

# Machine Learning

---

mit  
Python

# Lizenz

---

## Lizenz

- Die Charts & begleitende Kursmaterialien wurden – soweit nicht anders angegeben – von Ralf Bendig erstellt. [Lizenz CC BY 4.0](#).
- Mit einem Badge gekennzeichnete Charts, Aufgaben & Lösungen (Workflows) stammen aus:
  - [L2-DS] KNIME Analytics Platform for Data Scientists: Advanced, Stand 08/2022 der [KNIME AG](#)
  - [L4-ML] Introduction to Machine Learning Algorithms, Stand 08/2022 der [KNIME AG](#)
  - Michael R. Berthold et al., Guide to Intelligent Data Science Second Edition, 2020, [Teaching Material](#)
  - KI-Campus, Stifterverband für die Deutsche Wissenschaft e.V.



Lizenz der Quellen: [Lizenz CC BY 4.0](#).

Änderungen an den Charts: Übersetzung Englisch – Deutsch, textliche Anpassungen

Titelseite: Bild mit DALL·E erstellt

# Inhalt

---

- Intro
- Machine Learning Basics
- Colab & Notebooks
- Machine Learning Process
- Model-Toolbox
- Special Neural Networks
- Media Pipe Studio
- Workflow Design
- XAI - eXplainable AI
- Anhang

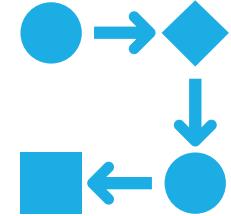
# Intro

---



# Kurs-Organisation

---



## ZEITPLANUNG

- 5 Tage
- Start: 09:00 Uhr
- Ende: 16:30 Uhr
- Pause nach 90 Min

## VORGEHEN

- Grundlagen/Basiswissen
- Beispiel
- Training/Fallstudie
- Diskussion Ergebnisse

## VERSCHIEDENES

- Pinboard

# Hinweise zum Skript & den Notebooks

---

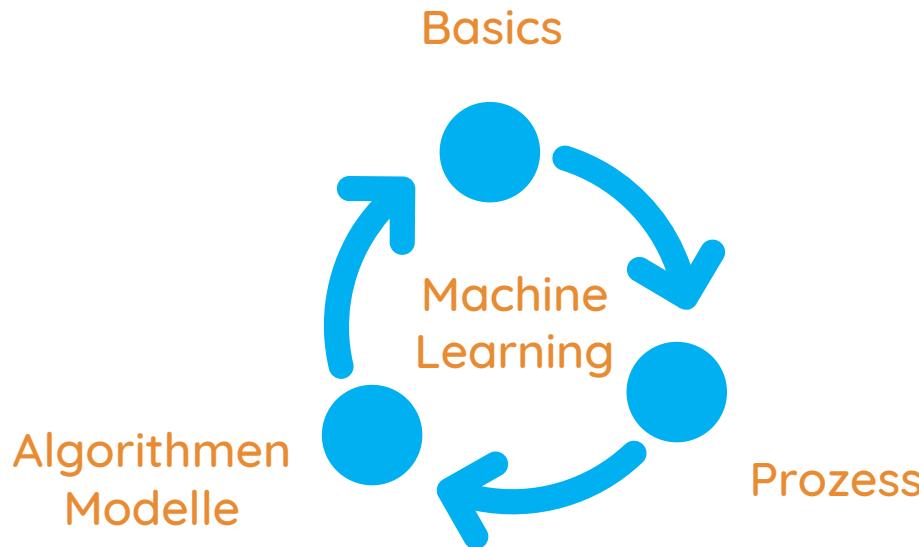


Bild von [Susanne Weitzhofer](#) auf [Pixabay](#)

- Das Skript ist eine **Sammlung** von Themen zu Machine Learning.
- Die wesentlichen Themenfelder, wie Algorithmen & Modelle, Prozesse & Einsatzbereiche, werden dargestellt.
- Das Skript ist als **Begleitmaterial** zum **Kurs** konzipiert und kein klassisches Lehrbuch.
- Hinweise zu **Lehrbüchern** für Machine Learning finden sich in der Literaturliste im Anhang.
- Die Notebooks beinhalten Beispiele & Trainingsmaterial und Lösungen.
- Die Charts & Notebooks sind **teilweise optional**.

# Vorgehensweise

---



- Wir beginnen mit ausgewählten Grundlagen.
- Diese Grundlagen sind der Startpunkt, von dem aus wir unser Wissen und unsere Fähigkeiten **schrittweise** erweitern.
- So können wir erreichen, dass
  - wir auf einem **ersten Fundament** aufbauen,
  - **schnell** erste Modelle entwickeln und
  - schrittweise **komplexere** Modelle und Prozesse konzipieren.

