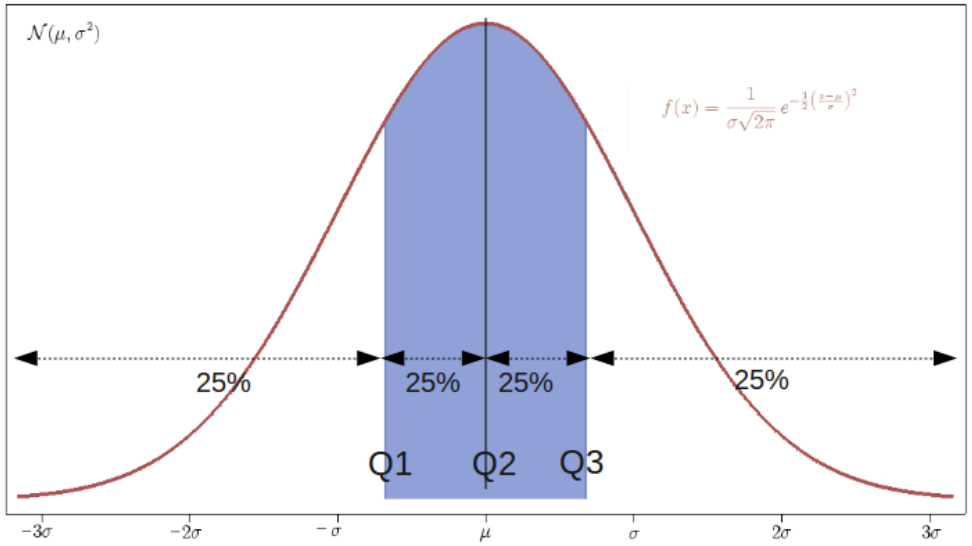
$$\tilde{x}_q = \begin{cases} x_{([nq]+1)} & \text{falls } nq \text{ keine ganze Zahl ist} \\ \dfrac{x_{(nq)} + x_{(nq+1)}}{2} & \text{falls } nq \text{ ganze Zahl} \end{cases}$$

Wobei $[a]$ den ganzzahligen Anteil von a bezeichnet

Es gilt: $m_e = \tilde{x}_{0,5}$

Besondere Bedeutung besitzen auch:

- **untere empirische Quartil** $\tilde{x}_{0,25}$
- **obere empirische Quartil** $\tilde{x}_{0,75}$



https://en.wikipedia.org/wiki/Quantile#/media/File:Iqr_with_quantile.png

Beschreibende Statistik

Kennzahlen für ein Merkmal - empirische Streuungsmaße

Spannweite: $x_{(n)} - x_{(1)} = x_{max} - x_{min}$

Streuung: $\sigma = s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ (s: Standardabweichung)
(Varianz)

F-Spanne: $s_F = \tilde{x}_{0,75} - \tilde{x}_{0,25}$ (F-Spread; empirischer Quartilabstand)

Die F-Spanne liefert Hinweise auf mögliche Ausreißer im Datenmaterial:

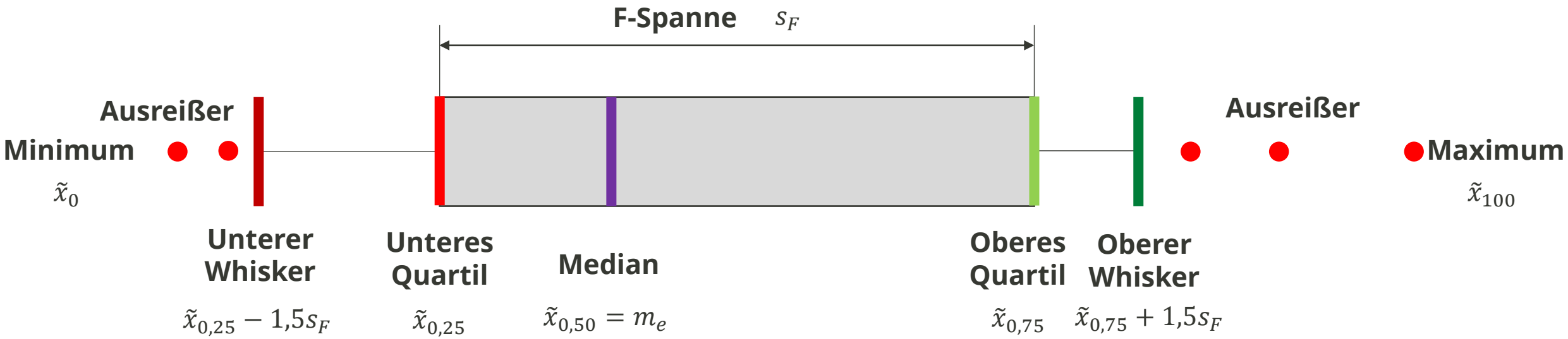
- Messwerte, die größer als $\tilde{x}_{0,75} + 1{,}5s_F$
- Messwerte, die kleiner als $\tilde{x}_{0,25} - 1{,}5s_F$

Mittlere absolute Abweichung (AAD): $D_{AAD} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$

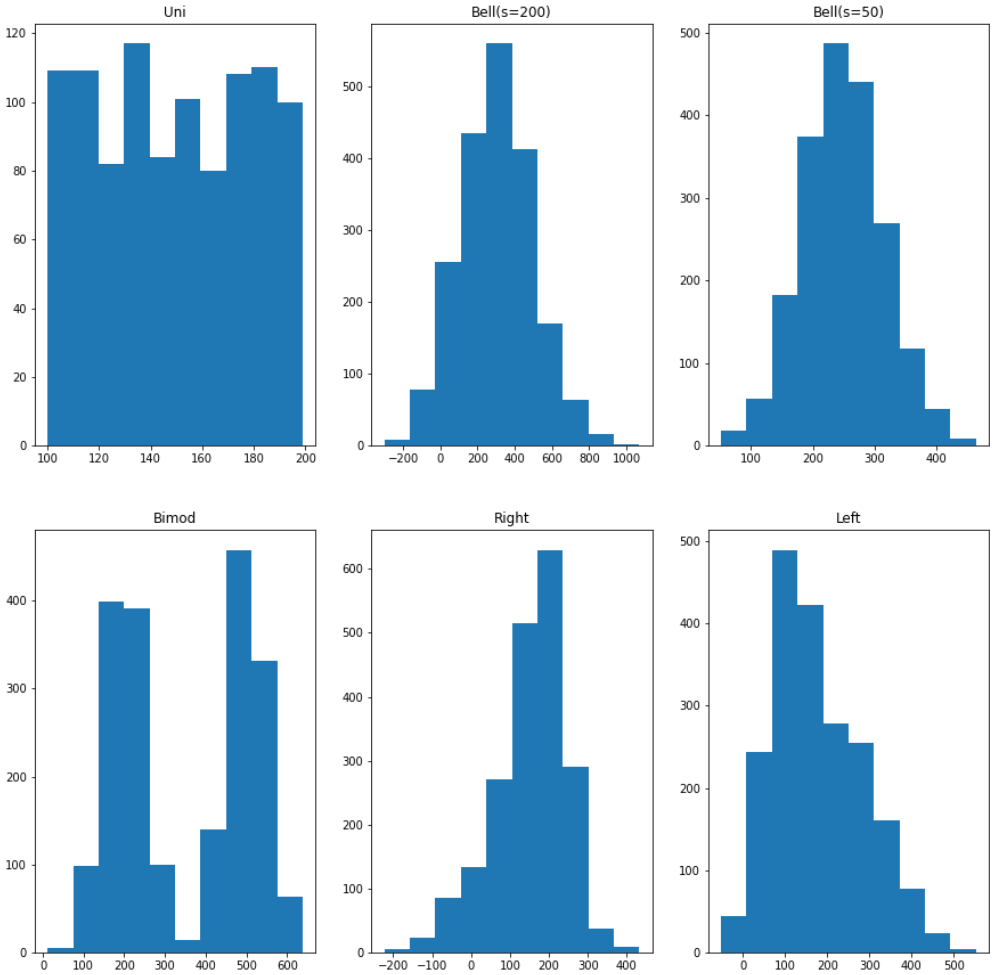
Beschreibende Statistik

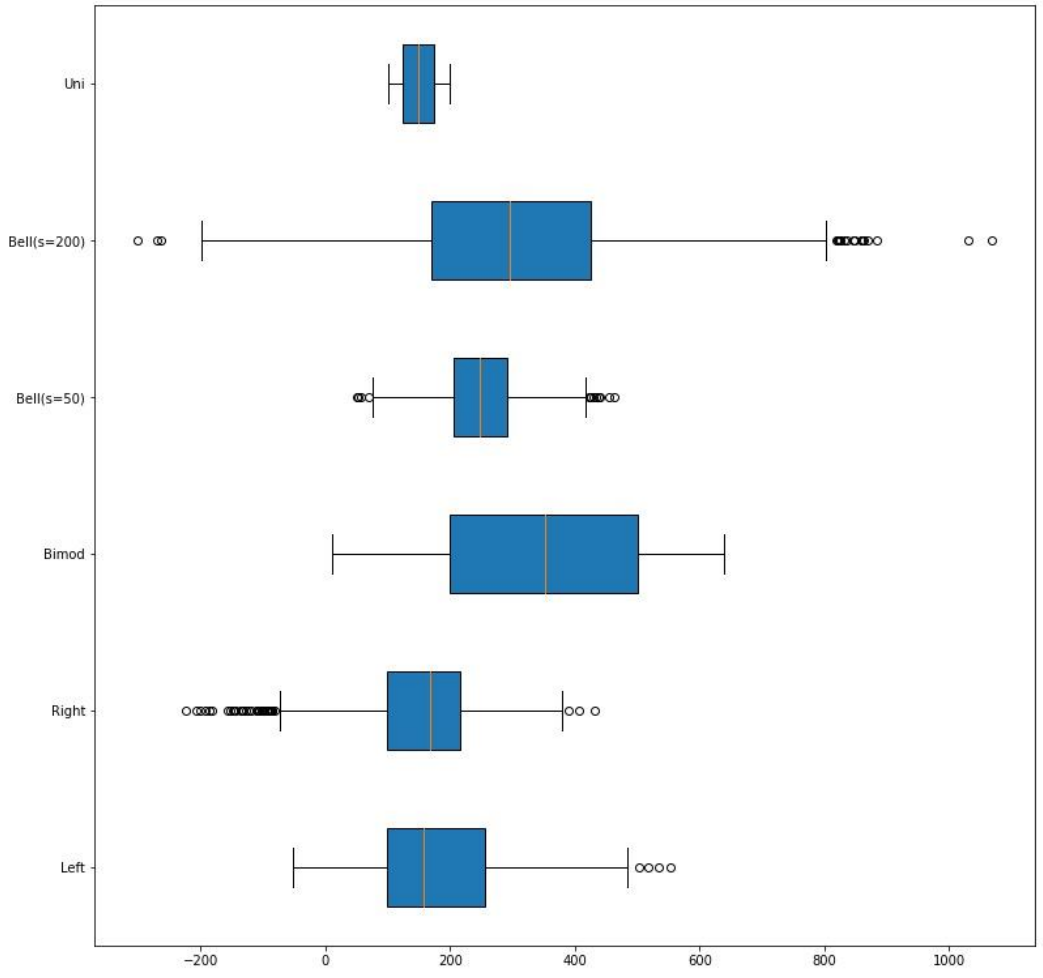
Kennzahlen für ein Merkmal - BOX-PLOT (Box-Whisker-Plot)

