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

Einführung, Grundbegriffe, End-to-End Workflows

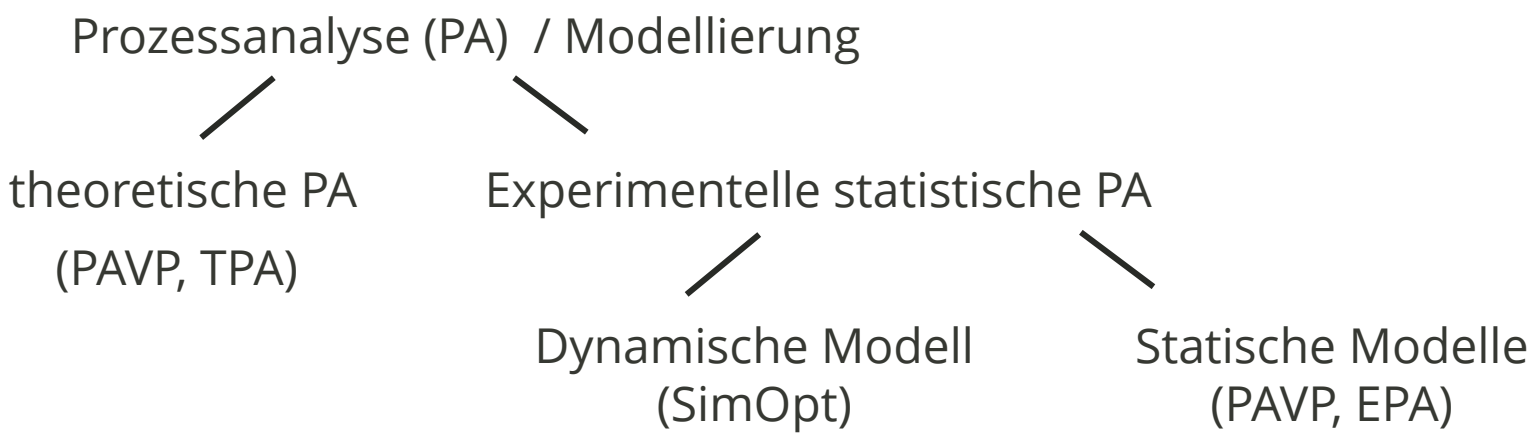
Vorlesung 1, Lehrveranstaltung Experimentelle Prozessanalyse

Gliederung

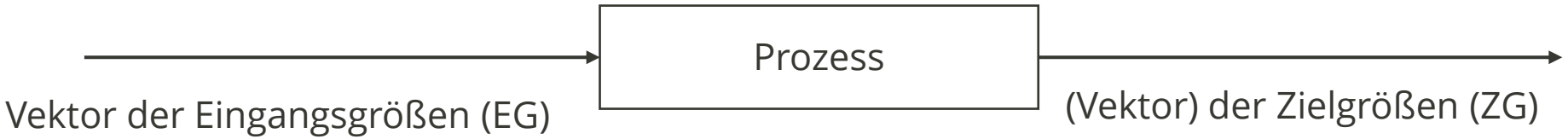
- Motivation, Grundlagen und Definition Machine Learning
- Aufgaben für Machine Learning
- Vorgehensweise
- (Exemplarische Modellierungsansätze)
- Zusammenfassung
- (Beispiel)

Motivation, Grundlagen und Definition

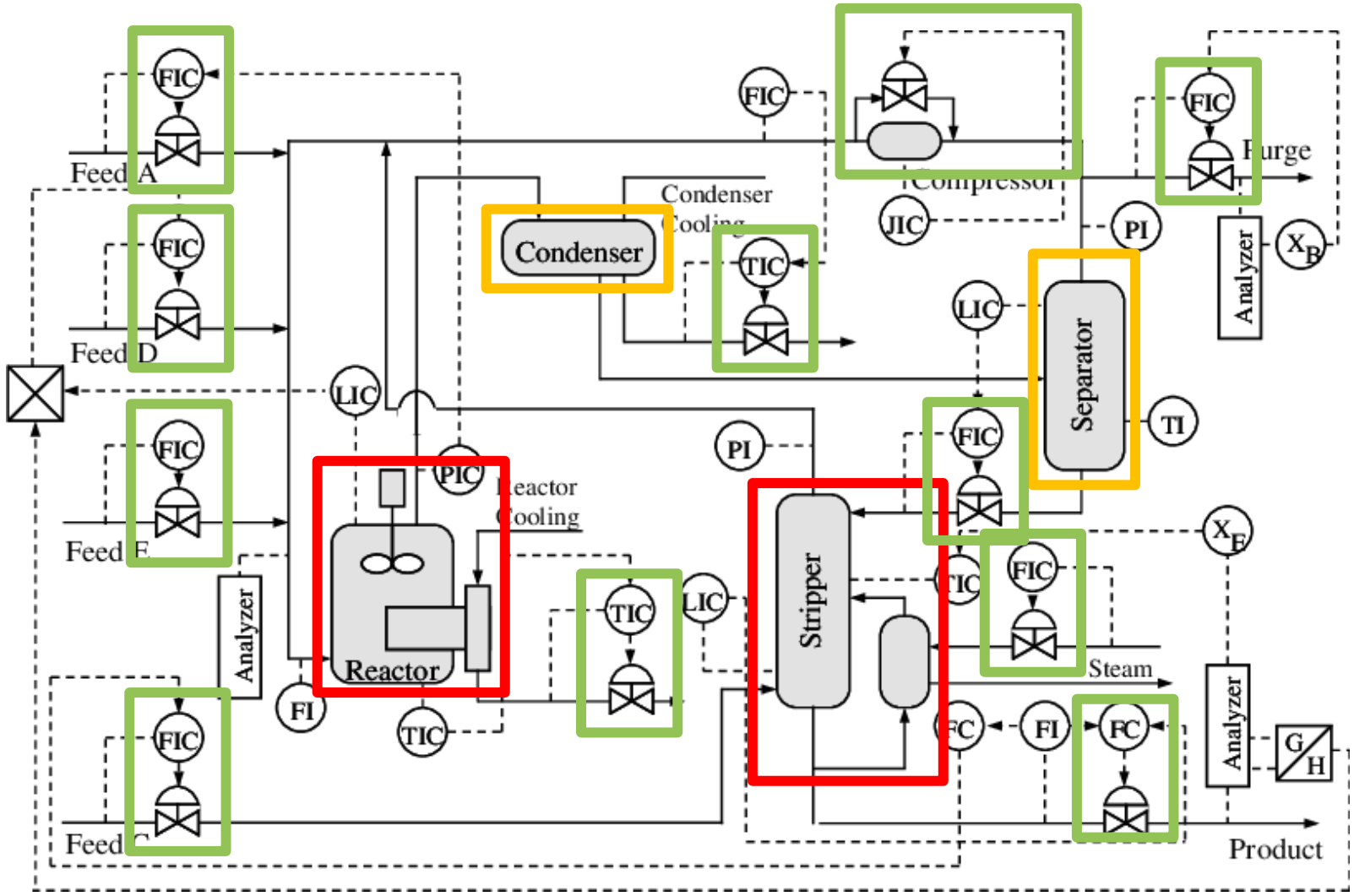
Einführung in die Prozessanalyse



Einführung in die Prozessanalyse



Tennesse-Eastman Verfahren



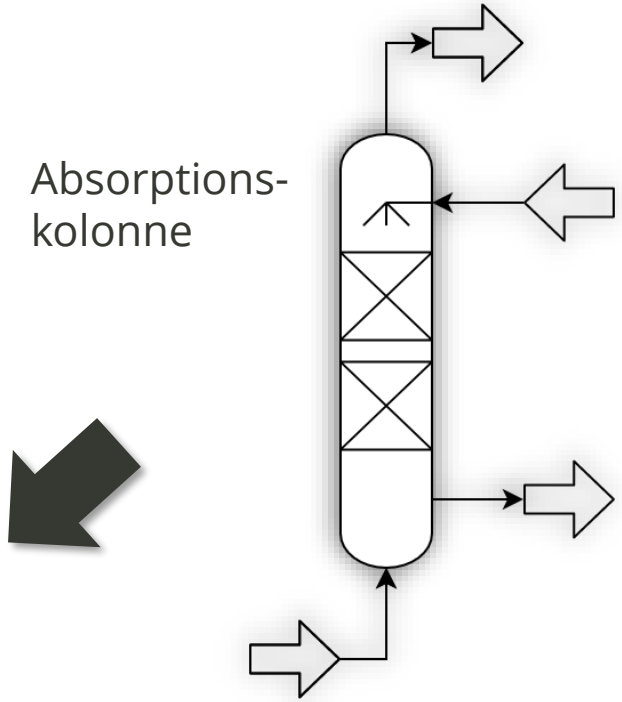
Monitoring

Optimierung

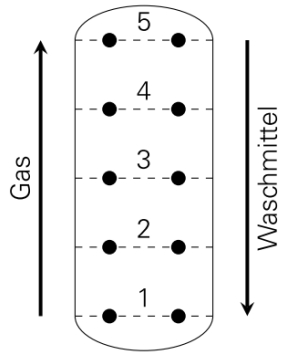
Kano u. a., „Contribution Plots for Fault Identification Based on the Dissimilarity of Process Data“.

Rigorose und datengetriebene Modellierungsansätze

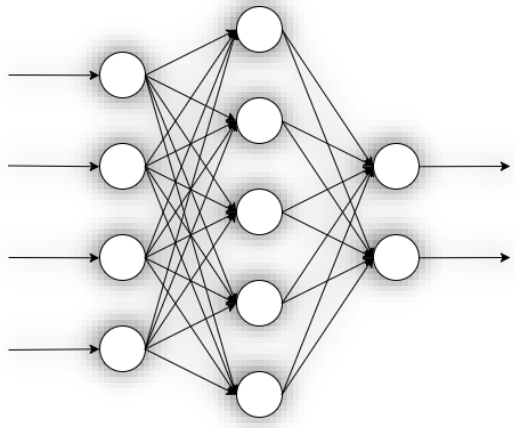
Absorptions-
kolonne



Rigorose Modellierung:
First-Principle, White-Box, theoretisch, wissensbasiert



Datengetriebene Modellierung:
Machine Learning, Black-Box, experimentell, datenbasiert, empirisch



Hybride Modellierung oder Gray-Box

Vergleich der rigorosen und datengetriebenen Modellierung

Folgende Kriterien sprechen für die Wahl des **rigorosen Modellierungsansatzes**:

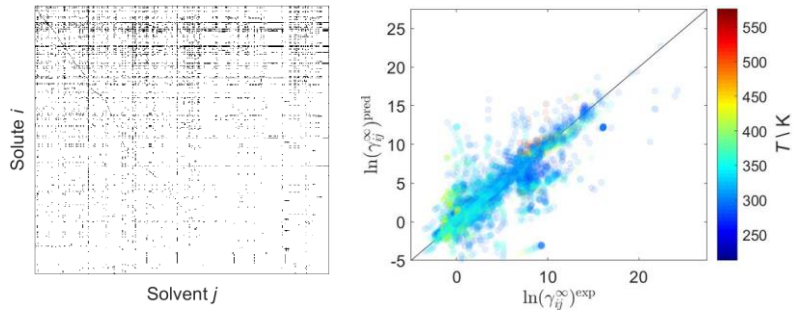
- Modellierungsziel: Verbesserung der Kenntnisse über die Elementarprozesse und deren Wechselwirkung
- Große Teile der relevanten elementaren Prozesse sind bereits sehr gut verstanden
- Eine begrenzte Menge qualitativ hochwertiger Daten vorhanden
- Die Durchführung von Experimenten ist teuer (z.B. weil die Anlage im regulären Produktionsbetrieb arbeitet oder die Ressourcen teuer sind)
- Es ist ein großes Budget verfügbar (Geld, Zeit, Arbeitskraft) und es soll ausreichend Zeit aufgewendet werden um den Prozess besser zu verstehen

Folgende Kriterien sprechen für die Wahl der **datengetriebene Modellierung**:

- Die relevanten elementaren Prozesse sind kaum verstanden
- Eine große Menge qualitativ hochwertiger Daten ist vorhanden
- Die Durchführung von Experimenten ist verhältnismäßig kostengünstig
- Es ist ein kleines Budget verfügbar (Geld, Zeit, Arbeitskraft) und es sollen schnell Verbesserungen erzielt werden