# IT-Setup

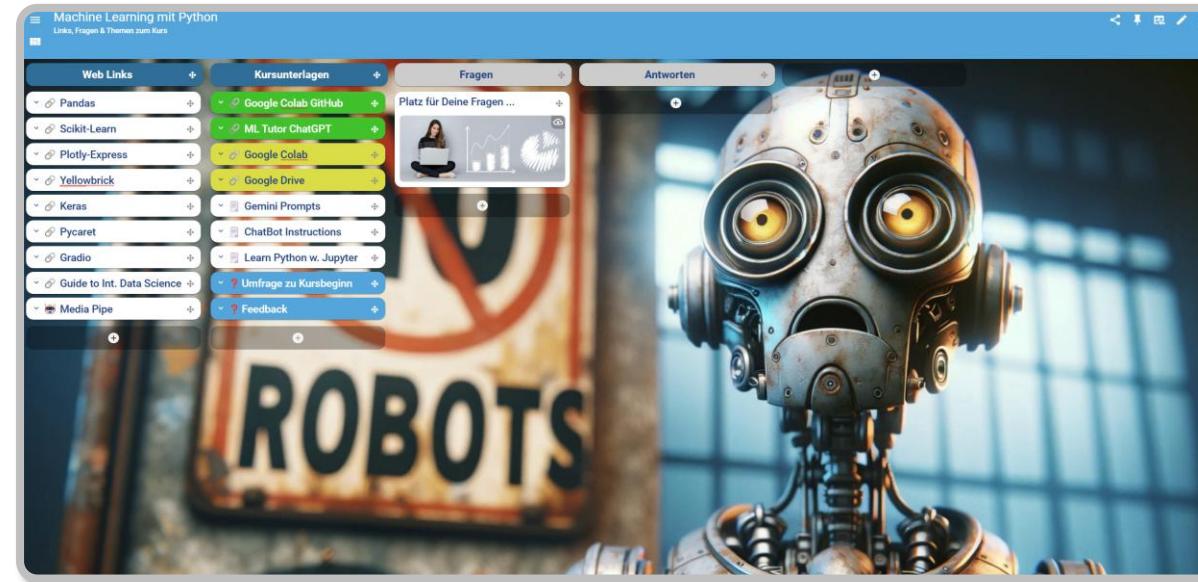
---



# Pinboard Machine Learning

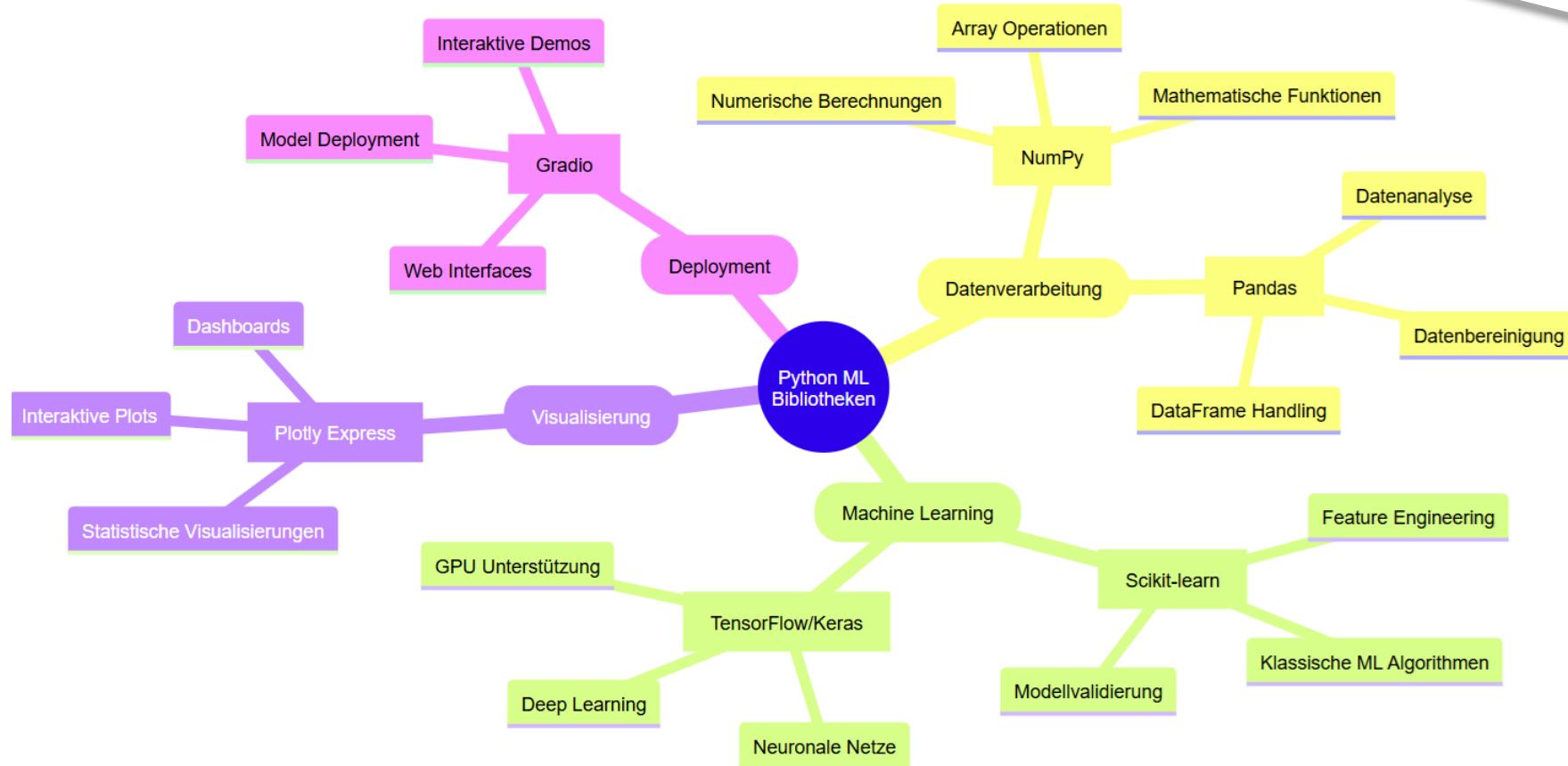
---

**<https://bit.ly/3KwuWaX>**



# Python-ML-Bibliotheken

Ausschnitt



# Machine Learning Basics

---



# Lernen

---



Bild von [StockSnap](#) auf [Pixabay](#)

- Lernen ist ein Prozess, bei dem **Wissen**, Fähigkeiten, Verhaltensweisen oder Einstellungen **erworben**, verändert oder verstärkt werden.