kompakte graphische Darstellung der Messwerte (mittels empirischer Kennwerte)



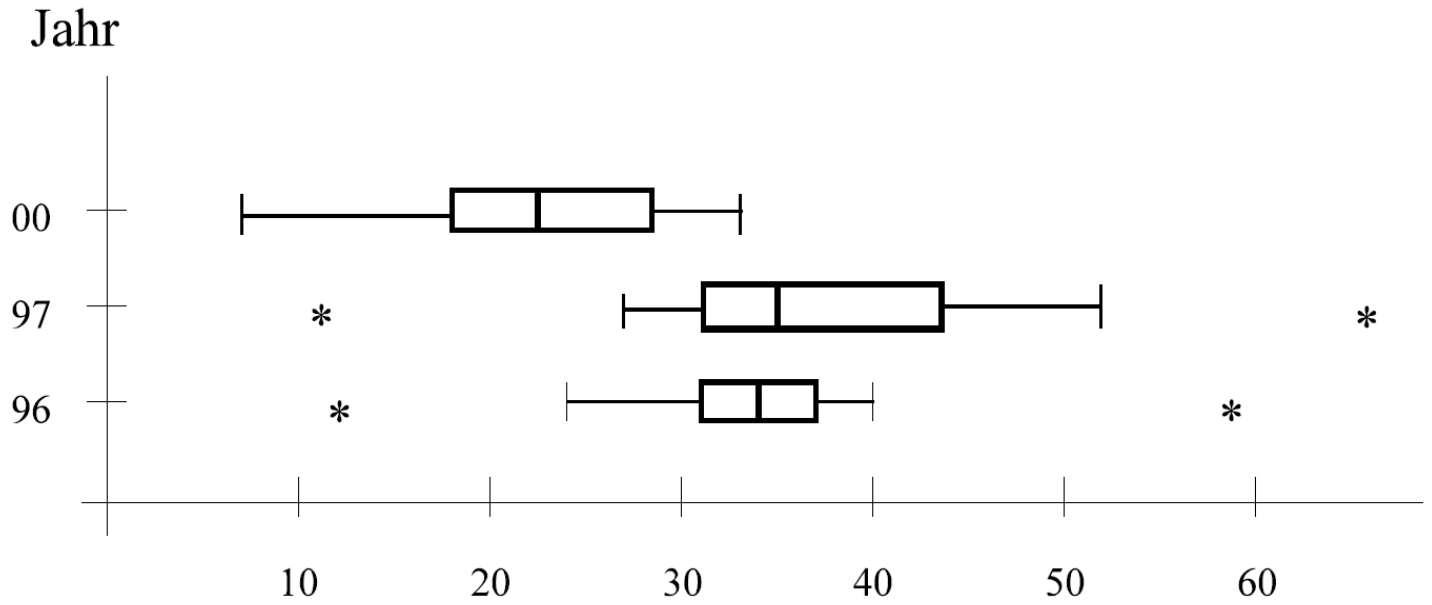
Beschreibende Statistik
BOX-PLOT





Beschreibende Statistik

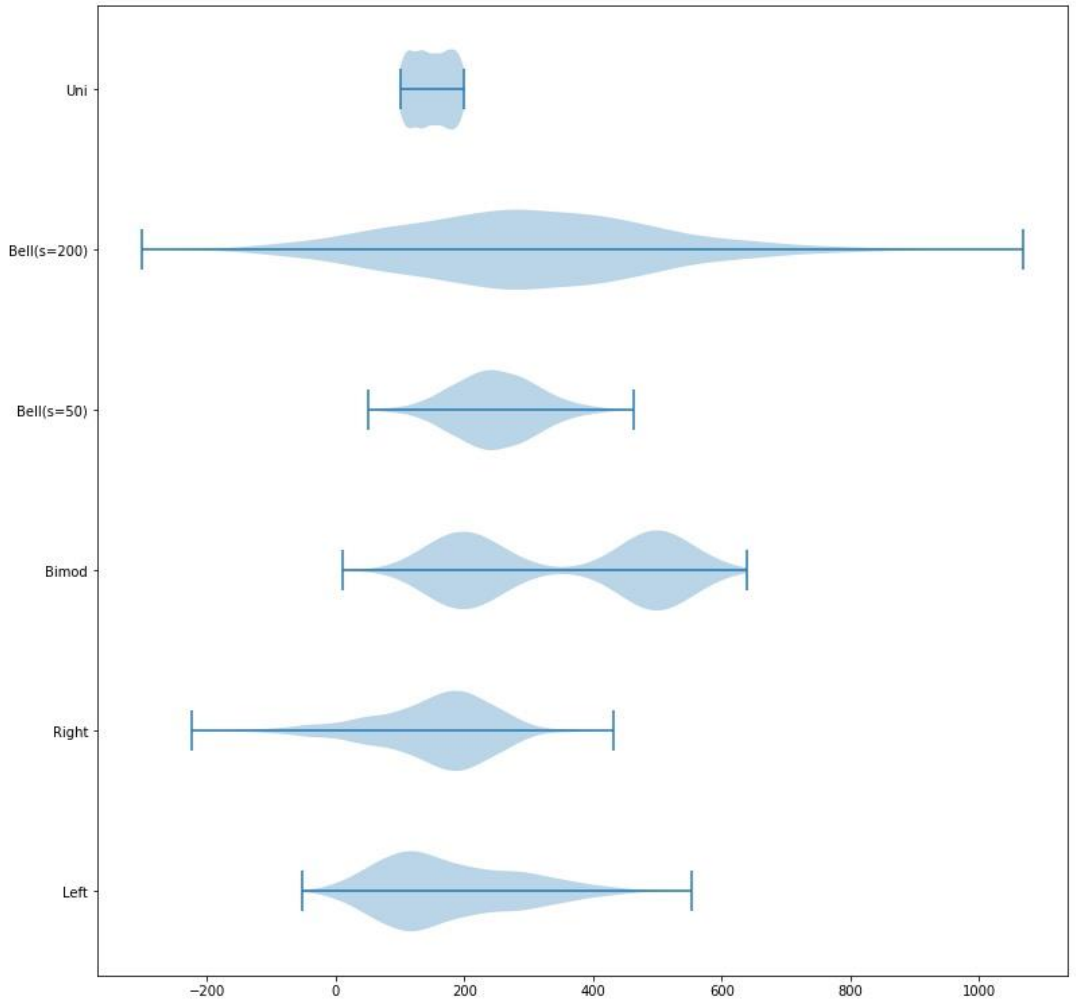
Kennzahlen für ein Merkmal - Mehrere BOX-PLOT



Schlussfolgerung:

- Vergrößerung der Varianz im Jahre 1997, aber sonst keine Veränderung
- Signifikante Änderung (Verringerung) der Werte im Jahre 2000

Beschreibende Statistik
VIOLIN-PLOT



Beschreibende Statistik

Kennzahlen für zwei Merkmal - Ermittlung paarweiser Abhängigkeiten

- mehrere Merkmale an den Objekten => empirische Abhängigkeiten zwischen den Merkmalen
- zweidimensionale Messreihe aus n Wertepaaren $(x_i, y_i)(i = 1, \dots, n)$
  (zwei beliebige Spalten einer Datenmatrix $\underline{\boldsymbol{X}}$)
- **empirische** Maßzahlen für die Stärke und Richtung des **linearen** Zusammenhangs zwischen den Merkmalen angegeben werden:
  - Kovarianz
  - Korrelationskoeffizient
  - Bestimmtheitsmaß

Beschreibende Statistik

Kennzahlen für zwei Merkmal - Ermittlung paarweiser Abhängigkeiten

Kovarianz:

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$\bar{x}$, $\bar{y}$ sind die Mittelwerte und s_x^2, s_y^2 die Streuungen von $\underline{x} = (x_1, \dots, x_n)^T$ und $\underline{y} = (y_1, \dots, y_n)^T$