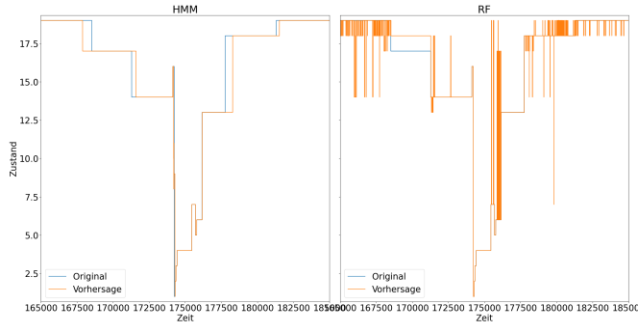
Beispiele in der Prozessindustrie

Prädiktion von Stoffeigenschaften in Gemischen



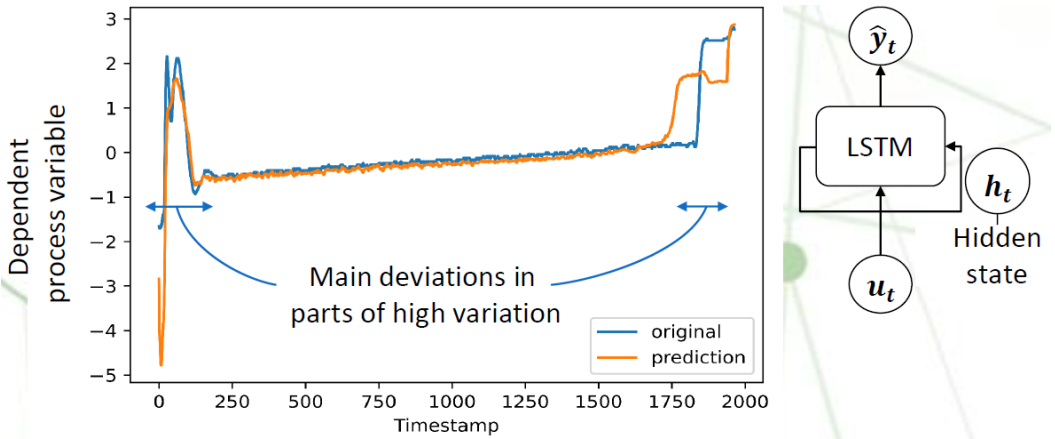
J. Damay, M. Bortz, H. Hasse (2021)
Predicting activity coefficients at infinite dilution with matrix completion

Identifikation von Batch-Phasen



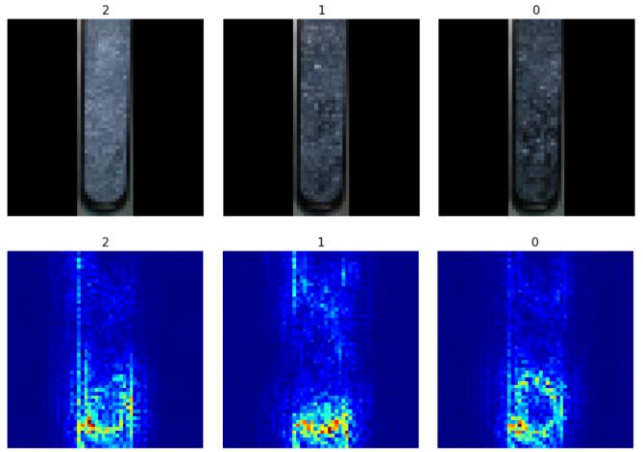
G. Just (2021)
Student Thesis: Classification of batch phases in the process industry using time series analysis (TU Dresden)

Surrogat-Modellierung dynamischer Prozesse



J. Winz, U. J. Ravali Theeda, B. Bordas, K. Kurt, A. Bamberg, S. Engell (2021)
Hybrid modeling and control of a batch distillation process of polymer solutions

Bilddaten-basierte Klassifikation von Strömungsregimen



C. Kröger (2021)
Diploma Thesis: AI based detection of flow regimes (TU Dresden)

Kanonische Definition Maschinelles Lernen

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

Tom Mitchell, 1997

Erforderliche Komponenten:

- **Erfahrung E** = Datensatz bzw. Grundlage für die Modellbildung

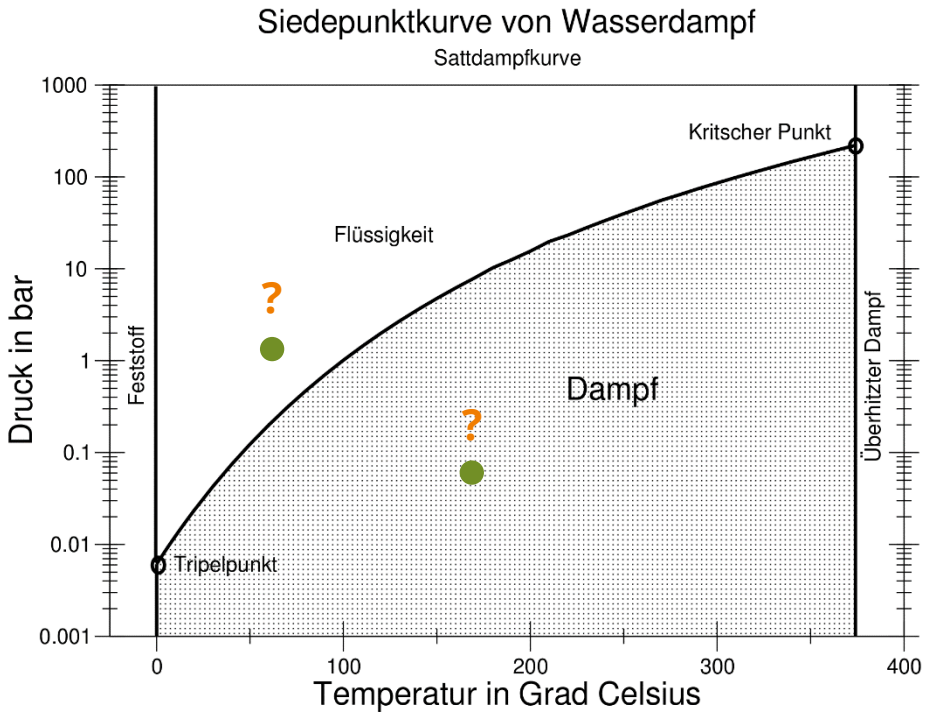
Datenpunkte in der Form: **Temperatur**, **Druck**, **Zustand**

- **Aufgabe T** = Modell bzw. Transformation von Input in Output

Erkennung des **Zustandes** bei vorgegebenen **Temperatur** und **Druck**: $Z = f(T, p)$

- **Leistung P** = Zielfunktion bzw. Optimierungsziel

Weniger falsch erkannte **Zustände**



https://de.wikipedia.org/wiki/Datei:Dampfdruckkurve.svg

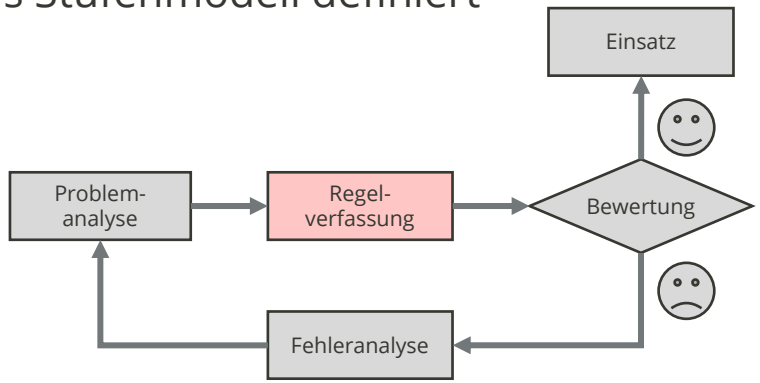
Weitere Definition

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

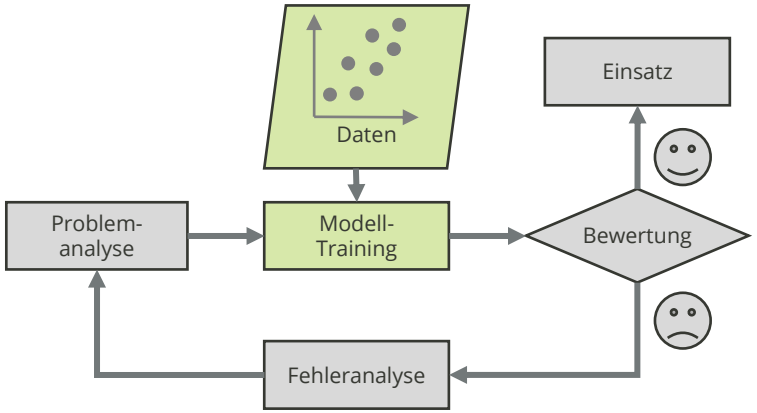
Traditionelle Programmierung – feste, vordefinierte Regel (Merkmale) z.B.:

- Gleichgewicht zw. Gas- und Flüssigkeitsphasen gemäß Henry-Gesetz
- Die Konzentrationsänderung von Stoffen ist durch das Stufenmodell definiert



Machine-Learning-Ansatz – Regel werden aus Daten abgeleitet z.B.:

- Initiale Konzentrationen, Temperatur- und Druckprofile als Input
- Konzentrationsprofile als Output



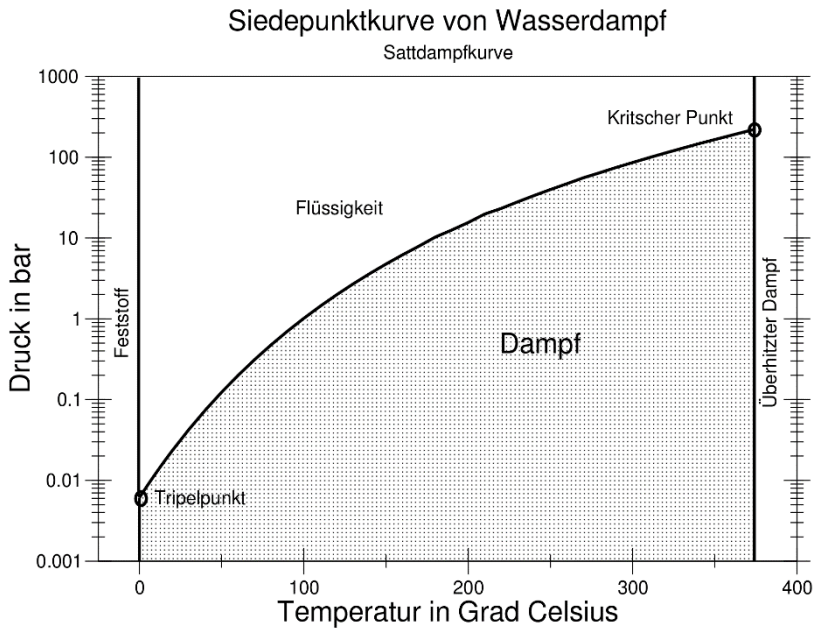
Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Weitere Definition

Feste, vordefinierte Regeln:

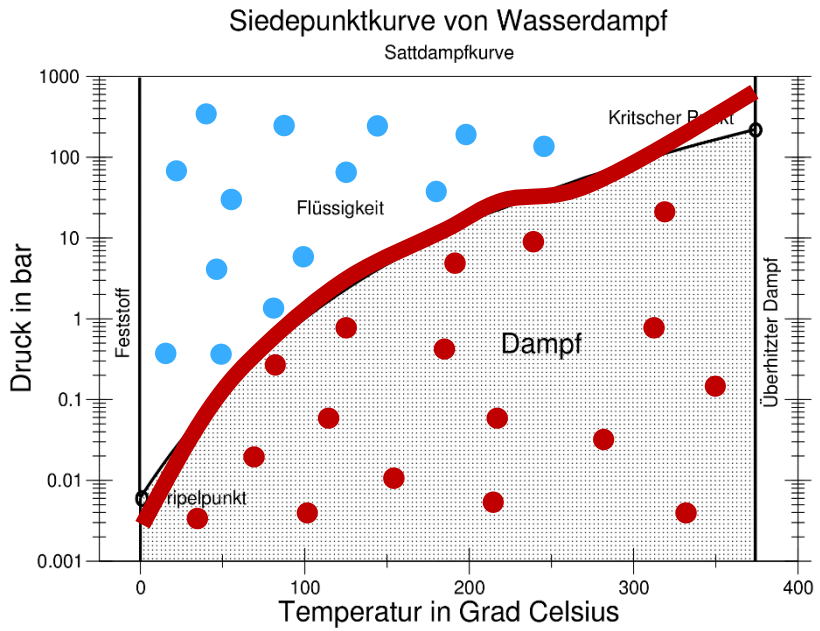
- Im Bereich $T > 0°C$ Wasser ist gasförmig, wenn

$$\mathrm{p} \geq p_{sat} = 288{,}68 \left(1{,}098 + \frac{T}{100°C}\right)^{8{,}02} Pa$$