- Es erfolgt durch **Aufnahme**, **Verarbeitung** und **Behalten** von Informationen.
- Es kann auf verschiedene Weise erfolgen, wie durch **Erfahrung**, **Unterricht** oder **Training**.
- Lernen ist ein wesentlicher Bestandteil (menschlicher) Entwicklung und ermöglicht die **Anpassung** an neue Situationen und das Erreichen von Zielen.

# Machine Learning

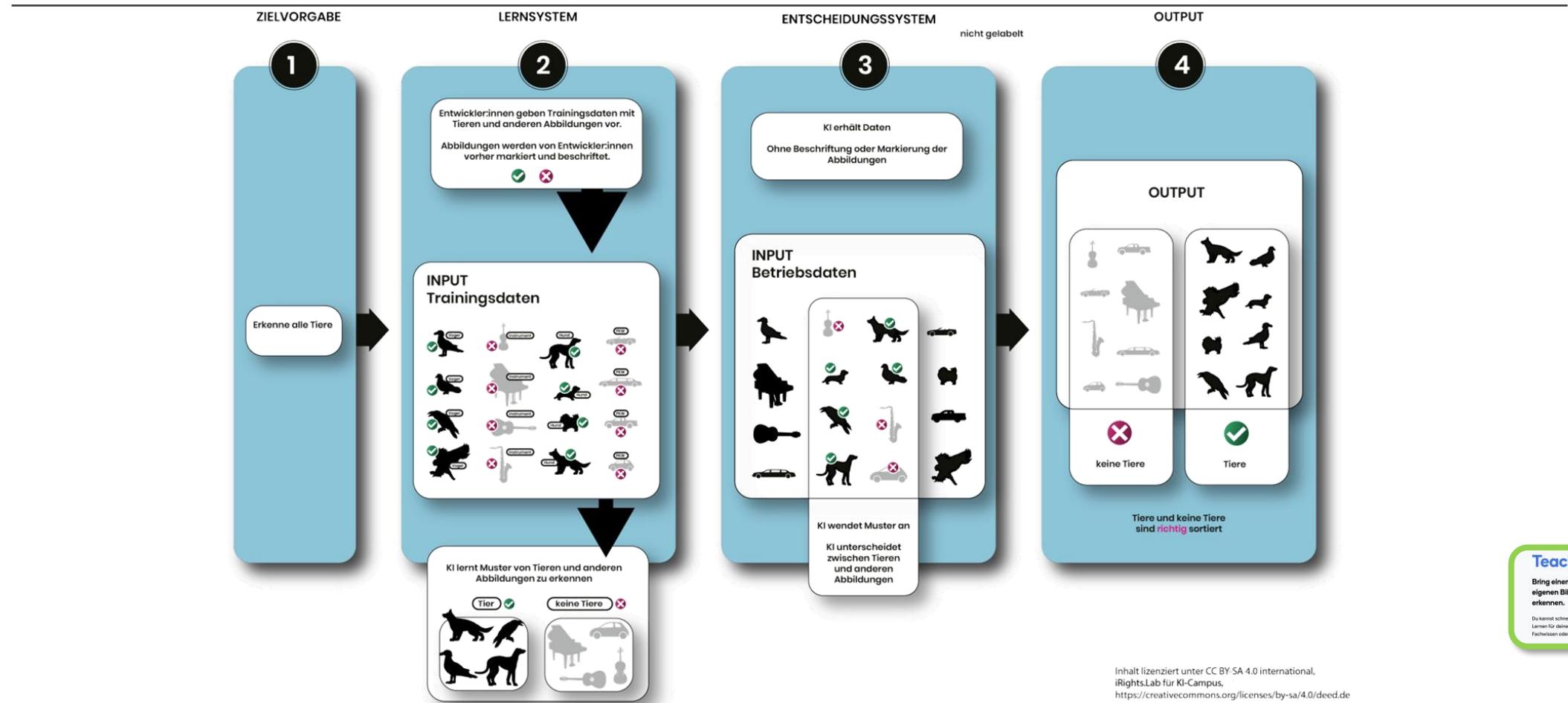
---



Foto von [Andrea De Santis](#) auf [Unsplash](#)