Korrelationskoeffizient:

$$r_{xy} = \frac{s_{xy}}{s_x s_y} = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Bestimmtheitsmaß:

$$B_{xy} = r_{xy}^2$$

$s_{xy} > 0$ -> gleichsinniger Zusammenhang

$s_{xy} < 0$ -> gegensinniger Zusammenhang

$-1 \leq r_{xy} \leq +1$ mit $r_{xy} = 0$ ->x,y unkorreliert und

$|r_{xy}| \to 1$ -> x,y hoch korreliert (stark linear abhängig)

für standardisierte Merkmale $(m = 0, s^2 = 1)$ ist $r_{xy} = s_{xy}$

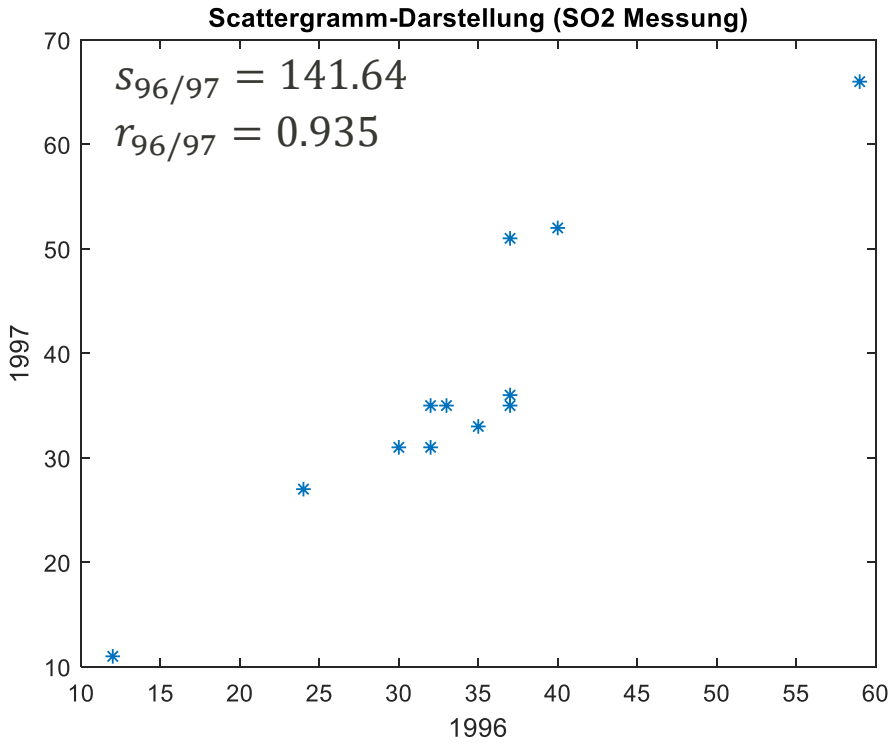
=> Korrelationsaussage ist keine Ursache-Wirkungsaussage (https://www.tylervigen.com/spurious-correlations)

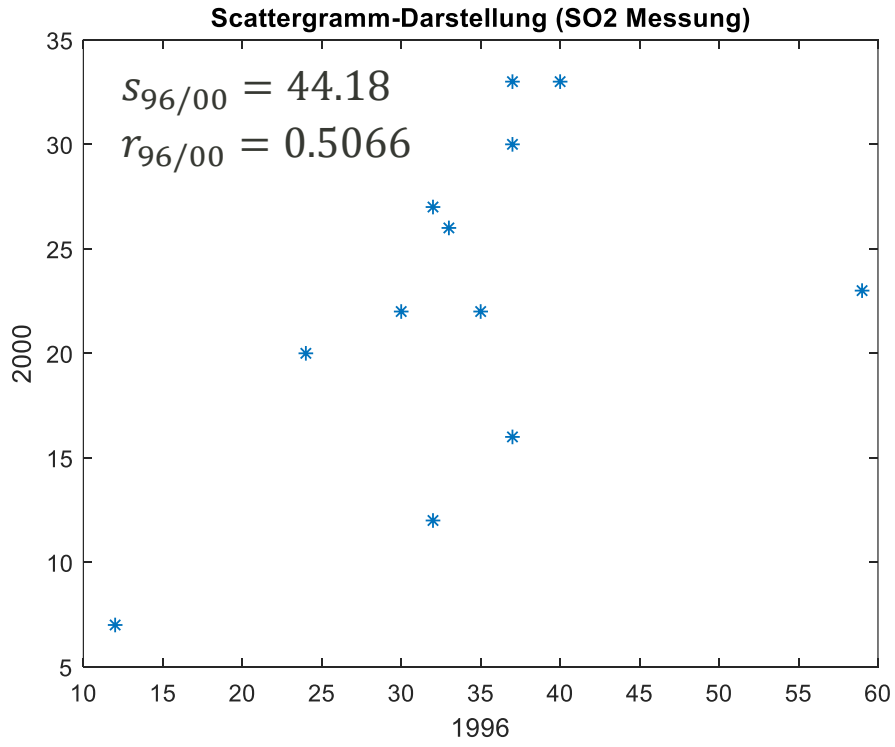
Beschreibende Statistik

Kennzahlen für zwei Merkmal - Graphische Darstellung Scatter-Plot

Korrelationskoeffizienten und Scattergramme für SO2 Messung

1996: $\bar{x}_{96} = 34.00, \quad s_{96}^2 = 118,$ **1997**: $\bar{x}_{97} = 36.92, \quad s_{96}^2 = 194.45,$ **2000**: $\bar{x}_{00} = 22.58, \quad s_{96}^2 = 64.45$





Beschreibende Statistik

Kennzahlen für mehr als zwei Merkmal - Kovarianzmatrix & Korrelationsmatrix

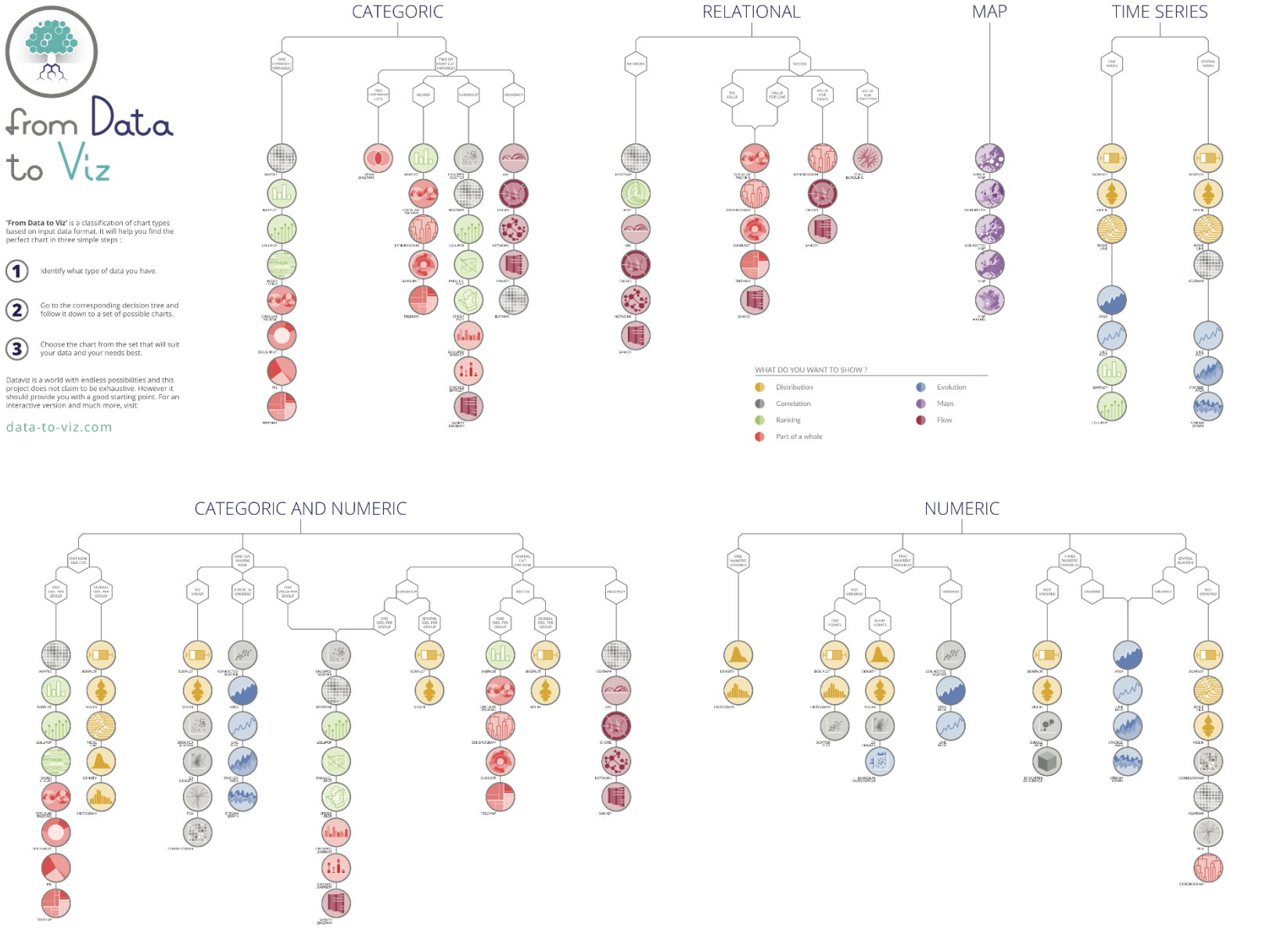
- Datenmatrix $\underline{\boldsymbol{X}}$ mit Messwertvektoren $\underline{\boldsymbol{x_1}}, \ldots, \underline{\boldsymbol{x_m}}$ für m gemessene Merkmale
- Zusammenfassung der paarweisen empirischen Kovarianzen bzw. Korrelationskoeffizienten zur **empirischen Kovarianzmatrix $\underline{\boldsymbol{S}}$** bzw. der **empirischen Korrelationsmatrix $\underline{\boldsymbol{R}}$**

$$\underline{\boldsymbol{S}} = \begin{pmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mm} \end{pmatrix} \qquad \underline{\boldsymbol{R}} = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}$$

mit $s_{ij} = s_{x_i x_j}$ und $r_{ij} = r_{x_i x_j}$

Mit $\underline{\boldsymbol{X}}^{norm}$ (die aus $\underline{\boldsymbol{X}}$ durch Standardisierung aller Merkmale hervorgegangene Matrix) gilt:

$$\underline{\boldsymbol{R}} = \frac{1}{n-1} \underline{\boldsymbol{X}}^{norm^T} \underline{\boldsymbol{X}}^{norm}$$