Regeln werden aus Daten abgeleitet:

- Datenpunkte sind wie Stichproben
- Modell generalisiert Datenpunkte



https://de.wikipedia.org/wiki/Datei:Dampfdruckkurve.svg

Position des ML in der KI-Landschaft

Künstliche Intelligenz

Imitation menschlicher kognitiver Fähigkeiten:

- Robotik
- Natural-Language-Processing
- Sprachverarbeitung
- Maschinelles Sehen
- Experten- und Assistenzsysteme

Maschinelles Lernen

Reihe von statistischen Methoden für daten-getriebene Modellierung

Deep Learning

Methoden, die Neuronale Netze mit 3 und mehr Schichten ermöglichen

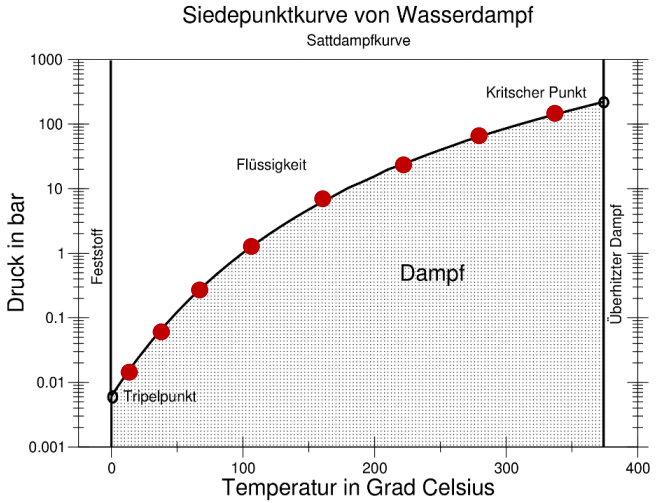
1950er → 1960er → 1970er → 1980er → 1990er → 2000er → 2006er → 2010er → 2012er → 2017er

Aufgaben für Machine Learning

Problemstellung für ML-Anwendung

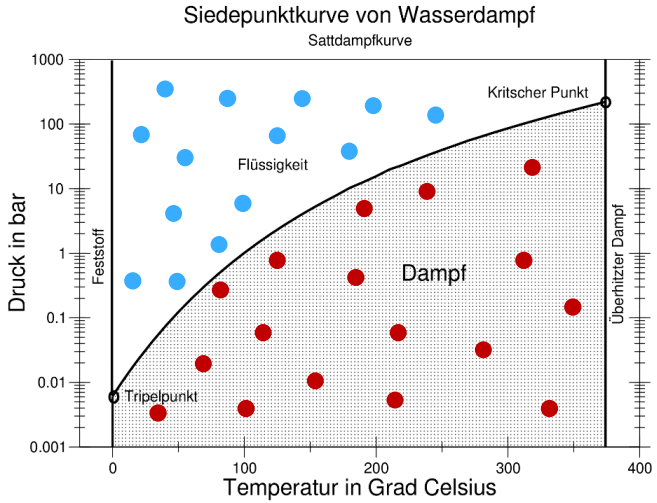
Regressionsaufgabe
Zielvariable: stetig, kontinuierlich → Kardinalskala

- Konzentration im Reaktor
- Temperatur am Ausgang
- Produktnachfrage
- **Siedepunktekurve**



Klassifikation/Clustering
Zielvariable: diskret→ Nominal- und Ordinalskalen

- Prozessphase
- Störung
- Strömungsregime
- **Aggregatzustand (Entscheidungsgrenze)**



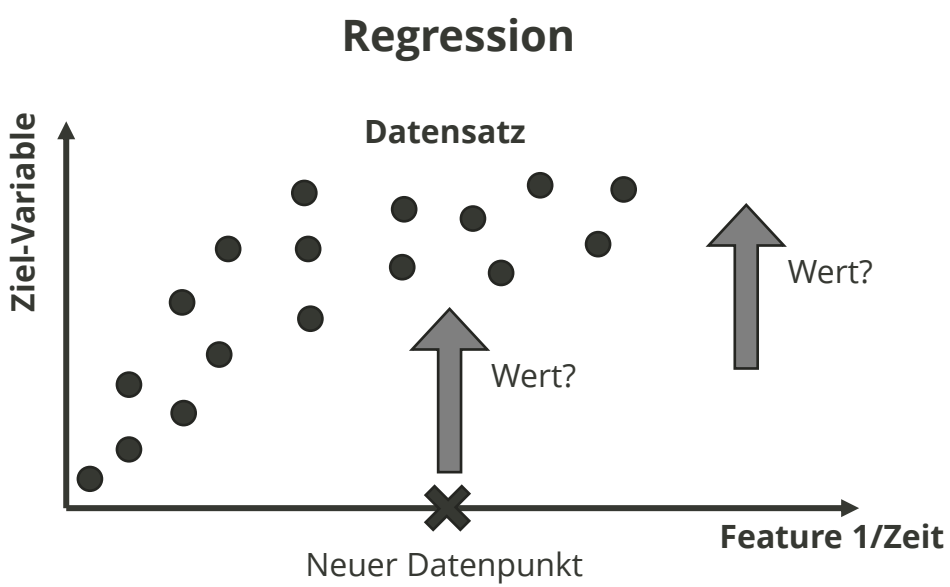
https://de.wikipedia.org/wiki/Datei:Dampfdruckkurve.svg

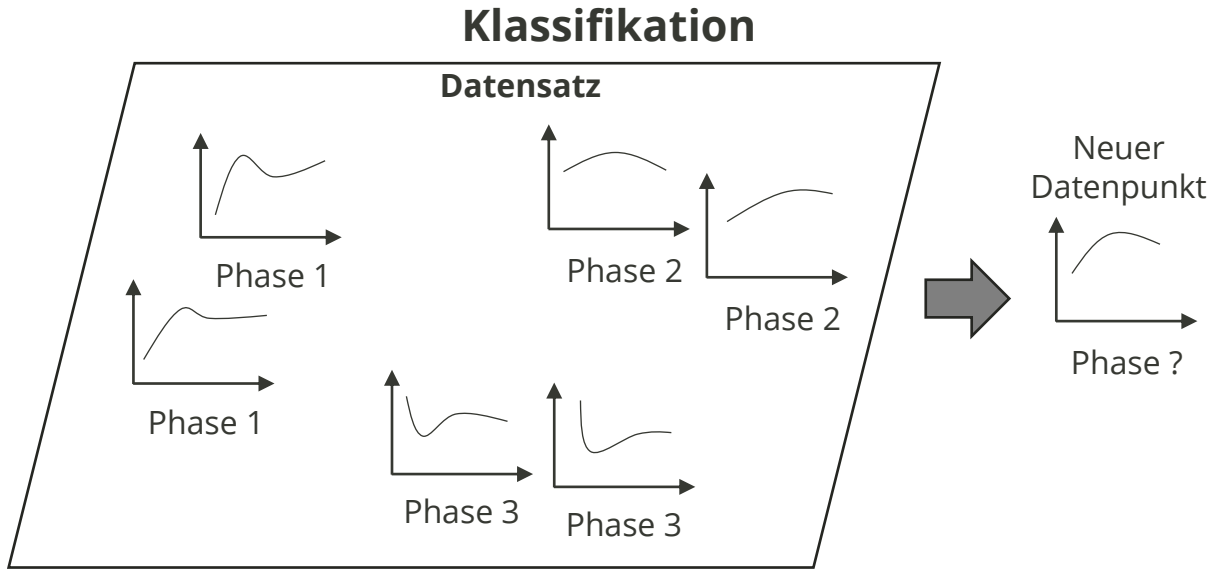
Problemstellung für ML-Anwendung

	Erfahrung	Aufgabe	Leistung
Überwachtes Lernen	Input-Variablen und Ziel-Variablen für alle Datenpunkte	Prädiktion Ziel-Variablen	Regression: MSE, MSLE, MAE Klassifikation: Cross-Entropy, Hinge
Unüberwachtes Lernen	Nur Input-Variablen	Erkennung Muster in Input-Variablen	Clustering: SE
Semiüberwachtes Lernen	Input-Variablen und Ziel-Variablen für begrenzte Anzahl Datenpunkte	Prädiktion Ziel-Variablen für Datenpunkte ohne Labels im Datensatz	divers
Bestärkendes Lernen	Agent und Umgebung	Entwicklung optimaler Strategie für Agent zum Umgang mit Umgebung	Belohnung

Überwachtes Lernen

Erfahrung	Aufgabe	Leistung
Input-Variablen und Ziel-Variablen für alle Datenpunkte	Prädiktion Ziel-Variablen	Regression: MSE, MSLE, MAE Klassifikation: Cross-Entropy, Hinge



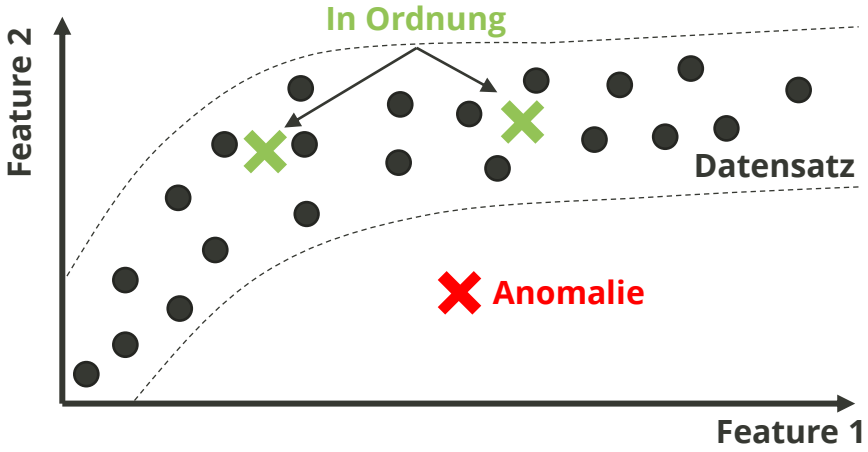


Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

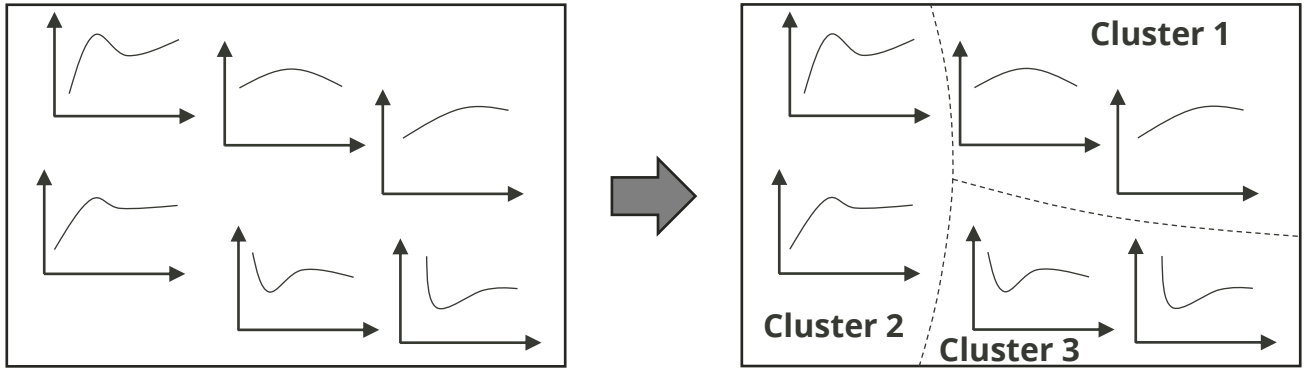
Unüberwachtes Lernen

Erfahrung	Aufgabe	Leistung
Nur Input-Variablen	Erkennung Muster in Input-Variablen	Clustering: SE