- Machine Learning bezeichnet einen Bereich der künstlichen Intelligenz, der es Computern ermöglicht, **automatisch aus Informationen** und Erfahrung zu **lernen** und ihre Leistung bei bestimmten Aufgaben kontinuierlich zu verbessern.
- Hierbei werden Algorithmen und statistische Modelle verwendet, um aus Daten Muster und **Zusammenhänge** zu **erkennen und** diese in Vorhersagen, Entscheidungen oder Aktionen **umzusetzen**.
- Maschinelles Lernen ist in vielen Anwendungsgebieten wie der Bild- und Spracherkennung, der Datenanalyse, der Prognose und der automatisierten Entscheidungsfindung verbreitet.

# Wie funktioniert Machine Learning?



## Teachable Machine

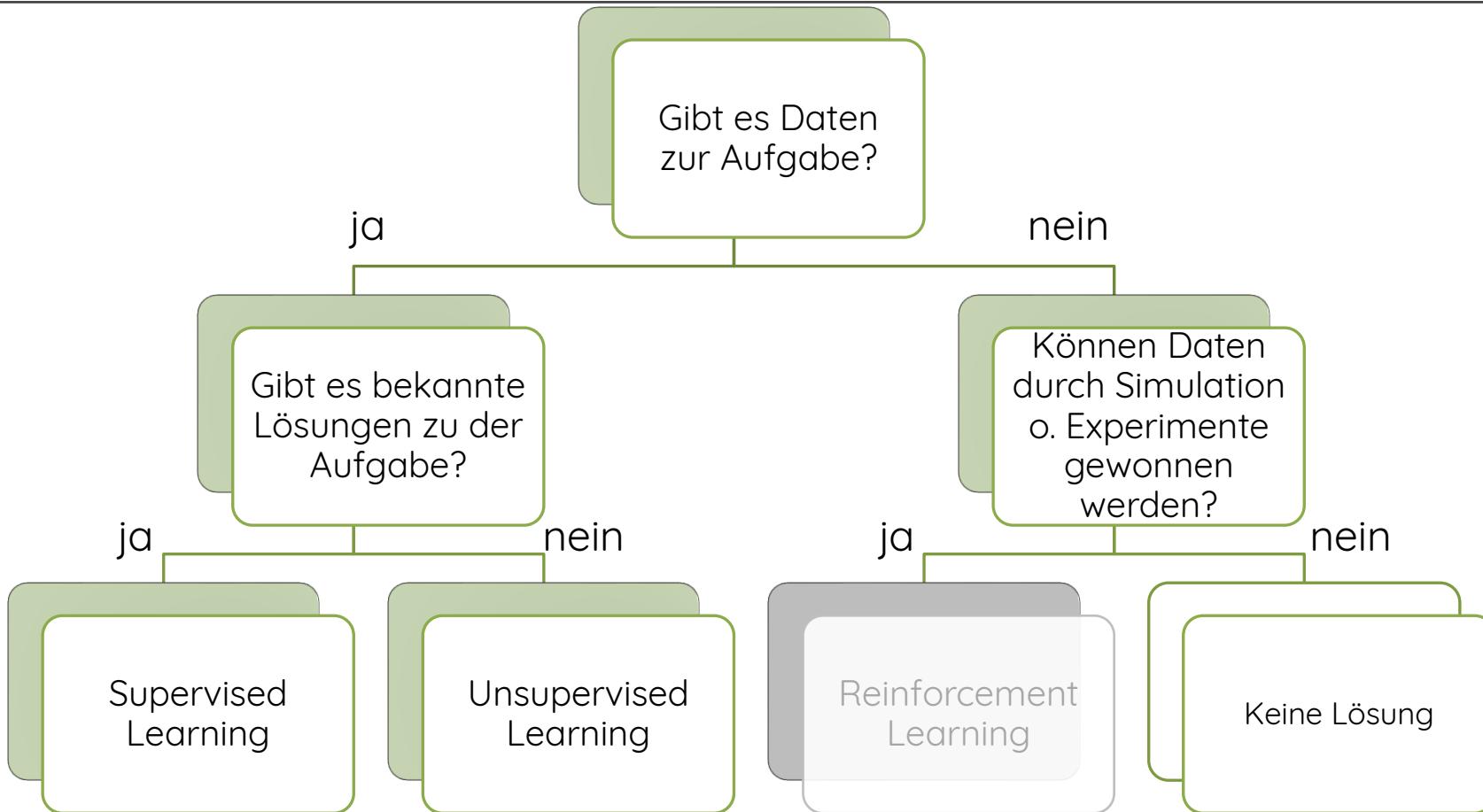
Bring einen Computer bei, deine eigenen Bilder, Töne und Posen zu erkennen.

Du kannst schnell und einfach Modelle für maschinelles Lernen für deine Websites und Apps erstellen – ganz ohne Fachwissen oder Programmierkenntnisse.