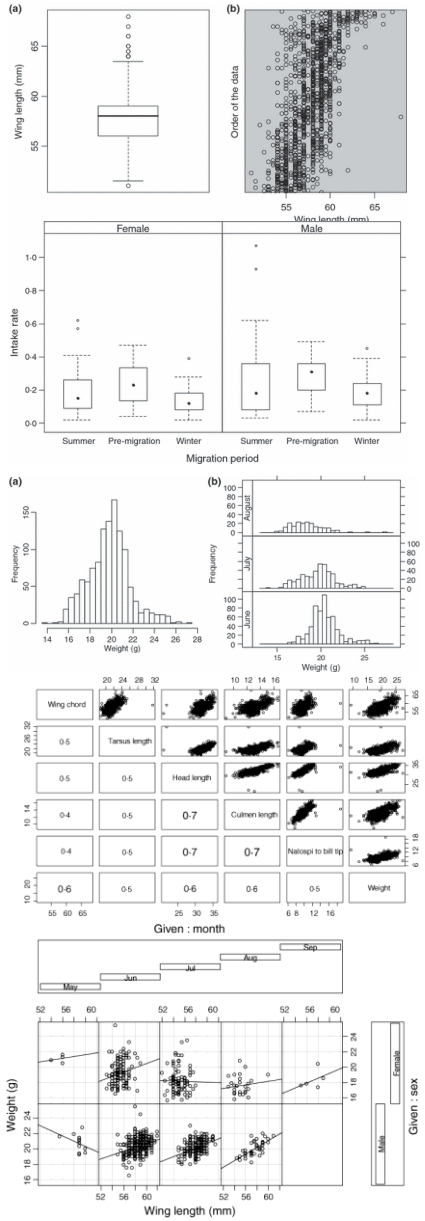

from Data to Viz
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps :
1 Identify what type of data you have.
2 Go to the corresponding decision tree and follow it down to a set of possible charts.
3 Choose the chart from the set that will suit your data and your needs best.
Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:
data-to-viz.com
CATEGORIC
RELATIONAL
MAP
TIME SERIES
WHAT DO YOU WANT TO SHOW ?
Distribution
Correlation
Ranking
Part of a whole
Evolution
Maps
Flow
CATEGORIC AND NUMERIC
NUMERIC

Protokoll für Datenexploration (Zuur, 2010)

1. Ermittlung von Ausreißern in X und Y: Box-Plot, Cleveland Dotplot, Multivariate Analysis
2. Annahme der Varianzhomogenität Y: Box-Plot für Gruppen
3. Normalverteilung von Y: Histogramm, Q-Q-Plot
4. Anteil von Nullen in Y: Frequenzhistogramm
5. Multikolinearität von X: VIF, Corellogram, PCA
6. Zusammenhänge zwischen X und Y: Scatterplots, Box-Plot für Gruppen
7. Interaktionseffekt: Coplot
8. Unabhängigkeit von Y: ACF, Variogramm, Plot Y vs. Zeit/Ort

- Bei Anwendung von Modellen sollen Annahmen recherchiert werden.
- Die Reihenfolge lässt sich je nach Datensatz variieren.

Tools: Jupyter Notebook, Matplotlib, Plotly, Seaborn; Matlab Livescript



A. F. Zuur, E. N. Ieno, and C. S. Elphick, "A protocol for data exploration to avoid common statistical problems," *Methods Ecol. Evol.*, vol. 1, no. 1, pp. 3–14, Mar. 2010, doi: 10.1111/j.2041-210X.2009.00001.x.

Schritt 3: Data cleaning

Typische Probleme industrieller Daten

Allgemeines Vorgehen:

1. Identifikation möglicher Probleme
2. Suche nach Instanzen von Problemen
3. Korrektur von Fehlern
4. Dokumentierung von Fehlerpunkten
5. Anpassung der Routine zur Datenexploration

Typische Probleme:

- **Fehlende Werte**
  - Ausgefallen Geräte, Unterbrochene Verbindung
  - ➢ Datenimputation
- **Ausreißer**
  - Falsche Einstellung, zufälliger Sensorausfall
  - ➢ Korrektur/Entfernung
- **Datendrift**
  - Systematische Sensordegradation
  - ➢ Drift-Identifikation und Korrektur
- **Multikolinearität**
  - Redundanz, Regelkreise
  - ➢ Manuelle Feature-Selection, PCA, PLS
- **Abtastrate und Verzögerungen**
  - ➢ Multi-rate-Modelle, Resampling

Schritt 4: Data labeling

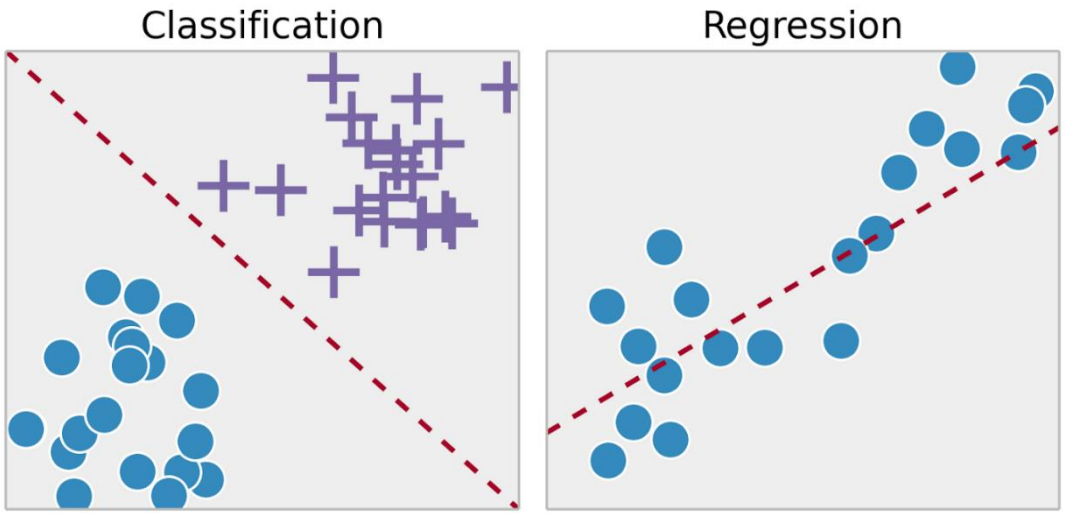
Labels

Labels sind im Trainingsdatensatz erhaltene **Zielvariablen**, die dem Trainieren des Modells dienen.

Relevant nur für **Überwachtes Lernen**.

Labelling ist ein Verfahren in dem Labels für Datensatz **manuell**, **semi-automatisch** oder **automatisch** erhoben werden.

Labelling setzt **Wissensexpertise** oder **automatische Gewinnung** von bereits annotierten Daten voraus.

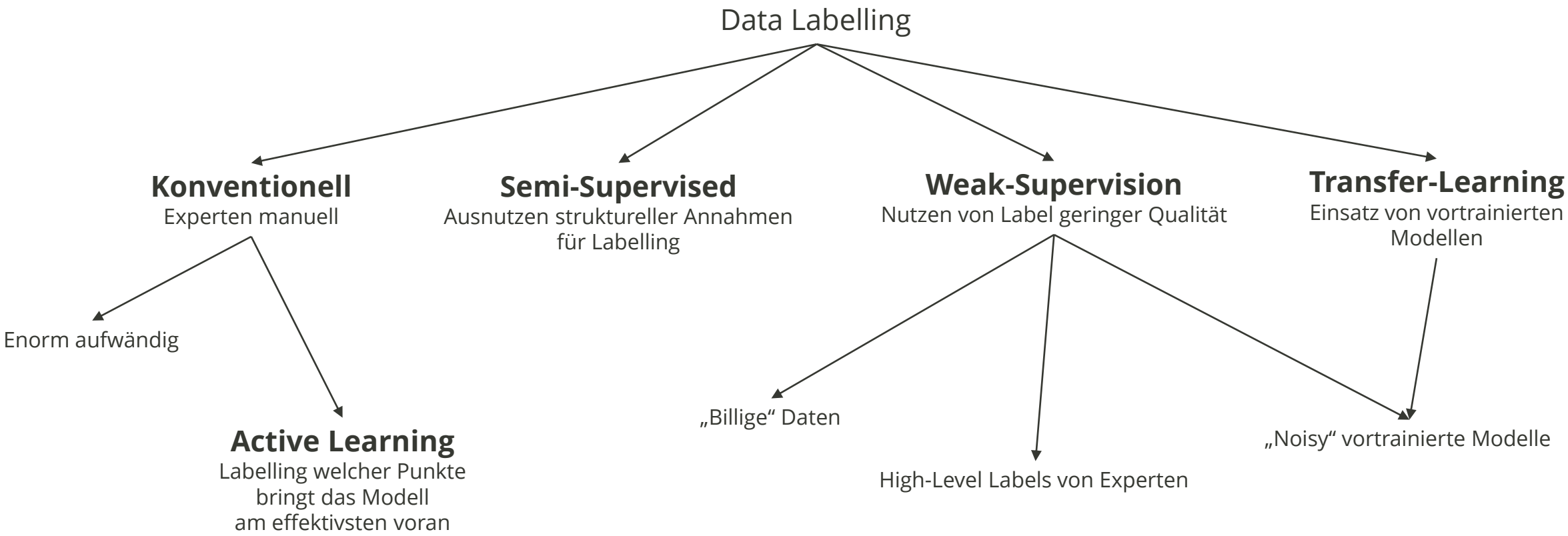


Labels: Klassen

Labels: Werte der Zielvariablen

Bild: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Ansätze für Labelling



http://ai.stanford.edu/blog/weak-supervision/

Semi-supervised learning

Motivation:

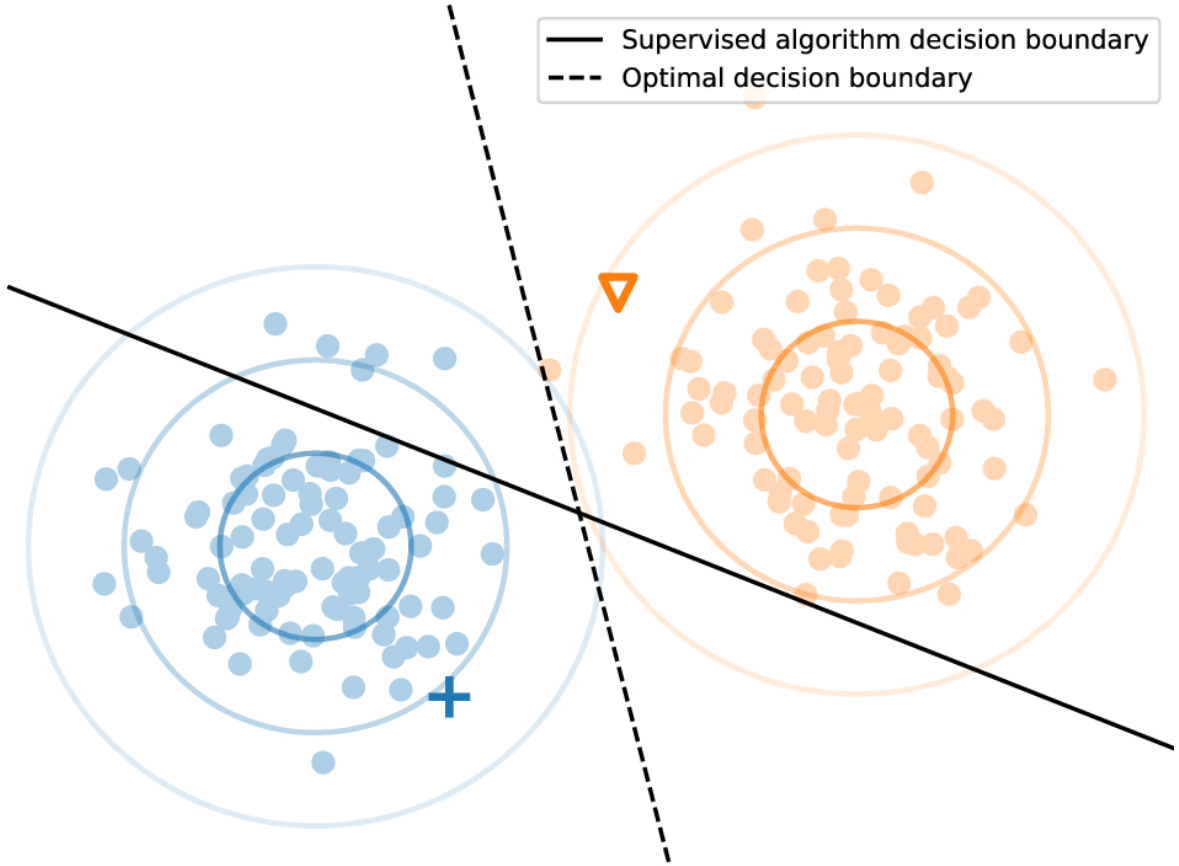
- Gewinnung von ungelabelten Daten ist mehrfach einfacher
- Labelling ist aufwändig

Grundidee:

- Nicht annotierte Daten liefern weitere nützliche Information für das Trainieren des Modells

Annahmen:

- **Glattheit**: wenn x_1 und x_2 nahe stehen, dann y_1 und y_2 sollen auch
- Entscheidungsgrenze liegt in der Region mit geringer Dichte
- Manifold-Annahme



J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020, doi: 10.1007/s10994-019-05855-6.

Schritt 5: Feature engineering

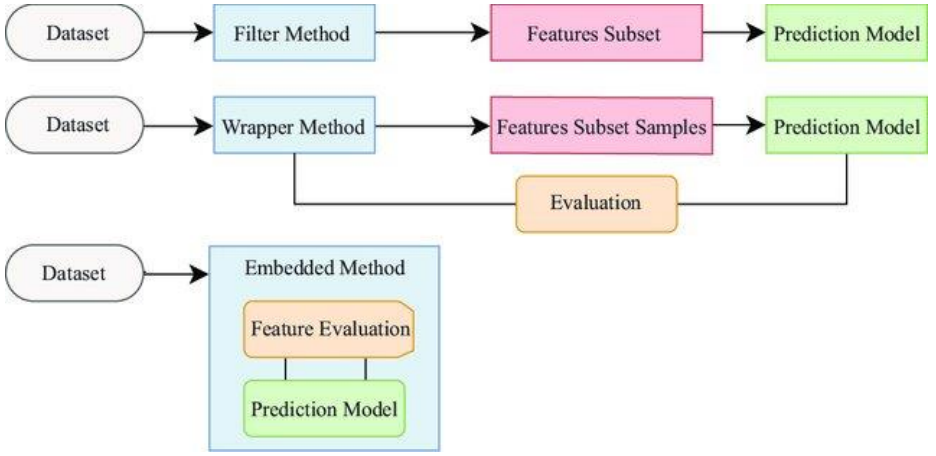
Feature Engineering

Feature-Engineering ist ein Schritt im Rahmen der Vorbereitung eines Trainingsdatensatzes aus Rohdaten, in dem Eingangsvariablen **extrahiert**, **ausgewählt** und **transformiert** werden.

Mehrwert von effektiv ausgewählten Features ist ein effektiveres Modell:

- Geringe Modellkomplexität
- Höhere Generalisierbarkeit und Reduzierung des Overfitting-Effekts
- Bessere Erklärbarkeit
- Schnelles Modelltraining

Feature Selection



M. BABIKER, et. al. Doi: 10.3906/elk-1812-18

Ansätze

Konventionell
Experten manuell
Iterativ und aufwändig

Supervised
Anwendung eines daten-basierten Ansatzes

Filter
Auswahl von Features basiert auf **statistischen Metriken**:
- Varianz-Threshold
- Information-Gain
- Chi-Square-Test

Modellunabhängig

Wrapper
Numerische Optimierung, wo **das Zielkriterium dem Modellgüte** entspricht:
- Forward selection
- Backward elimination
- Stepwise selection

Overfitting-Gefahr, da Modell-spezifisch; Zeitaufwändig

Embedded
Feature-Selection ist **ein Anteil des Modells**:
- Regularization z.B. L1, L2
- Feature-Importance

Modellunabhängig

http://ai.stanford.edu/blog/weak-supervision/

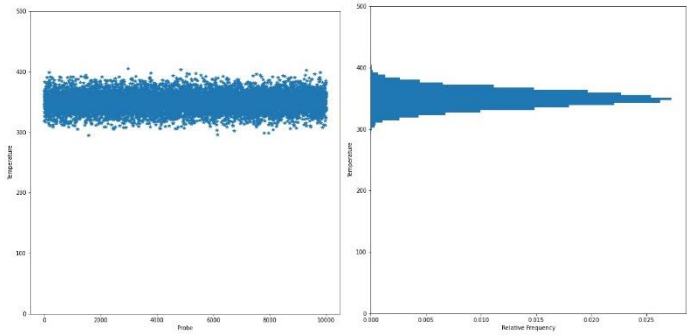
Konventionelle Feature Selection

Mögliche Merkmale:

- Minimum
- Maximum
- Mittelwert
- Median
- Modus
- Varianz
- Schiefe
- Spannweite
- Interquartilsabstand
- ...

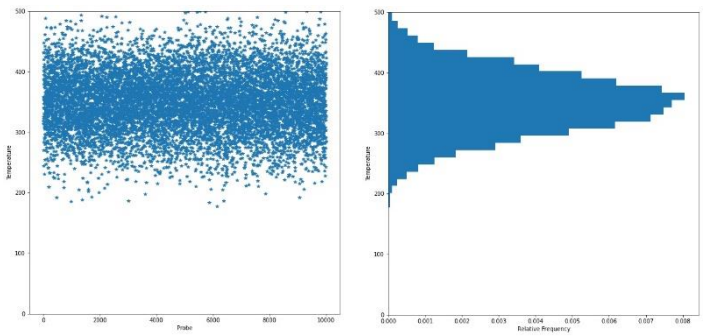
Datenanalyse + Fachexpertise

Normaler Betrieb



Merkmal	Wert
Min	299,05
Max	413,07
Mittelwert	349,74
Varianz	222,63
Schiefe	-0,004
...	