Anomalieerkennung



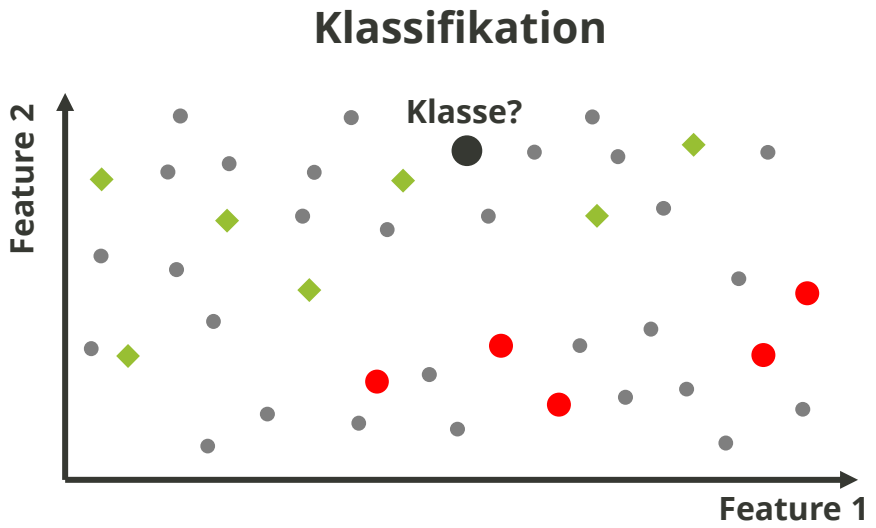
Clustering



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

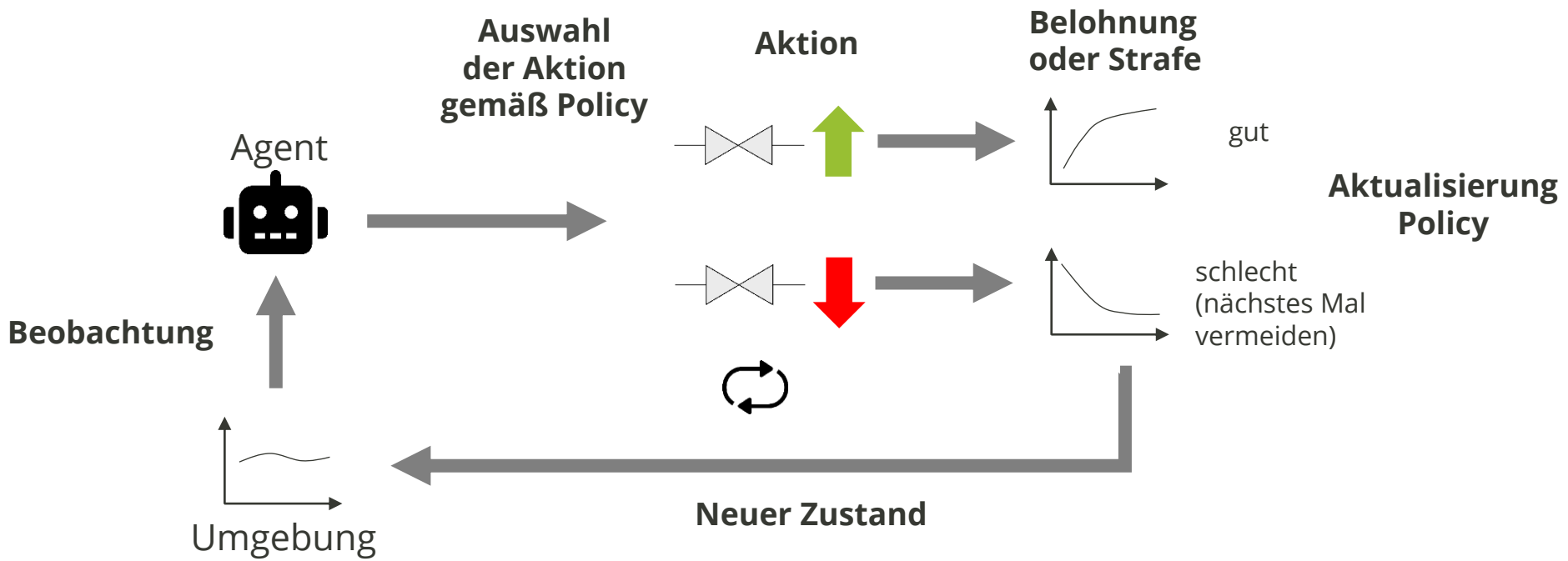
Semiüberwachtes Lernen

Erfahrung	Aufgabe	Leistung
Input-Variablen und Ziel-Variablen für begrenzte Anzahl v. Datenpunkten	Prädiktion Ziel-Variablen für Datenpunkte ohne Labels im Datensatz	divers



Bestärktes Lernen

Erfahrung	Aufgabe	Leistung
Agent und Umgebung	Entwicklung optimaler Strategie für Agent zum Umgang mit Umgebung	Belohnung



Vorgehensweise

Vorgehensmodelle

Motivation:

- Methodische Grundlage für effektives **Projektmanagement** und Ergebnisgarantie
  - Möglichkeit Projekt zu beeinflussen (allgemein, Aussuchen v. Teammitglieder, Identifikation von Meilensteinen und Deliverables, Zeitplanung)
- Fruchtvolle **Zusammenarbeit** und Interaktionen zwischen verschiedenen Spezialisten
  - Wer und wem werden welche Artefakte geliefert?
- **Allgemeine Sprache** zwischen allen Beteiligten des Prozesses (Stackholder, Entwickler, Manager, Koordinator, Tools-Entwickler)
  - Schneller Start des Projekts möglich ohne Abstimmung von Begriffen
  - Gemeinsames Verständnis, was derzeit im Projekt passiert und was noch ansteht
  - Dokumentation
  - Zielführende Entwicklung neuer Werkzeuge
- Erhöhte **Vertraulichkeit** in Ergebnisse
  - Ergebnisse sind nachvollziehbar und wiederholbar
- Strukturierte **Ausbildung** und vereinfachte **Einarbeitung** in das Themagebiet

Vor der Entwicklung ist immer ein Vorgehensmodell **abzustimmen** und dann **zu verfolgen**

R. Wirth, “CRISP-DM : Towards a Standard Process Model for Data Mining,” *Proc. Fourth Int. Conf. Pract. Appl. Knowl. Discov. Data Min.*, no. 24959, pp. 29–39, 2000.

CRISP-DM

Vorgehensmodell für Data-mining-Projekte

Wasserfallmodell mit Backtracking (trial and error)

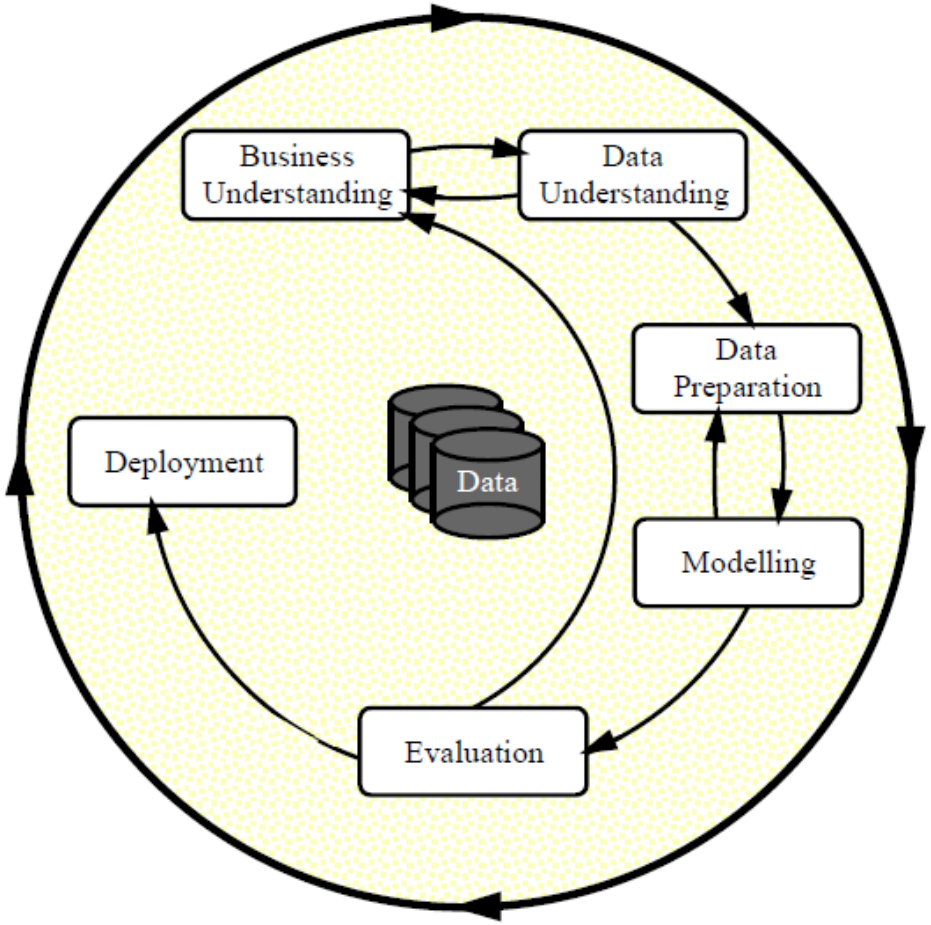
- Ausarbeitung und Abbruch von Teillösungen bis eine Gesamtlösung gefunden werden kann

Besteht aus allgemeinen Aufgaben (s. Abbildung)

Spezialisiert für verschiedene Domänen:

- CRISP-TDM, CRISP-DM0, CRISP-MED-DM

CRISP-ML(Q) – Modifikation mit QA-Methodologie in jeder Phase



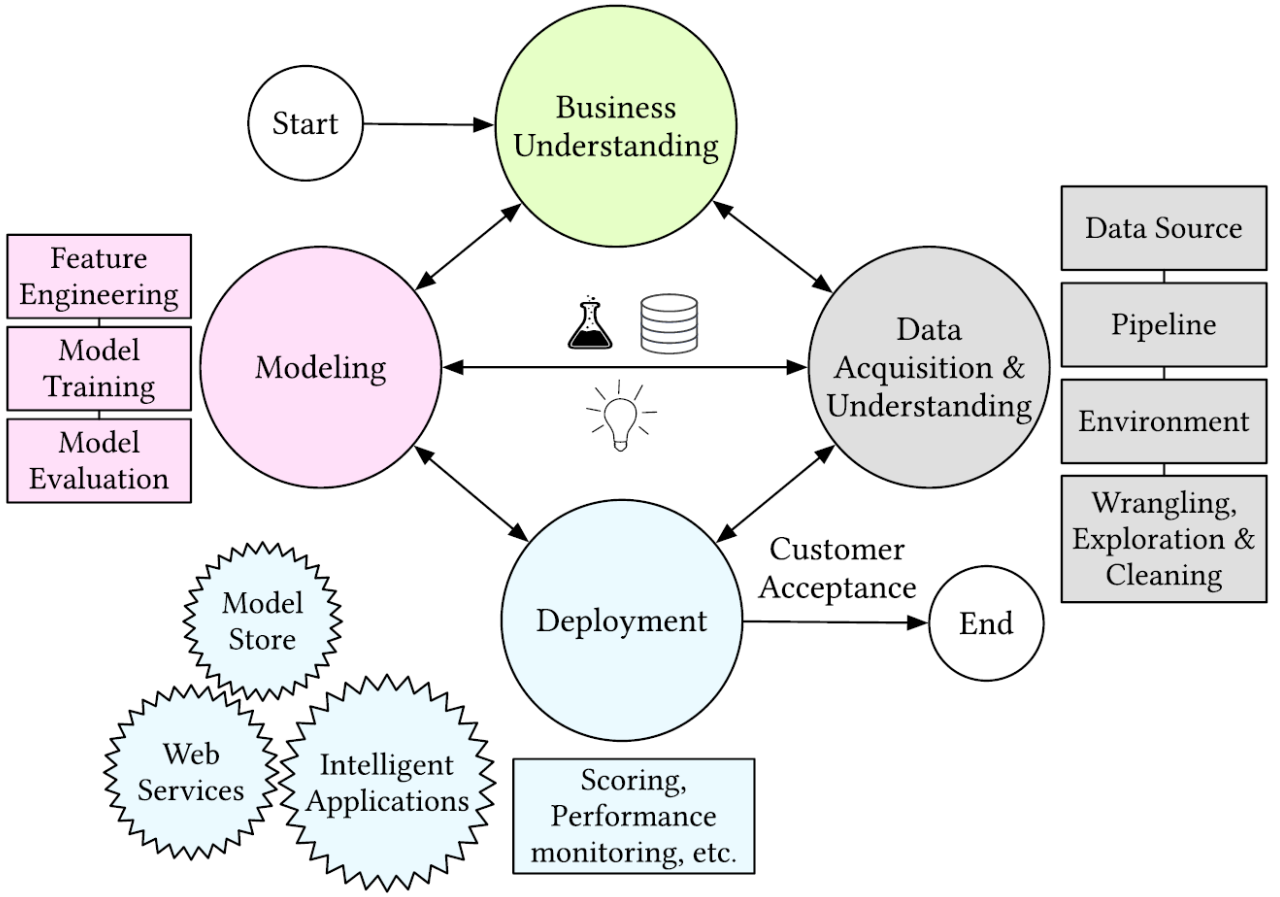
R. Wirth, "CRISP-DM : Towards a Standard Process Model for Data Mining," *Proc. Fourth Int. Conf. Pract. Appl. Knowl. Discov. Data Min.*, no. 24959, pp. 29–39, 2000.
S. Studer *et al.*, "Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology," 2020, [Online]. Available: http://arxiv.org/abs/2003.05155.