Inhalt lizenziert unter CC BY-SA 4.0 international,  
iRightsLab für KI-Campus,  
<https://creativecommons.org/licenses/by-sa/4.0/deed.de>

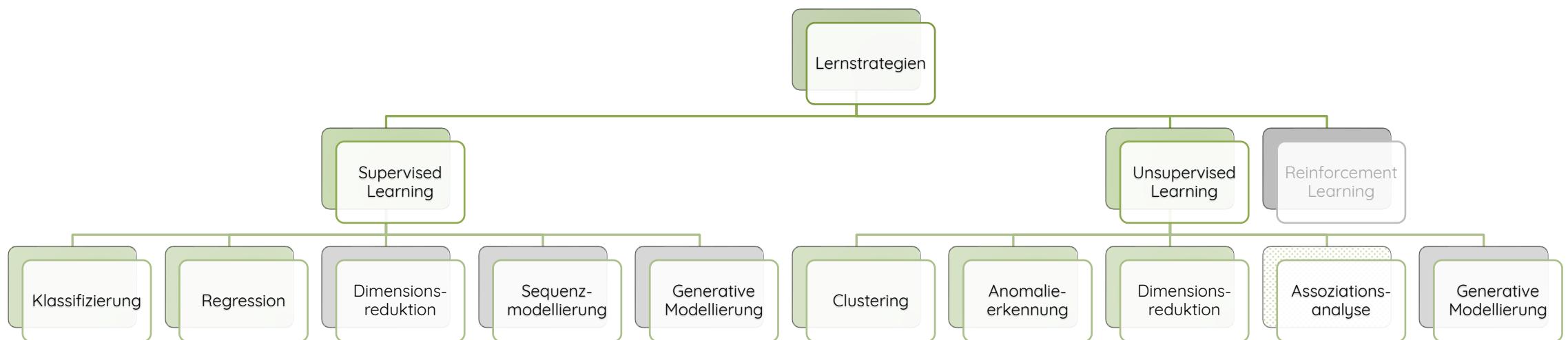
# Lernparadigma & Daten

---



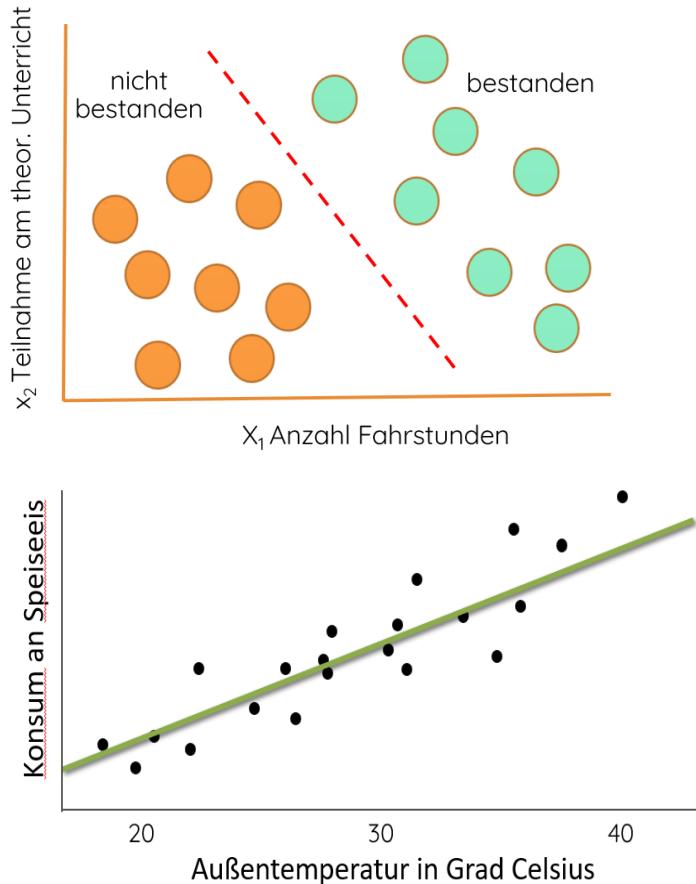
# Lernparadigma & Aufgaben

---



# Supervised Learning

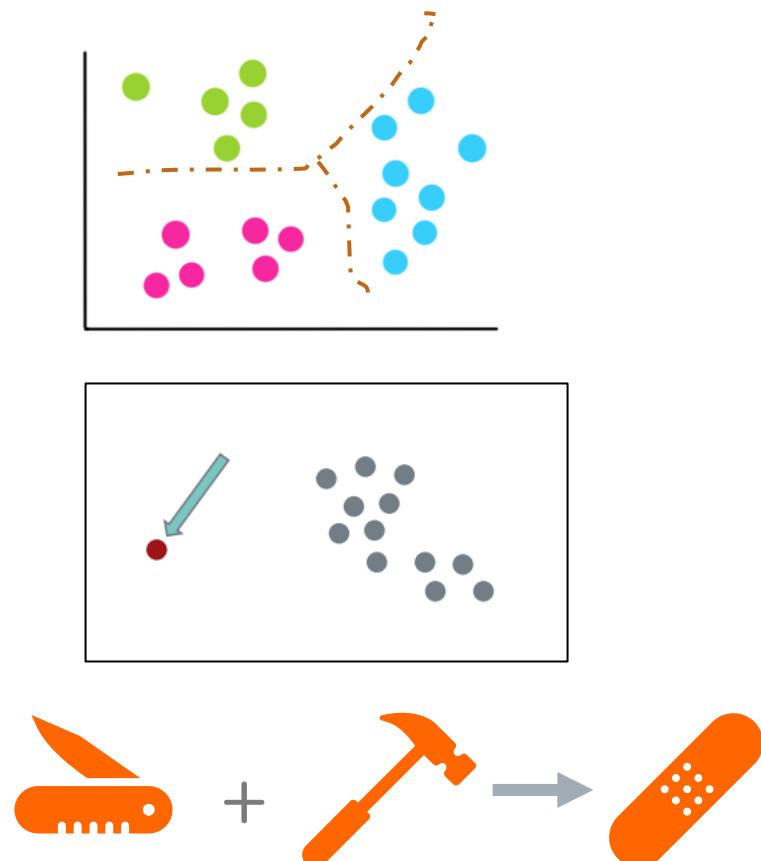
---



- Beim Supervised Learning (überwachtem Lernen) werden i.W. zwei Aufgaben unterschieden:
- **Klassifizierung:** Ein Klassifizierungsmodell sagt **kategorial Werte** voraus.
- Beispiel: Ist eine bestimmte E-Mail-Nachricht Spam oder kein Spam? Besteht ein Fahrschüler seine Prüfung?
- **Regression:** Ein Regressionsmodell sagt stetige, **numerisch Werte** voraus.
- Beispiel: Was ist der Wert einer Immobilie? Wie hoch ist der Konsum an Speiseeis bei einer Außentemperatur von 29°?

# Unsupervised Learning

---



- Das Ziel des unüberwachten Lernens ist es, versteckte Strukturen bzw. Muster in unmarkierten Daten zu finden.
- Im Regelfall werden drei Aufgabengruppen unterschieden:
  - **Clustering/Segmentierung** - Verfahren zur Entdeckung von Ähnlichkeitsstrukturen in (meist relativ großen) Datenbeständen. Die so gefundenen Gruppen von „ähnlichen“ Objekten werden als Cluster bezeichnet.
  - **Anomalierekennung** - soll Datensätze identifizieren, die für die gesamte Datenbasis untypisch sind.
  - **Assoziationsanalyse** - dient dem Auffinden von Zusammenhängen in transaktionsbasierten Datenbasen, die in Form sogenannter Assoziationsregeln dargestellt werden.

# Colab & Jupyter Notebooks

---



# Google Colab(oratory)

---



- Google Colaboratory, kurz Colab, ist eine kostenlose Entwicklungsumgebung, die vollständig in der Cloud ausgeführt wird.
- In Colab können Jupyter-Notebooks erstellt, bearbeiten und ausgeführt werden.
- Colab unterstützt viele beliebte Machine-Learning-Bibliotheken, die einfach in ein Notebook geladen werden können.
- Colab erlaubt es unterschiedliche Laufzeitumgebungen zu definieren in denen man neben einer CPU auch GPUs und TPUs verwenden kann.
- Colab hat mit Gemini eine integriert GenKI für Coding.

CPU = Central Processing Unit, GPU = Graphics Processing Unit, TPU = Tensor Processing Unit