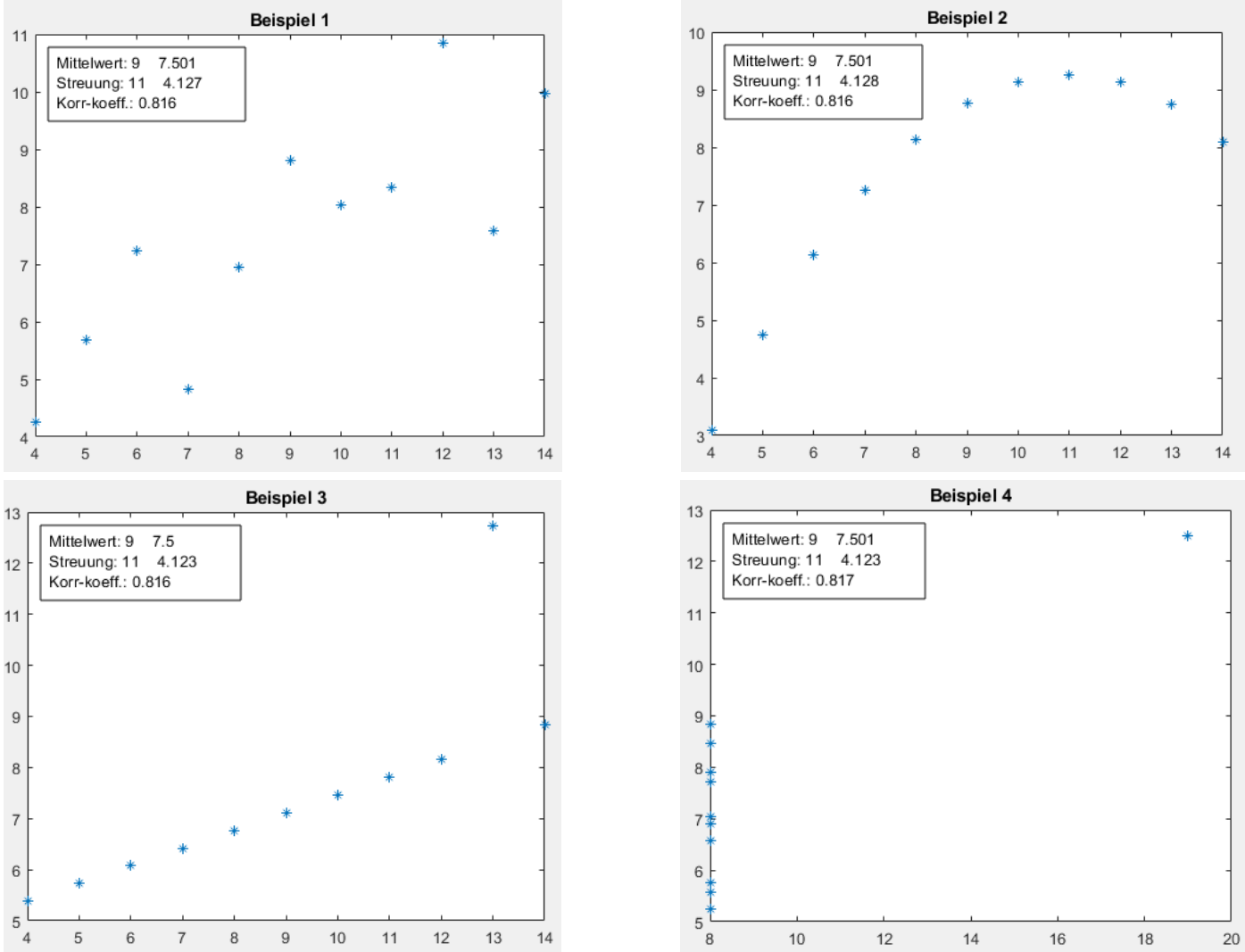
Fehlbetrieb



Merkmal	Wert
Min	177,21
Max	532,63
Mittelwert	350,48
Varianz	2460,57
Schiefe	-0,03
...	

Konventionelle Feature Selection

Beispiel: Das Anscombe-Quartett[1]



Feature Transformation

Feature Transformation - zielgerichtete Wandlung von Daten **ohne Verlust von Information**.

Mögliche Ansätze: Logarithmieren, Multiplikation von Inputvariablen und Darstellung als einzelnes Feature, Differenzieren, Wechsel des Koordinatensystems

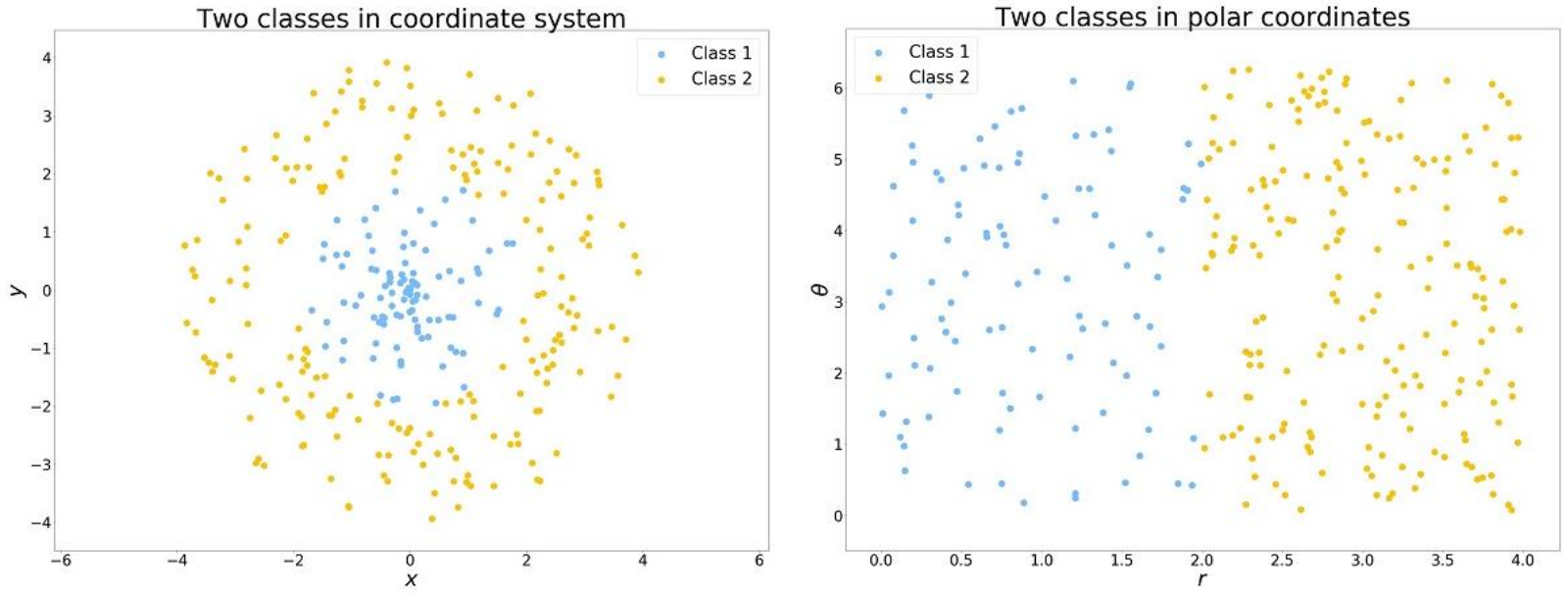


Bild: https://www.kdnuggets.com/2018/12/feature-engineering-explained.html

Feature Scaling

Standard-Scaler (=Standardisation):

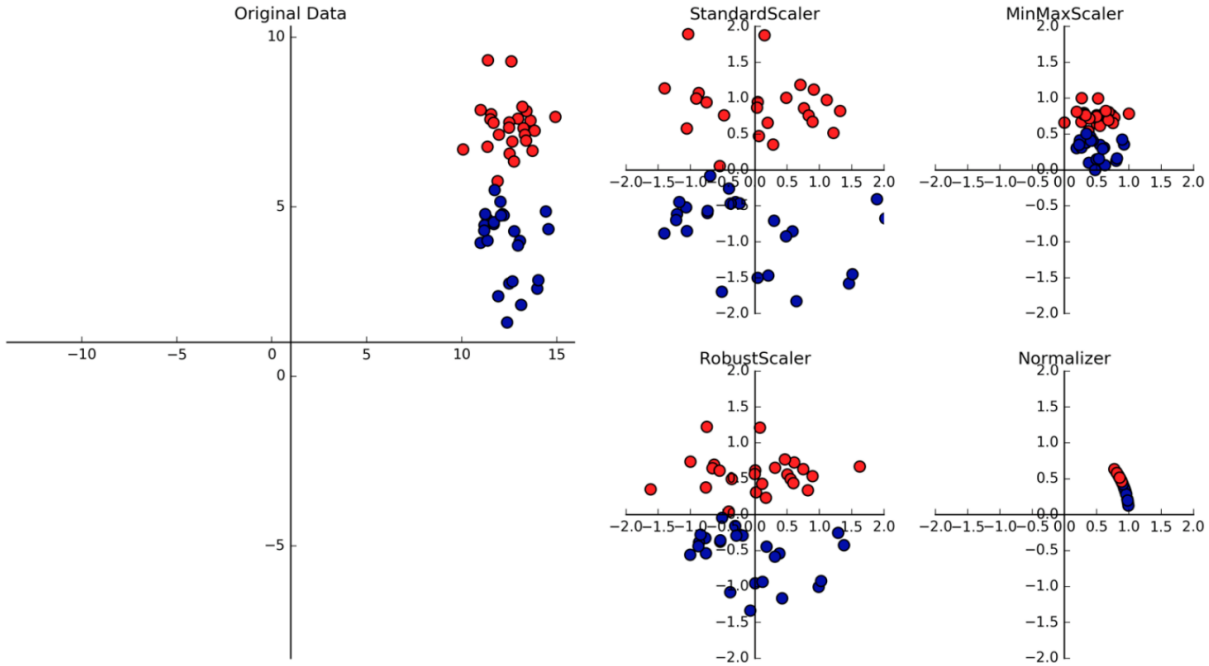
$$z = \frac{x - \bar{x}}{\sigma}$$

MinMax-Scaler (=Normalisation):

$$x_{sc} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Robust-Scaler:

$$x_{sc} = \frac{x - m_e}{\tilde{x}_{0,75} - \tilde{x}_{0,50}}$$

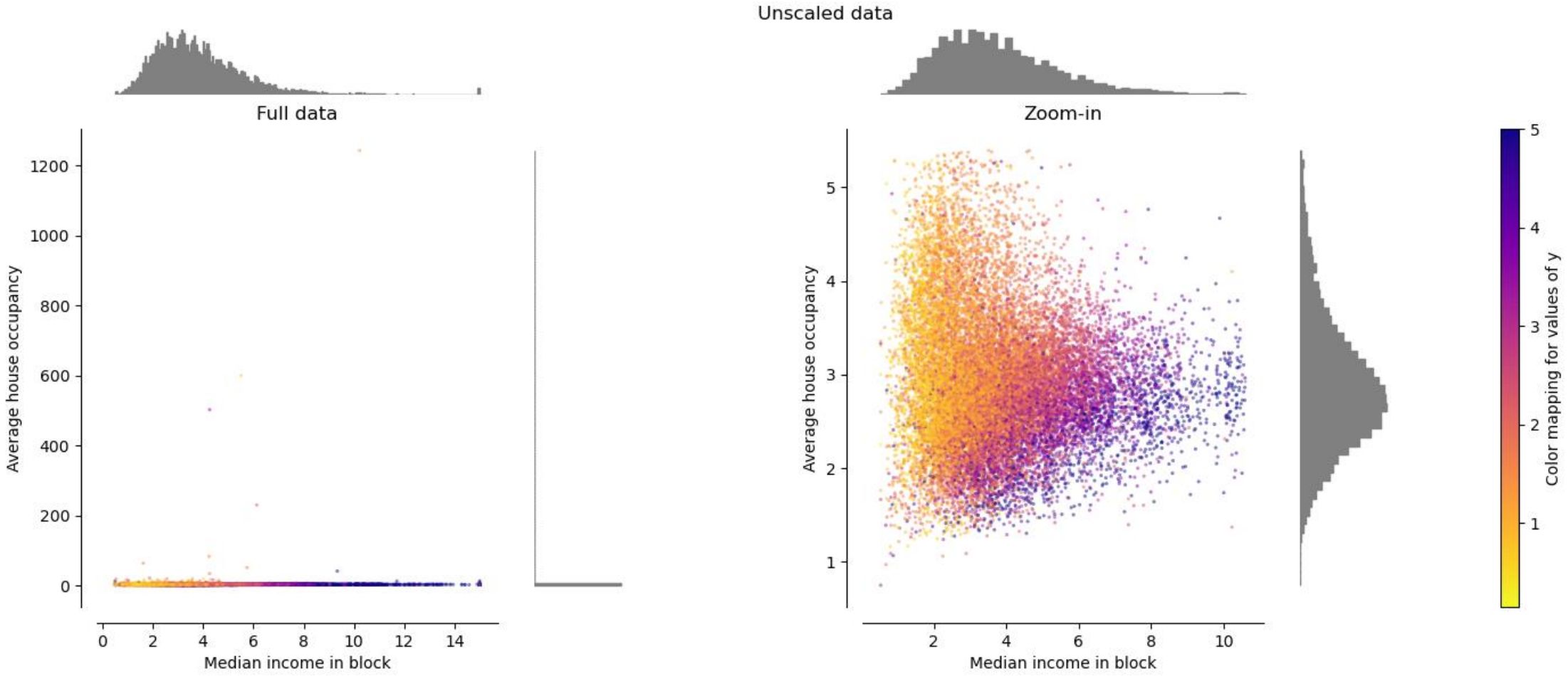


https://python-data-science.readthedocs.io/en/latest/_images/scaling.png

Ergebnis:

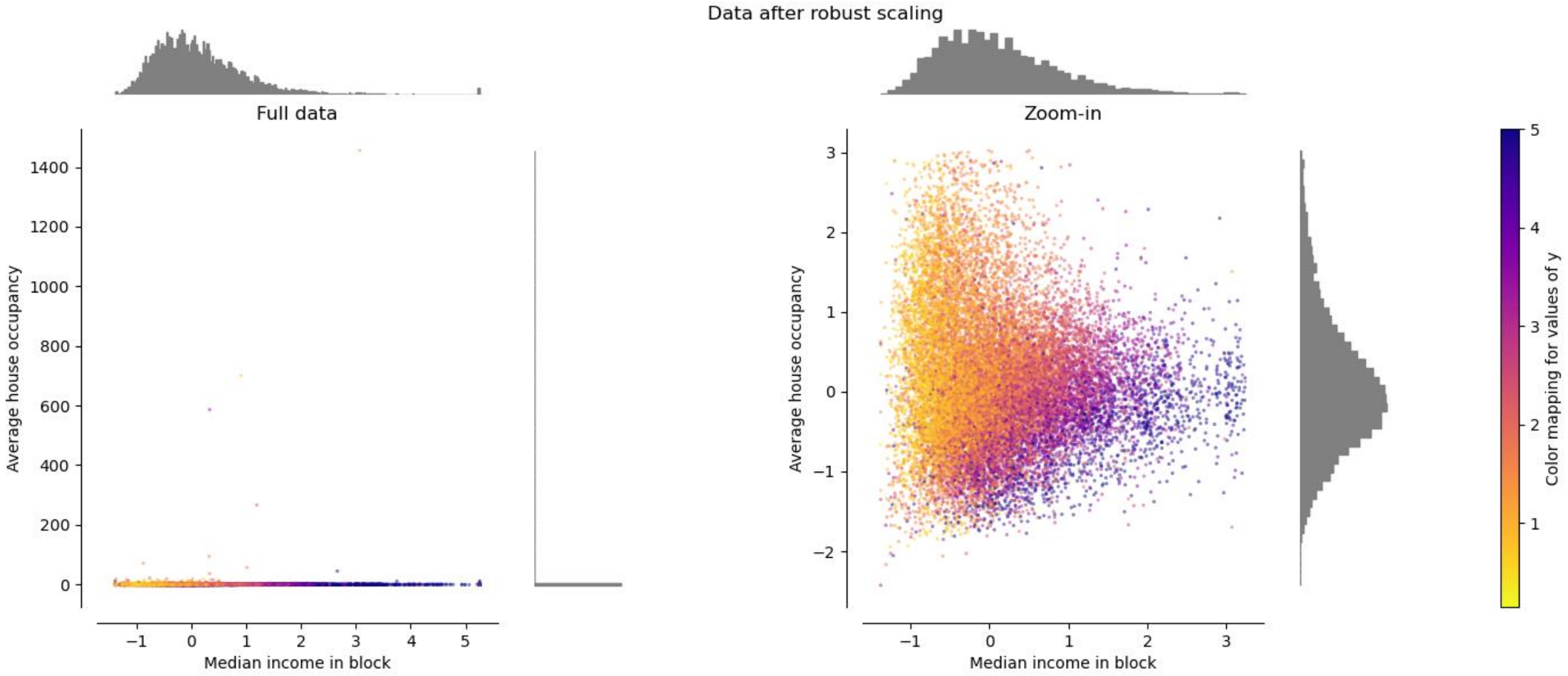
- Gewicht von einzelnen Features ist ausgeglichen
- Ausreißer werden wenig ausgeprägt

Feature Transformation



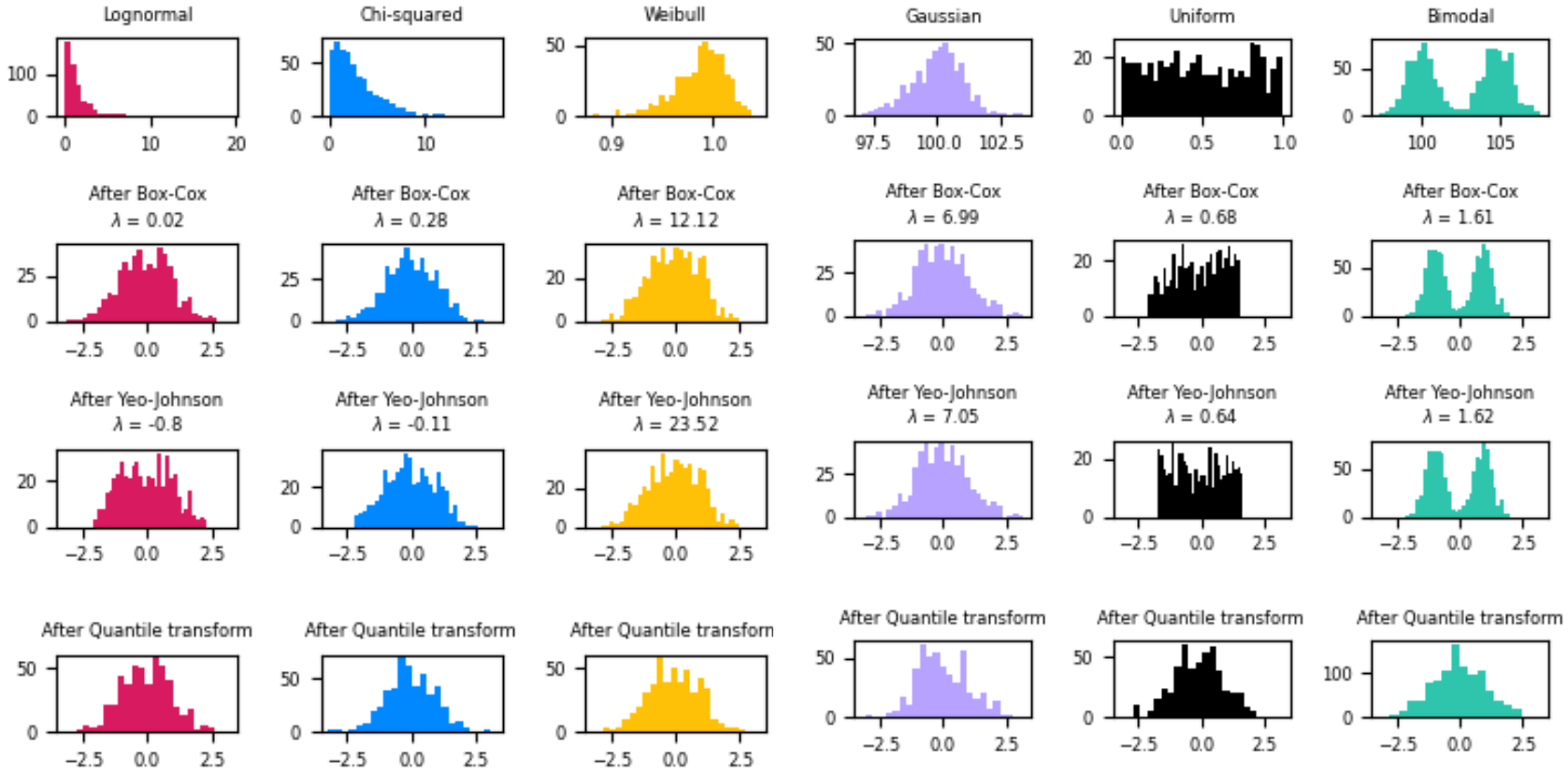
https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

Feature Transformation



https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

Feature Transformation

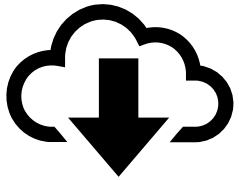


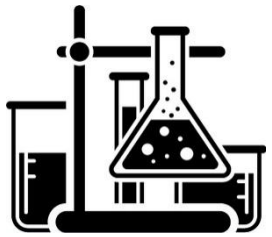
https://scikit-learn.org/stable/modules/preprocessing.html