Team Data Science Process

TDSP ist eine feindetaillierte Methodologie für Entwicklung im Bereich Prädiktive Analytik und Intelligente Anwendungen

Komponenten:

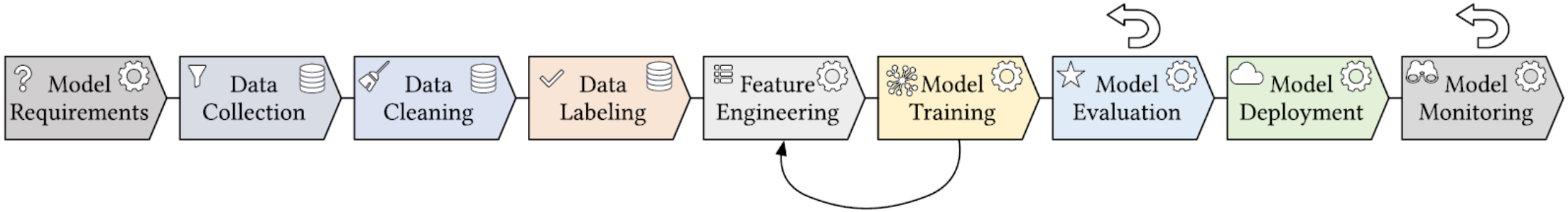
- Lebenszyklus (s. Abbildung rechts)
- Rollen: Solution Architect, Projekt manager, Data engineer, Data scientist, Application developer, Project lead. Jede Rolle hat bestimmte Aufgaben
- Projektstruktur
- Infrastruktur-Stack
- Entwicklungsstack



https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview

M. Haakman, L. Cruz, H. Huijgens, and A. van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–30, 2021, doi: 10.1007/s10664-021-09993-1.

Vorgehensmodell nach Amershi



Hintergrund des Modells ist der Prozess der Softwareentwicklung (agile Methoden, DevOps, Verschwinden von Grenzen zwischen Rollen: Entwickler und Tester → kombiniert in einer Rolle im Rahmen von DevOps)

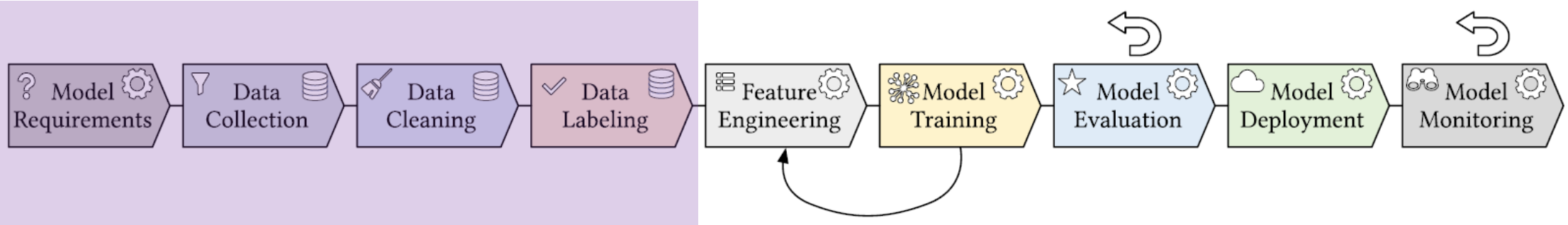
Amershi-Modell:

- Deckt den Lebenszyklus einer ML-Anwendung **End-to-End** ab
- Nur Entwicklung ohne Geschäftsebene: es fehlt **Business-Understanding-Phase** (vgl. CRISP-DM)
- Das Modell ist ebenfalls datenzentriert (wie z.B. CRISP-DM)
- Mit Rückführung für Backtracking, entspricht agiler Vorgehensweise (**iterativ**)

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.

M. Haakman, L. Cruz, H. Huijgens, and A. van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–30, 2021, doi: 10.1007/s10664-021-09993-1.

Vorgehensmodell nach Amershi



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

Input: **Anforderungen**

Arbeit mit Daten:

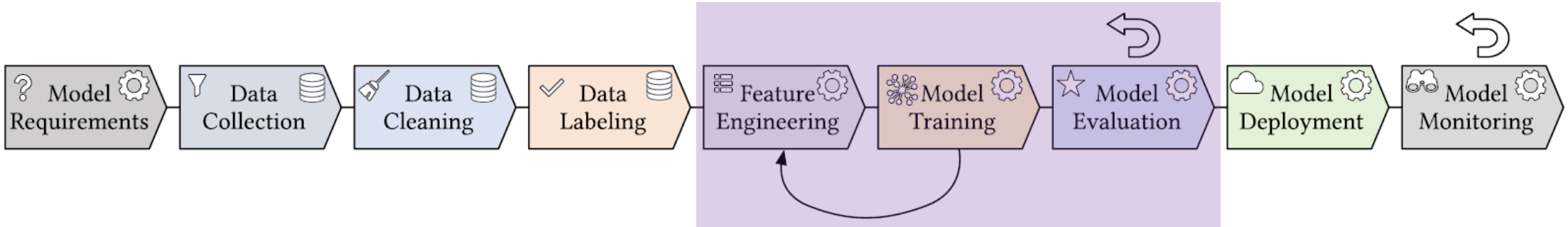
- Data collection – Bereitstellung des Datensatzes: Datenimport oder –gewinnung
- Data clearning – Aufbereitung des Datensatzes
- Data labeling – Markierung von Daten (Überwachstes Lernen)

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.

M. Haakman, L. Cruz, H. Huijgens, and A. van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–30, 2021, doi: 10.1007/s10664-021-09993-1.

Vorgehensmodell nach Amershi



Input: **Anforderungen und Datensatz**

Arbeit mit Modell:

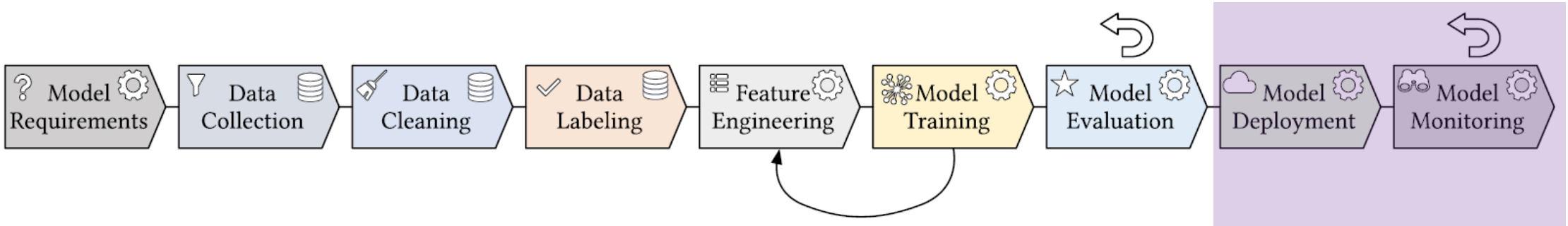
- Feature engineering – Auswahl von Features und deren Aufbereitung für das Training
- Model training – Trainieren, Optimierung von Modell- und Training-Hyperparametern
- Model evaluation – Testen des Modells mit einem Test-Datensatz, Berechnung von Metriken, Auswahl eines Modells für Einsatz in Produktion

Output: **Production-ready Modell**

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.

M. Haakman, L. Cruz, H. Huijgens, and A. van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–30, 2021, doi: 10.1007/s10664-021-09993-1.

Vorgehensmodell nach Amershi



Input: **Production-ready Modell**

Deployment und 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)

Output: **ML-Anwendung in Betrieb**

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.

M. Haakman, L. Cruz, H. Huijgens, and A. van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," *Empir. Softw. Eng.*, vol. 26, no. 5, pp. 1–30, 2021, doi: 10.1007/s10664-021-09993-1.

Exemplarische Methoden

Lineare und nichtlineare Regression

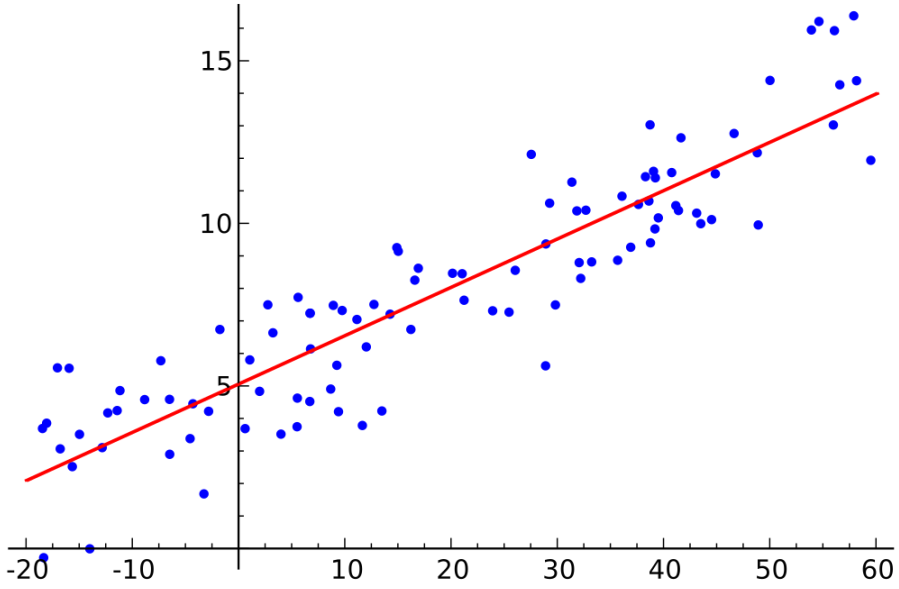
Lineare Regression ist das einfachste und meist verwendete Modell für Regressionsaufgaben für Aufgaben des **Überwachten Lernens**.

Prinzip

- Modellansatz (Hypothese) $y = b_0 + b_1 x_1 + \cdots + b_{m_1} x_{m_1} (+e)$
- Schätzung der Parameter in der Regel nach Methode der kleinsten Quadrate aus Stichprobe oder des Abstiegsverfahrens

Abbildung der Nichtlinearität durch das Hinzufügen neuer Features

- Polynomiale Hypothesen (Quadratische, Kubische etc.)
  - $y = b_0 + b_1 x_1 + \cdots + b_{m_1} x_{m_1} + b_{n_1} x_1^2 + b_{n_1} x_1 x_2 + \cdots$
- Nichtlineare Regression $y = \frac{e^{b_1 x_1 x_2 + b_2 x_2}}{b_3 + b_4 x_1}$
- Weitere Hypothesen beliebiger Form möglich