# Jupyter-Notebook

---

- Die Jupyter App ist ein Entwicklungsumgebung, die das Bearbeiten und Ausführen von Programmiersprachen, u.a. Python, über einen Webbrowser ermöglicht.
- Der Name Jupyter bezieht sich auf die drei wesentlichen Programmiersprachen Julia, Python und R und ist auch eine Hommage an Galileos Notizbucheinträge zur Entdeckung der Jupitermonde.
- Mit der Jupyter App kann man Notizbücher erstellen. Diese Notizbücher (Dateiendung .ipynb) enthalten:
  - **Programmcode**, der ausgeführt werden kann,
  - **Markdown Zeilen**, das sind Textzeilen mit Formatierungsangaben.
- Die Jupyter-Notebooks werden v.a. für interaktive, wissenschaftliche Analysen und Berechnungen, z.B. Data Analytics und Machine Learning, verwendet.

# Google Colab - Einstieg

A screenshot of a web browser displaying the Google Colab interface. The title bar shows the URL <https://colab.research.google.com/notebooks/intro.ipynb>. The main content area displays the 'Willkommen bei Colab!' (Welcome to Colab!) page. On the left, there's a sidebar with sections like 'Erste Schritte', 'Data Science', 'Maschinelles Lernen', 'Weitere Ressourcen', and 'Vorgestellte Beispiele'. The main content includes a heading '(Neu) Gemini API testen' with a list of links: 'Generate a Gemini API key', 'Talk to Gemini with the Speech-to-Text API', 'Gemini API: Quickstart with Python', 'Gemini API code sample', 'Compare Gemini with ChatGPT', and 'More notebooks'. Below this is a video thumbnail titled '3 Cool Google Colab Features' with a play button. At the bottom, there's a code cell with the text '[ ] 1 Beginnen Sie mit dem Programmieren oder generieren Sie Code mit KI.' and a question 'Was ist Colab?'. A note at the bottom states: 'Mit Colab oder „Colaboratory“ können Sie Python-Code in Ihrem Brower schreiben und ausführen. Sie können Folgendes tun:' followed by a list: '• Keine Konfiguration erforderlich', '• Kostenloser Zugriff auf GPUs', and '• Einfache Freigabe'. A final note says: 'Egal, ob Sie Student, Data Scientist oder AI-Forscher sind – Colab erleichtert Ihnen die Arbeit. Im Video [Einführung in Colab](#) erhalten Sie weitere Informationen.'