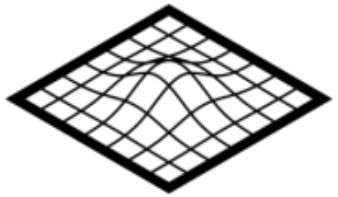
Zusammenfassung

Zusammenfassung

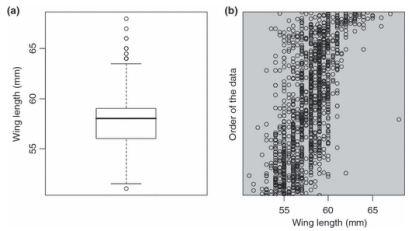
Datenbeschaffung

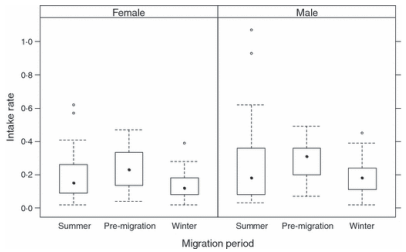


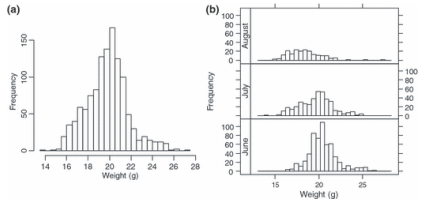


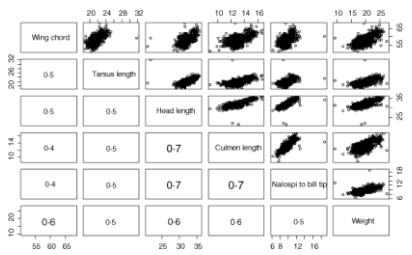


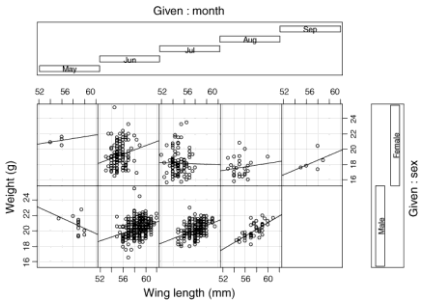
Datenexploration









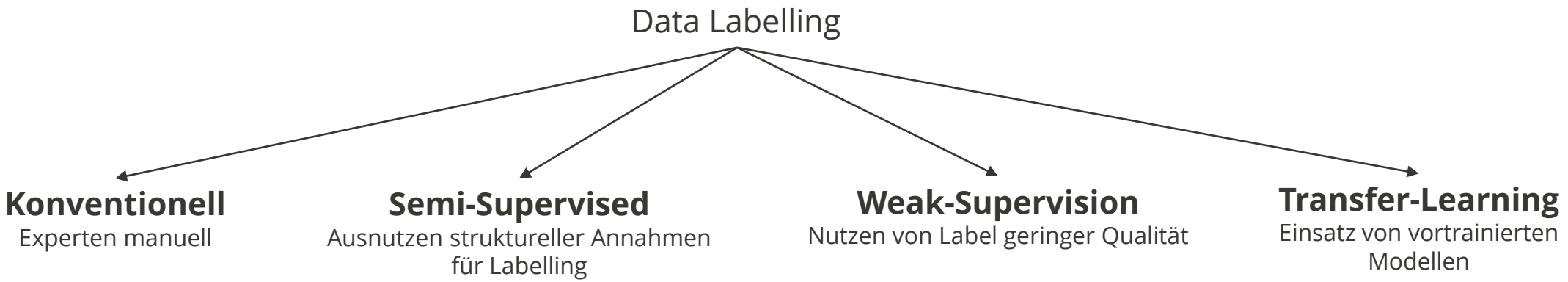


Zusammenfassung

Cleaning

Fehlende Werte, Ausreißer, Datendrift, Multikolinearität, Abtastrate und Verzögerungen

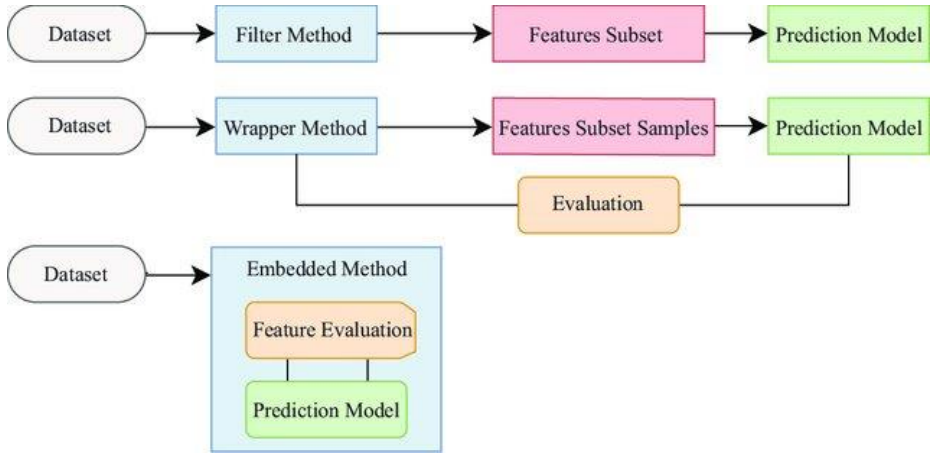
Labeling



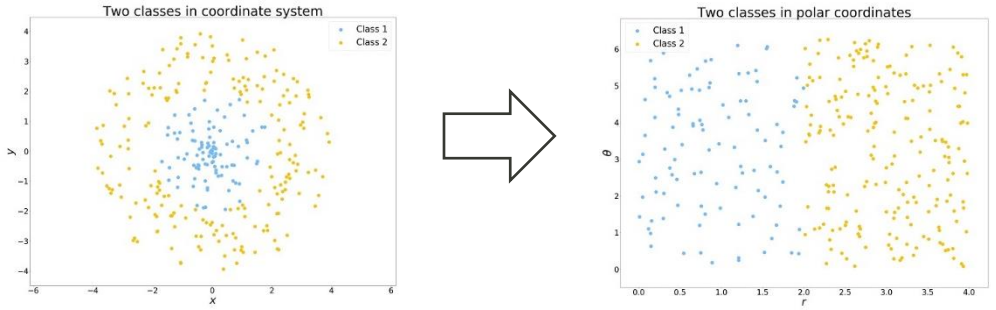
Zusammenfassung

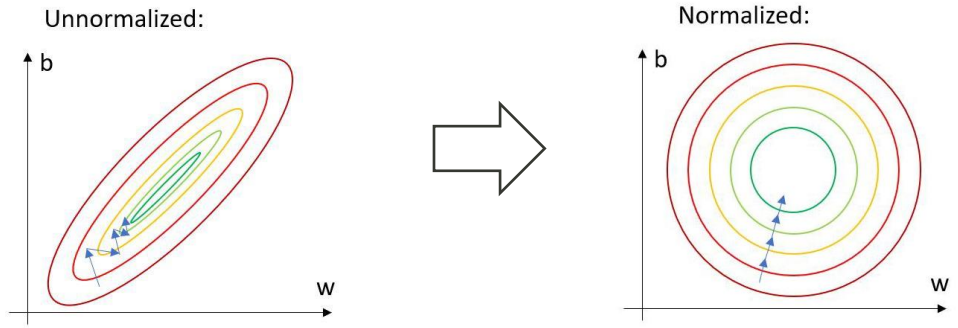
Feature Engineering

Selection



Transformation und Normalization

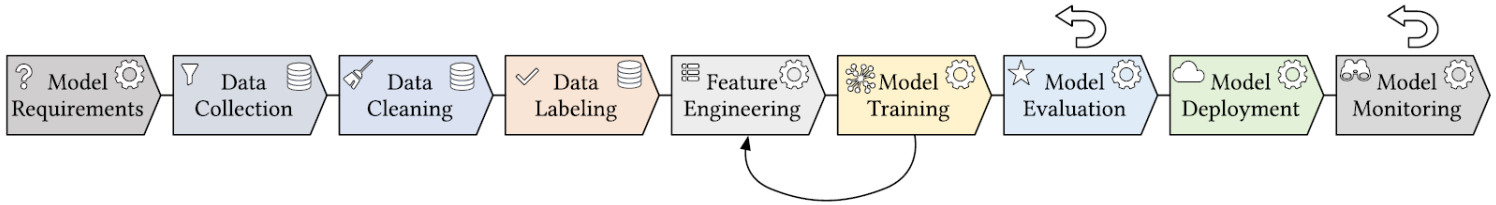




https://towardsdatascience.com/how-to-calculate-the-mean-and-standard-deviation-normalizing-datasets-in-pytorch-704bd7d05f4c

Zusammenfassung

Vorgehensmodell nach Amershi:



- Data collection – Bereitstellung des Datensatzes: Datenimport oder –beschaffung
- Data cleaning – Aufbereitung des Datensatzes
- Data labeling – Markierung von Daten (Überwachstes Lernen)
- Feature engineering – Auswahl von Features und deren Aufbereitung für das Training
- Model training – Trainieren, Optimierung von Modell- und Training-Hyperparametern **Fokus der nächsten Vorlesung**
- Model evaluation – Testen des Modells mit einem Test-Datensatz, Berechnung von Metriken, Auswahl eines Modells für Einsatz in Produktion
- Model deployment – Aufbau Runtime-Umgebung für Modell-Inference, Einsetzen des Modells
- Model monitoring – Evalution des Modells im Betrieb, Sammlung Daten für Verbesserung
- Model maintenance – Aktualisierung des Modells (z.B. nach Erweiterung/Anpassung des Training-Datensatzes)

S. Amershi *et al.*, "Software Engineering for Machine Learning: A Case Study," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

Dr. rer. nat. Valentin Khaydarov
Email: valentin.khaydarov@tu-dresden.de
Telefon: 0351 463 33387

Vielen Dank für Ihre Aufmerksamkeit!