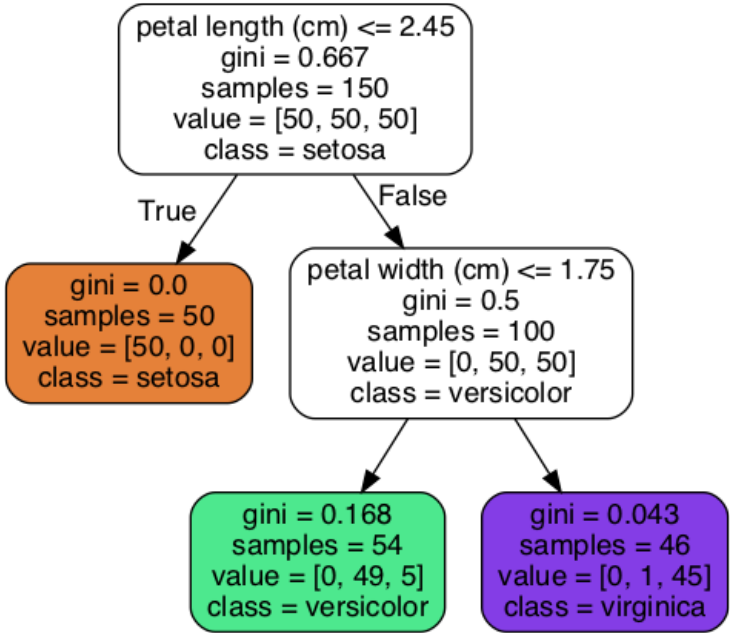
Entscheidungsbäume

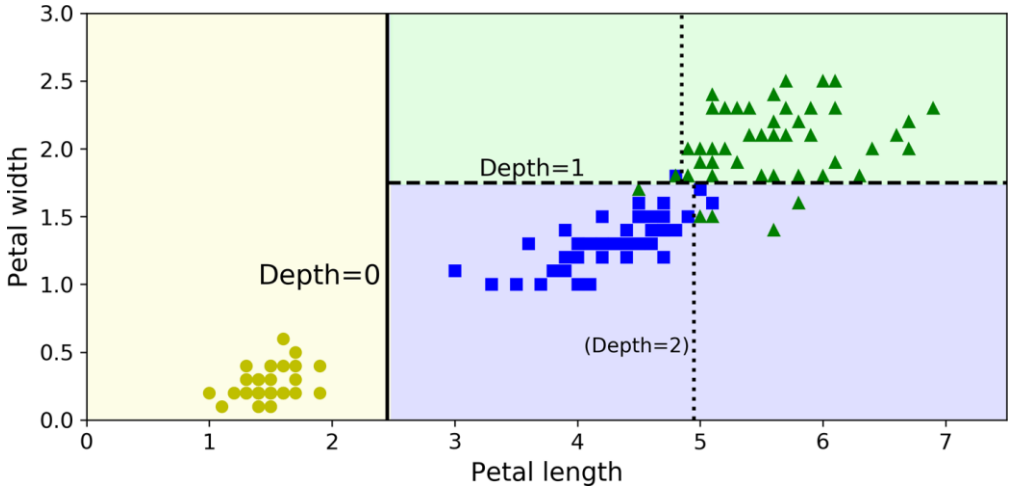
Der Entscheidungsbaum ist eine Methode für Regression und Klassifikation als Aufgaben **des Überwachten Lernens**

Arbeitsprinzip (für Induktion/Trainieren des Models):

- Top-Down
- Auswahl eines Features, das das beste Aufteilen ermöglicht (Maß ist sog. **Information-Gain** wie z.B. **Gini-Score** oder **Entropy**)
- Das ausgewählte Feature wird zur Aufteilung genutzt
- Rekursive Wiederholung der Auswahl und Aufteilung bis die Teilmenge entweder keine Klassifikation mehr ermöglicht oder die vorgegebene Tiefe erreicht ist
- **Post-Pruning** zum Entfernen von Blättern mit niedriger Relevanz (zur Bewältigung des Overfitting-Problems) →**Regularisation**

Entscheidungsbäume sind Komponente der Methode **Random-Forest** (sehr mächtiges Verfahren)



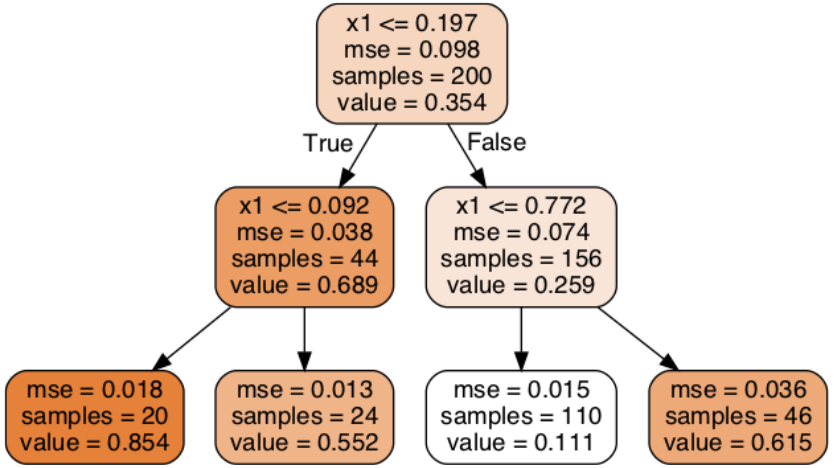


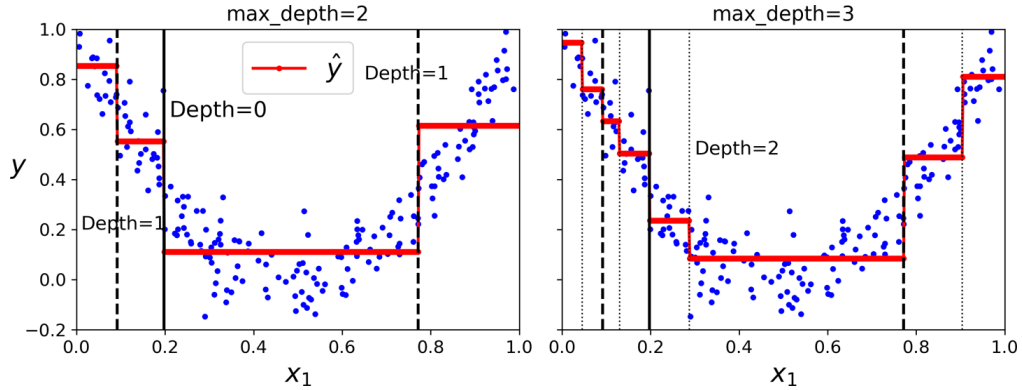
A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Entscheidungsbäume

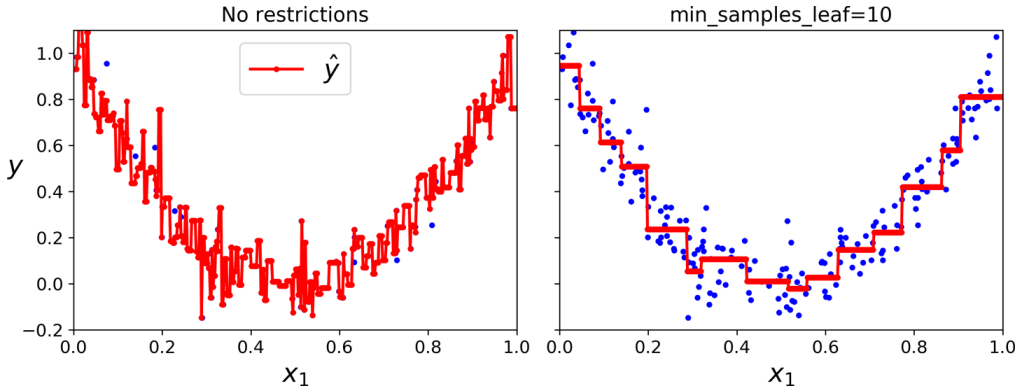
Für Regressions-Aufgaben: Einsatz einer anderen Zielfunktion (z.B. MSE oder MAE)

Prädiktion von (**diskreten**) Werten anstelle von Klassen





Overfitting: Ohne und mit Regularisation



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

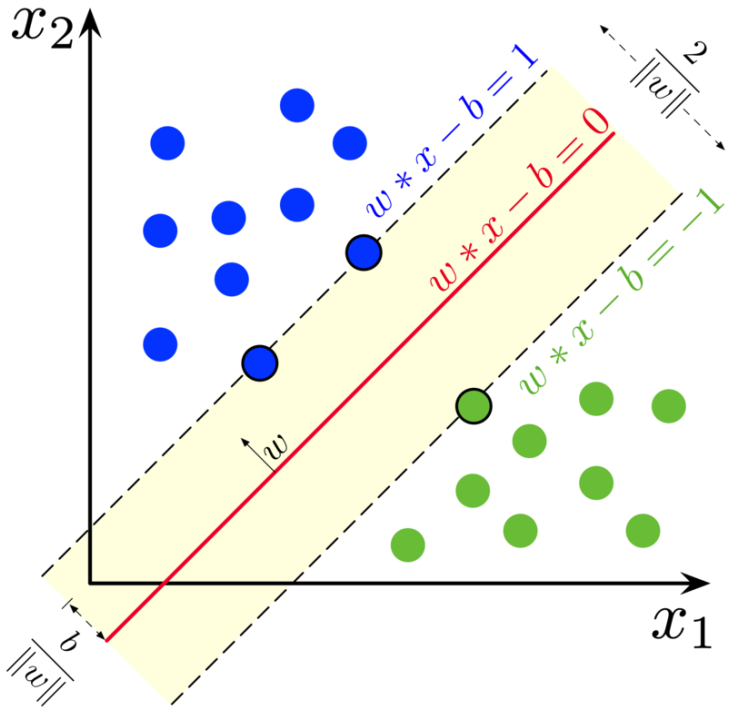
SVM (Support Vector Machine)

SVM ist eine Methode der Klasse **überwachtes Lernens (Ziel-Variable ist eine Klasse)** zur Klassifikation (und Regression)

Arbeitsprinzip: Finde eine Hypereben (Entscheidungsgrenze), die den Abstand zu Punkten möglichst groß hält und sich auf der Seite der Klasse befindet. Punkte, die am Rand der Streifen liegen heißen Support-Vektoren

- Ursprünglich entwickelt nur für linear, trennbare Klassen (Vapnik und Chervonenkis, 1963)
- Anwendung des Kernel-Tricks ermöglicht Abbildung nicht-linearer Grenzen (Polynomial, Radial-Basis-Funktion, Sigmoid-Kernel usw.)
- Soft-Margin-Ansatz führt einen C-Zusatzparameter zum Tolerieren von Ausreißer ein

Regression: alle Punkte sollen innerhalb der Streife liegen



Austauschverfahren oder K-Means-Algorithm

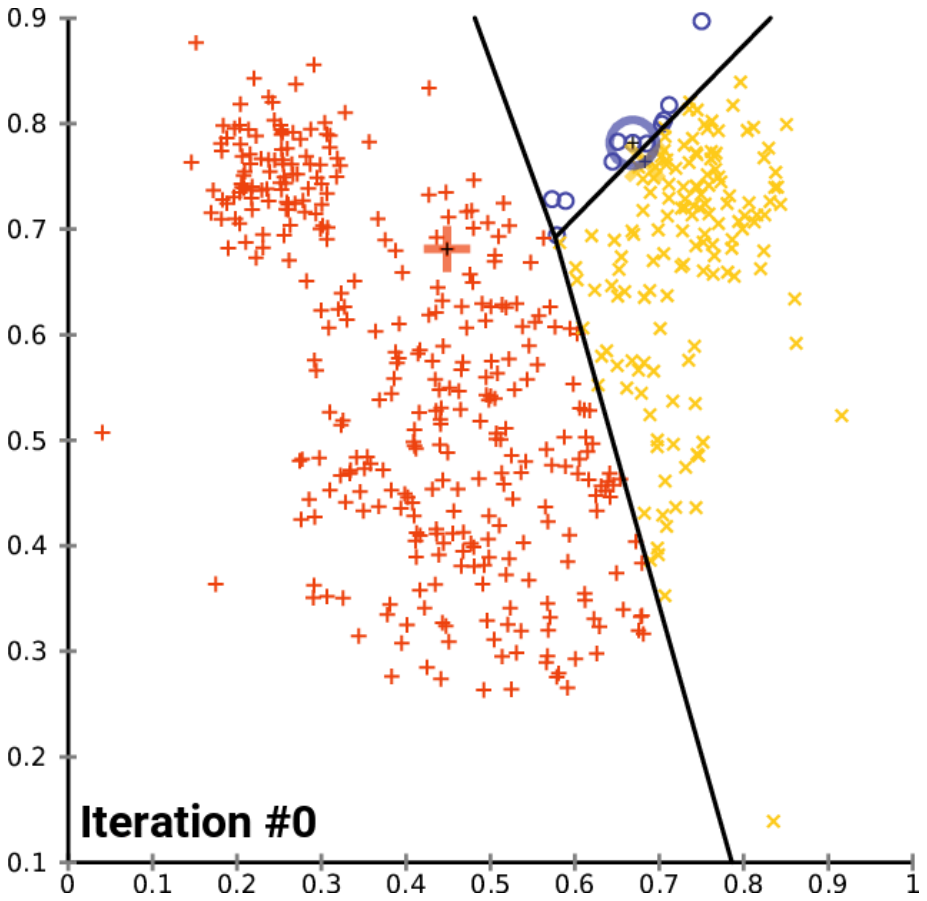
Austauschverfahren ist eine Methode für Suche nach einer **optimale Klasseneinteilung** bezüglich eines Kriteriums i.d.R. das Varianzkriterium oder der **Abstand zwischen Punkten verschiedener Klassen**.