Willkommen bei Colab!

(Neu) Gemini API testen

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

Wenn Sie Colab schon kennen, erfahren Sie in diesem Video mehr zu interaktiven Tabellen, zur Verlaufsansicht für ausgeführten Code und zur Befehlspalette.

3 Cool Google Colab Features

[ ] 1 Beginnen Sie mit dem Programmieren oder generieren Sie Code mit KI.

Was ist Colab?

Mit Colab oder „Colaboratory“ können Sie Python-Code in Ihrem Brower schreiben und ausführen. Sie können Folgendes tun:

- Keine Konfiguration erforderlich
- Kostenloser Zugriff auf GPUs
- Einfache Freigabe

Egal, ob Sie Student, Data Scientist oder AI-Forscher sind – Colab erleichtert Ihnen die Arbeit. Im Video [Einführung in Colab](#) erhalten Sie weitere Informationen.

# Gemini in Colab

---



## ■ Gemini Zeile



1 Beginnen Sie mit dem Programmieren oder generieren Sie Code mit KI.

- Code generieren direkt im Jupyter Notebook
- Code-Vervollständigung

## ■ Gemini Register



- Code optimieren
- Code kommentieren/dokumentieren
- Code erklären/erläutern
- allgemeine Prompts

# Tastatureinstellungen/Shortcuts

## Tastatureinstellungen

Editor-Tastaturbelegungen  
default  
 Mit der Eingabetaste werden Vorschläge akzeptiert

### Tastenkombinationen

Wenn Sie eine Tastenkombination hinzufügen oder ändern möchten, klicken Sie auf die entsprechende Tastenkombination und geben Sie dann die neue Kombination ein. Beachten Sie, dass Ctrl+M als Präfix für Tastenkombinationen mit mehreren Tasten verwendet werden kann.

Tastenkombination ipynb herunterladen

### Standardeinstellungen wiederherstellen

Ctrl+/ Aktuelle Zeile kommentieren

Tastenkombination py herunterladen

Alle Ausgaben löschen

Tastenkombination Alle Laufzeiten auf Werkseinstellungen zurücksetzen

Ctrl+Shift+A Alle Zellen auswählen

Ctrl+F9 Alle Zellen in Notebook ausführen

Ctrl+H Alle suchen/ersetzen

Ctrl+[ Alle/ausgewählte Abschnitte maximieren

Ctrl+] Alle/ausgewählte Abschnitte minimieren

Ctrl+M - An Cursorposition teilen

Tastenkombination Ansicht mit einem Tab ansehen

Tastenkombination Auf die zuletzt ausgeführte Zelle fokussieren

Ctrl+I Ausführung unterbrechen

Ctrl+M O Ausgabe einblenden/ausblenden

Tastenkombination Ausgabevollbild anzeigen

### Standardeinstellungen wiederherstellen

Ctrl+M . Laufzeit neu starten

Tastenkombination Laufzeit neu starten und alle Zellen im Notebook ausführen

Tastenkombination Laufzeitprotokolle anzeigen

Tastenkombination Layout des minimierten Abschnitts speichern

Ctrl+M Z oder Ctrl+Shift+Z Letzte Zellaktion rückgängig machen

Tastenkombination Linken Bereich in einen Tab verschieben

Ctrl+\* Maximieren/Minimieren des aktuellen Abschnitts aktivieren/deaktivieren

Tastenkombination Maximize/Minimize current cell

Shift+Click Mehrere Zellen auswählen

Tastenkombination Mit einer Laufzeit verbinden

Tastenkombination Mit einer benutzerdefinierten GCE-VM verbinden

Tastenkombination Mit lokaler Laufzeit verbinden

Tastenkombination Neues Notebook erstellen

Tastenkombination Ausgabevollbild anzeigen

## Tastatureinstellungen

### Tastenkombinationen

### Ausgewählte Ausgaben löschen

Ctrl+M K Ausgewählte Zellen nach oben verschieben

Ctrl+M J Ausgewählte Zellen nach unten verschieben

Ctrl+F10 Ausgewählte Zellen und alle folgenden Zellen ausführen

Ctrl+Shift+Enter Ausgewählte Zellen zusammenführen

Ctrl+Shift+Enter Auswahl ausführen

Shift+Down Auswahl erweitern, um nächste Zelle einzuschließen

Shift+Up Auswahl erweitern, um vorherige Zelle einzuschließen

Tastenkombination Automatische Ausführung der ersten Zelle oder des ersten Abschnitts bei jeder Ausführung aktivieren

Tastenkombination Automatische Vervollständigung

Ctrl+Space oder Tab Automatische Vervollständigung

Ctrl+Shift+P Befehlspalette anzeigen

Tastenkombination Code ein-/ausblenden

Ctrl+Alt+P Code-Snippets-Bereich anzeigen

Tab Code-docstring-Hilfe aktivieren/deaktivieren

?