Arbeitsprinzip:

- **Initialisierung**: Auswahl zufälliger Mittelwerte von Clustern
- **Iterationen**:
  - Jeder Datenpunkt wird einem Cluster zugeordnet, bei dem das Varianzkriterium minimal wird
  - Neuberechnung Mittelpunkte von Clustern
  - Wiederholung bis keine weitere Verschiebung von Mittelpunkten erfolgt

Anzahl von Clustern soll vorgegeben sein.

Lokales Verfahren (Ergebnis von Initialwerten abhängig)



https://upload.wikimedia.org/wikipedia/commons/e/ea/K-means_convergence.gif

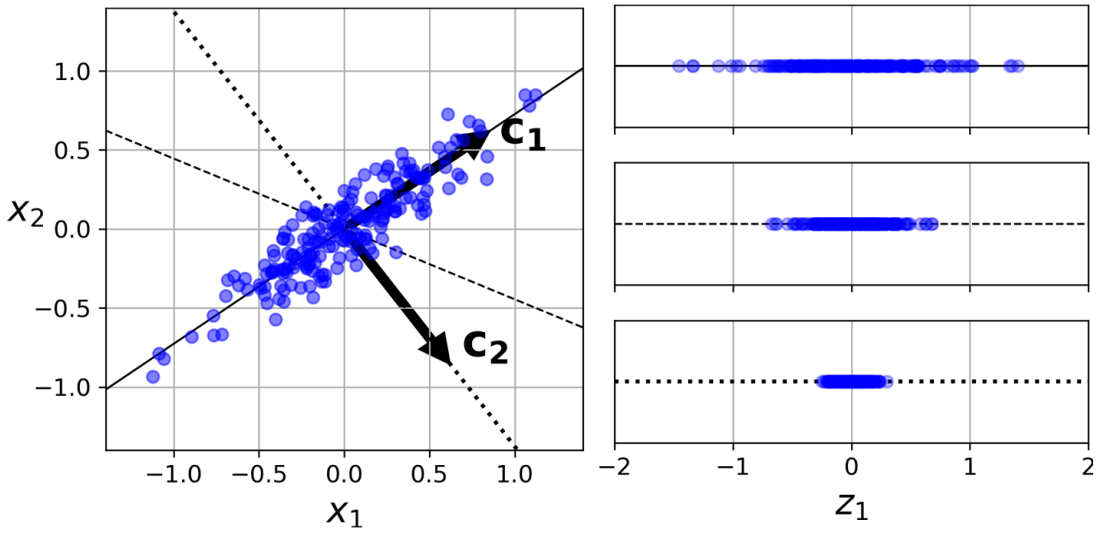
Hauptkomponentenanalyse (PCA)

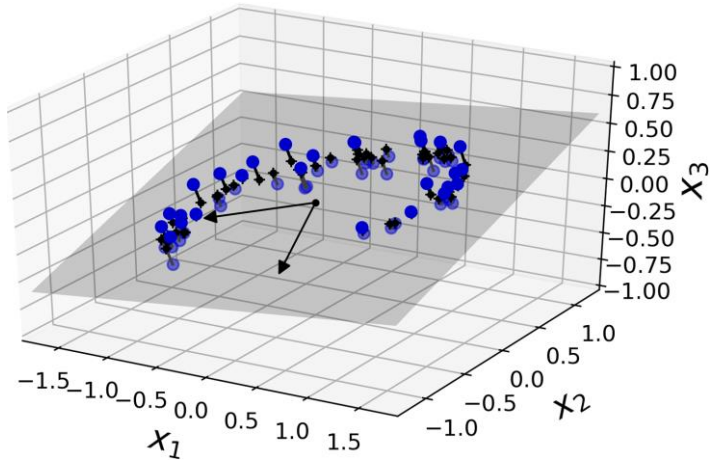
PCA ist eine Methode für **Dimensionsreduktion**, wobei die Messpunkte eines n-Dimensionsraums als Messpunkte eines m-Dimensionsraums modelliert werden, wobei m<n.

Arbeitsprinzip:

- Wähle eine neue Achse (Hauptkomponente) so, dass die Varianz maximal ist
- Wiederhole den vorherigen Schritt n-Mal
- Mittels erster Hauptkomponenten bilde eine Hyperebene, worauf Datenpunkte projiziert werden.

Anzahl von genommenen Hauptkomponenten und deren Varianz definiert die resultierende Varianz.





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

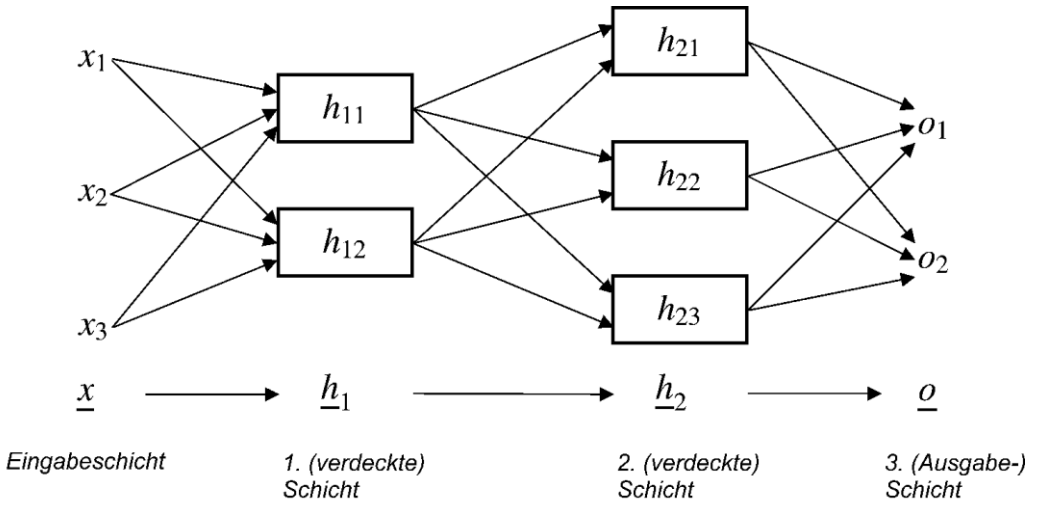
Neuronale Netze

$\underline{x} \in \mathbb{R}^m$ Eingabevektor (einschließlich BIAS)
$\underline{o} \in \mathbb{R}^{m'}$ Ausgabevektor
$\underline{h}_1, \underline{h}_2, .., \underline{h}_{n-1}$ Hilfsvektoren

f Transferfunktion (bzw. Aktivierungsfunktion)
$\underline{W}_1, \underline{W}_2, \dots, \underline{W}_n$ Gewichtsmatrizen passender Größe

n-stufiges Neuronales Netz berechnet den Vektor $\underline{o}$ aus $\underline{x}$ nach:

$$\underline{h}_1 = f(\underline{net}^1) = f(\underline{W}_1 \underline{x})$$
$$\underline{h}_2 = f(\underline{net}^2) = f(\underline{W}_2 \underline{h}_1)$$
$$\underline{h}_3 = f(\underline{net}^3) = f(\underline{W}_3 \underline{h}_2)$$
$$\vdots$$
$$\underline{o} = f(\underline{net}^2) = f(\underline{W}_n \underline{h}_{n-1})$$



Spezialfall: n = 1 und i.d.R. binäre Eingabewerte

- charakteristische Funktion als Transferfunktion und ein (binäres) Ausgabeneuron
  - (klassisches) **Perzeptron**

Perzeptron-Algorithmen: Gruppe Neuronaler Netze ähnlich einfacher Struktur (60er Jahren, ROSENBLATT)

Zusammenfassung

Zusammenfassung

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

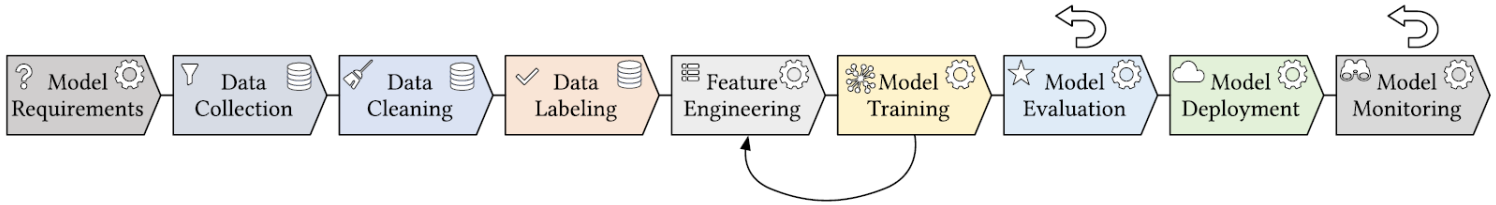
Je nach Ziel-Variable: **Regressionsaufgabe** (Kardinalskala), **Klassifikation** oder **Clustering** (Ordinal- oder Nominalskala)

Aufgaben:

	Erfahrung	Aufgabe
Überwachtes Lernen	Input-Variablen und Ziel-Variablen für alle Datenpunkte	Prädiktion Ziel-Variablen
Unüberwachtes Lernen	Nur Input-Variablen	Erkennung Muster in Input-Variablen
Semiüberwachtes Lernen	Input-Variablen und Ziel-Variablen für begrenzte Anzahl Datenpunkte	Prädiktion Ziel-Variablen für Datenpunkte ohne Labels im Datensatz
Bestärkendes Lernen	Agent und Umgebung	Entwicklung optimaler Strategie für Agent zum Umgang mit Umgebung

Zusammenfassung

Vorgehensmodell nach Amershi:



- Data collection – Bereitstellung des Datensatzes: Datenimport oder –beschaffung **Fokus der nächsten Vorlesung**
- 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
- 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.

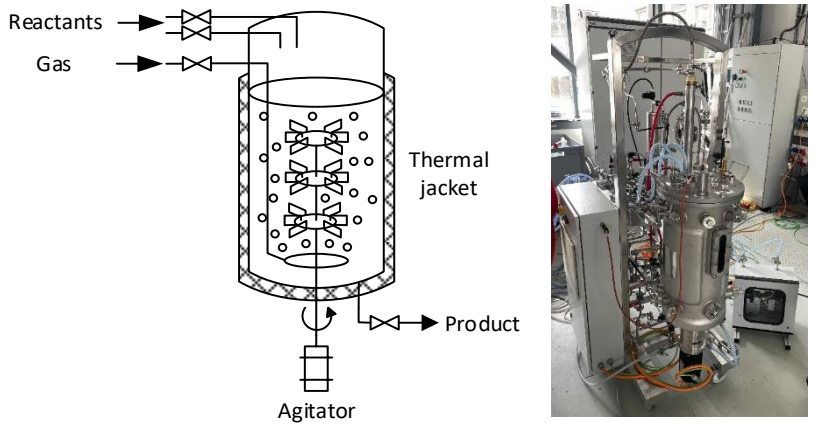
Dr. rer. nat. Valentin Khaydarov
Email: valentin.khaydarov@tu-dresden.de
Telefon: 0351 463 33387

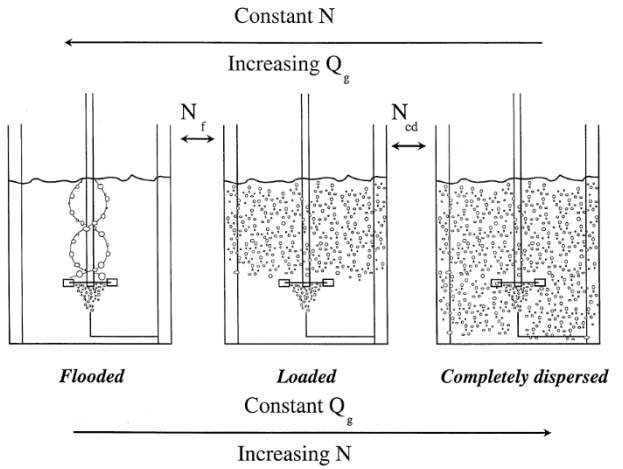
Vielen Dank für Ihre Aufmerksamkeit!

Beispiel:
Bilddaten-basierter Smart-Sensor

1. Business Understanding

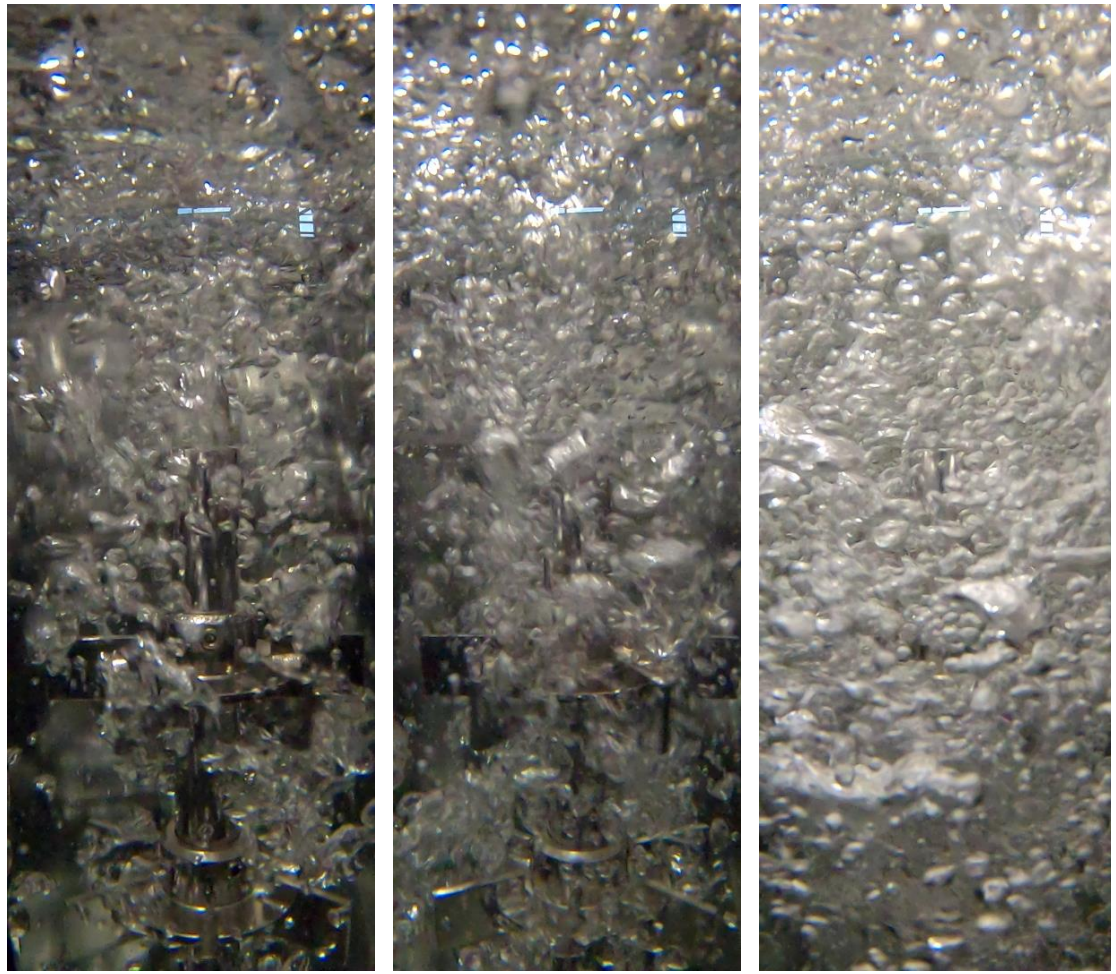
- Ziel: zuverlässig Strömungsregime erkennen
- Strömungsregime → Softsensor für Beobachtung und Regelung des chemischen Prozesses im Bioreaktor
- Input: Bilder
- Output: Strömungsregime (Klasse)
- Datensatz:
  - Kein existierender Datensatz
  - Manuell getaggte Fotos → **Datengewinnung**
- Echtzeitfähigkeit





A. Paglianti, S. Pintus, and M. Giona, "Time-series analysis approach for the identification of flooding/loading transition in gas-liquid stirred tank reactors," *Chem. Eng. Sci.*, vol. 55, no. 23, pp. 5793–5802, 2000, doi: 10.1016/S0009-2509(00)00125-1.

2. Data Understanding



Flooded Loaded C. dispersed

3. Data Preparation 

Data Collection

- Bildaufnahme mit GoPro Hero 7 und Fujifilm X-T20
- Varianz von Prozessdaten: Drehzahl, Gaszufuhr und Füllmenge
- Kameraeinstellungen
- Varianz von Aufnahmebedingungen: Kameraposition, Lichtbedingungen

Data Cleaning

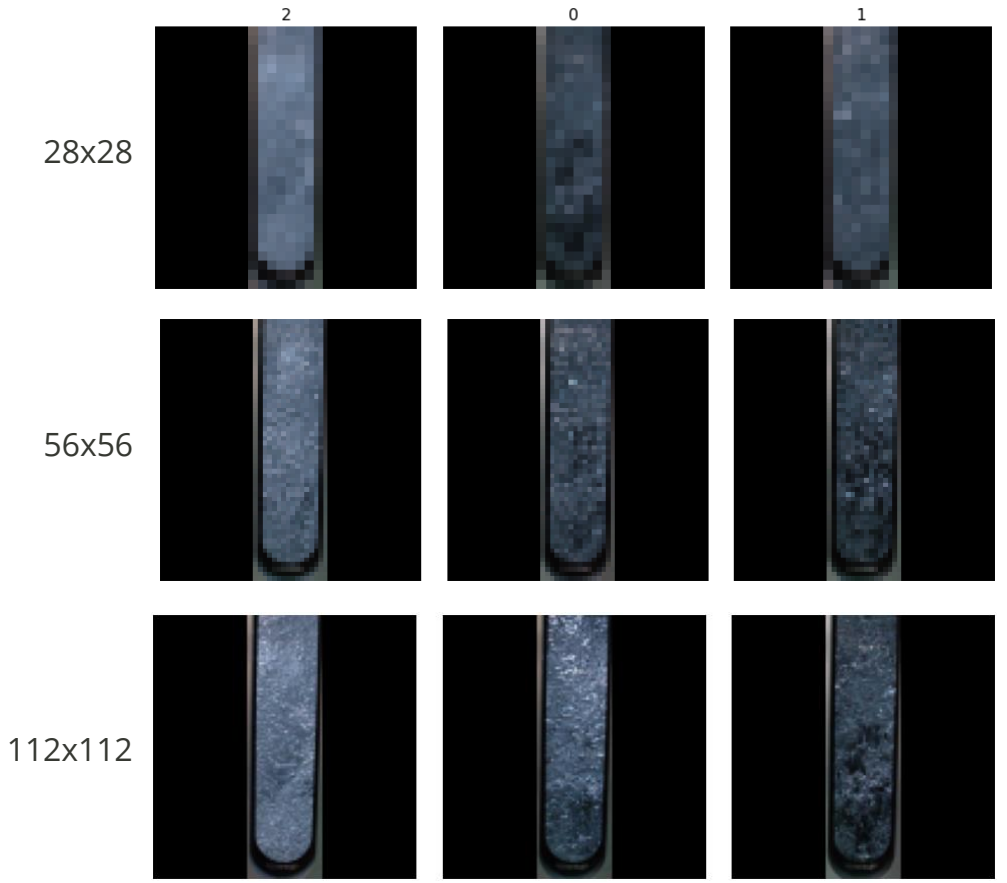
- Manuelle Inspektion von Bildern und Entfernung beschädigter Bilder

Data Labeling

- Manuell
- Validierung mittels Strömungskarte

3. Data Preparation

- cropping (manuell)
- sharpening (3x3, sharpening kernel)
- resizing (28x28, 56x56, 112x112)
- padding
- sobel/scharr filter
- canny edge detection
- histogram equalisers



C. Kröger (2021)
Diploma Thesis: AI based detection of flow regimes (TU Dresden)

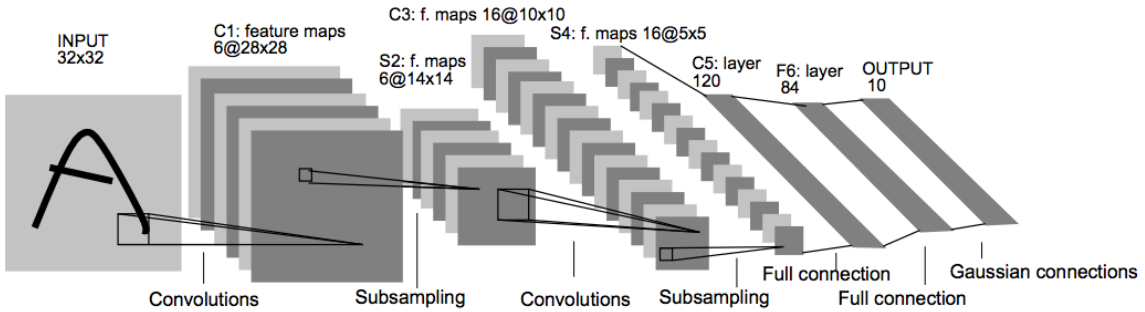
4. Model Training und Evaluation

Trainierte Modelle:

LeNet-5 (28x28): 79% accuracy

LeNet-5 (56x56): 86% accuracy

LeNet-5 (112x112): 38% accuracy - overfitted



Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

Custom CNN: 224x224, 2 conv, 2 maxpooling, 1 dense, RELU activation: 40%

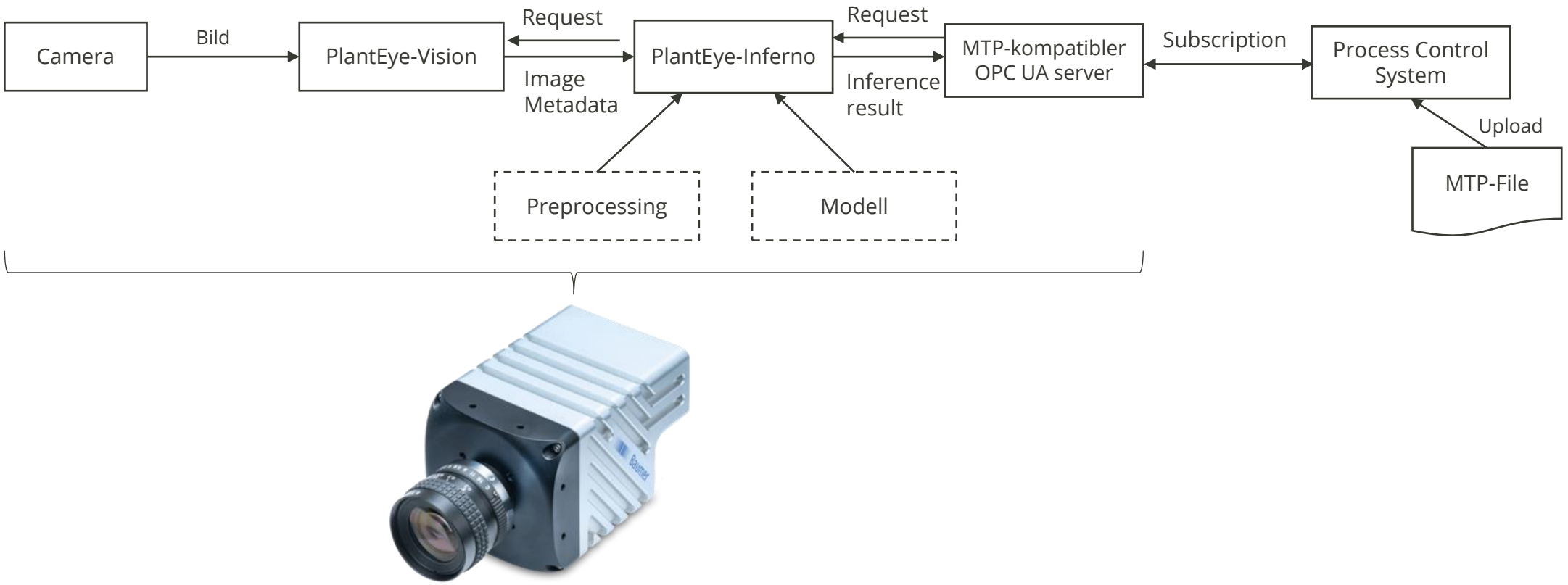
Custom CNN: 224x224, 3 conv, 3 maxpooling, 1 dense, RELU activation: 40%

VGG-16, MobileNet: 34% accuracy (40% mit Transfer-Learning)

Hauptproblem: **Overfitting**

Datenaugmentation und Dropout (0.2) – bessere Ergebnisse: 83%→90%

5. Model Deployment



https://www.baumer.com/sg/en/product-overview/industrial-cameras-image-processing/industrial-cameras/ax-series/c/43083