Tastenkombination Codezelle hinzufügen

Ctrl+M A Codezelle oben einfügen

Ctrl+M B Codezelle unten einfügen

Tastenkombination Colab Enterprise öffnen

## Tastatureinstellungen

### Tastenkombinationen

### Notebook freigeben

Tastenkombination Notebook hochladen

Tastenkombination Notebook in Drive verschieben

Tastenkombination Notebook in Google Drive markieren/Markierung aufheben

Tastenkombination Notebook in Papierkorb verschieben

Ctrl+S Notebook speichern

Ctrl+O Notebook öffnen

Tastenkombination Notebook-Einstellungen öffnen

Tastenkombination Notebook-Quelle anzeigen

Tastenkombination Notebook-Vergleich

Ctrl+M N Nächste Zelle

Ctrl+Shift+] Nächsten Tab hervorheben

Ctrl+G Nächstes Element suchen

Tastenkombination Ressourcen ansehen

Ctrl+Alt+N Scratchpad-Codezelle öffnen

Tastenkombination Seitenleiste für Kommentare

Tastenkombination Sichtbarkeit des Headers aktivieren/deaktivieren

Tastenkombination Sitzungen verwalten

Tastenkombination Statusleiste einblenden/ausblenden

Tastenkombination Tab in den nächsten Bereich verschieben

Tastenkombination Tab in den vorherigen Bereich verschieben

Tastenkombination Tabs in zwei Spalten anzeigen

## Tastatureinstellungen

### Tastenkombinationen

### Dateibrowser anzeigen

Tastenkombination Drive bereitstellen

Tastenkombination Drive trennen

Tastenkombination Editor-Einstellungen öffnen

Einstellungen öffnen

Shift+Tab Einzug für aktuelle Zeile entfernen

Tastenkombination Fokus mit Tabulatoraste verschieben

aktivieren/deaktivieren

Tastenkombination Formular hinzufügen

Ctrl+M F Formularansicht ändern

Tastenkombination Formularattribute bearbeiten

Tastenkombination Formularfeld hinzufügen

Tastenkombination Geplante Notebooks anzeigen

Ctrl+Enter Hervorgehobene Zelle ausführen

Ctrl+Shift+S Hervorgehobene Zelle auswählen

Tastenkombination Hervorgehobene Zelle mit nächster Zelle zusammenführen

Tastenkombination Hervorgehobene Zelle mit vorheriger Zelle zusammenführen

Esc Hervorgehobener Zelle aufheben

Tastenkombination Im Playground-Modus öffnen

Tastenkombination In Google Drive suchen

Tastenkombination In Scratchpad-Zelle kopieren

Shift+Ctrl+H In aktueller Zelle alle ersetzen

Tastenkombination Informationen zu Notebook-Datei anzeigen

Tastenkombination Inhaltsverzeichnis anzeigen

## Tastatureinstellungen

### Tastenkombinationen

### Tabs in zwei Zeilen anzeigen

Tastenkombination Tabs in zwei Zeilen anzeigen

Ctrl+M H Tastenkombinationen anzeigen

Tastenkombination Terminal anzeigen

Tastenkombination Textzeile hinzufügen

Tastenkombination Variablenübersicht anzeigen

Tastenkombination Verbindung mit gehosteter Laufzeit herstellen

Tastenkombination Verbindung trennen und Laufzeit löschen

Tastenkombination Verlauf der Codeausführung anzeigen

Ctrl+M P Vorherige Zelle

Ctrl+Shift+[ Vorherigen Tab hervorheben

Ctrl+Shift+G Vorheriges Element suchen

Tastenkombination Wieder verbinden

Ctrl+M L Zeilennummern aktivieren/deaktivieren

Ctrl+Shift+Y Zellaktion wiederholen

Ctrl+Click Zellauswahl aktivieren/deaktivieren

Alt+Enter Zelle ausführen und neue Zelle einfügen

Shift+Enter Zelle ausführen und nächste Zelle auswählen

Tastenkombination Zelle in Tab spiegeln

Tastenkombination Zelle mit Abschnittsüberschrift hinzufügen

Tastenkombination Zelle oder Auswahl ausschneiden

## Tastatureinstellungen

### Tastenkombinationen

### JSON-Notebook anzeigen

Tastenkombination JSON-Notebook anzeigen

Ctrl+Alt+M Kommentar hinzufügen

Tastenkombination Kommentarbereich öffnen

Tastenkombination Kopie als GitHub Gist speichern

Tastenkombination Kopie in Drive speichern

Tastenkombination Kopie in GitHub speichern

Ctrl+M Y Zu Codezelle konvertieren

Ctrl+M M Zu Textzelle konvertieren

Tastenkombination Zu einer konkreten Zelle

Ctrl+M S Überarbeitung speichern und anpinnen

Tastenkombination Überarbeitungsverlauf anzeigen

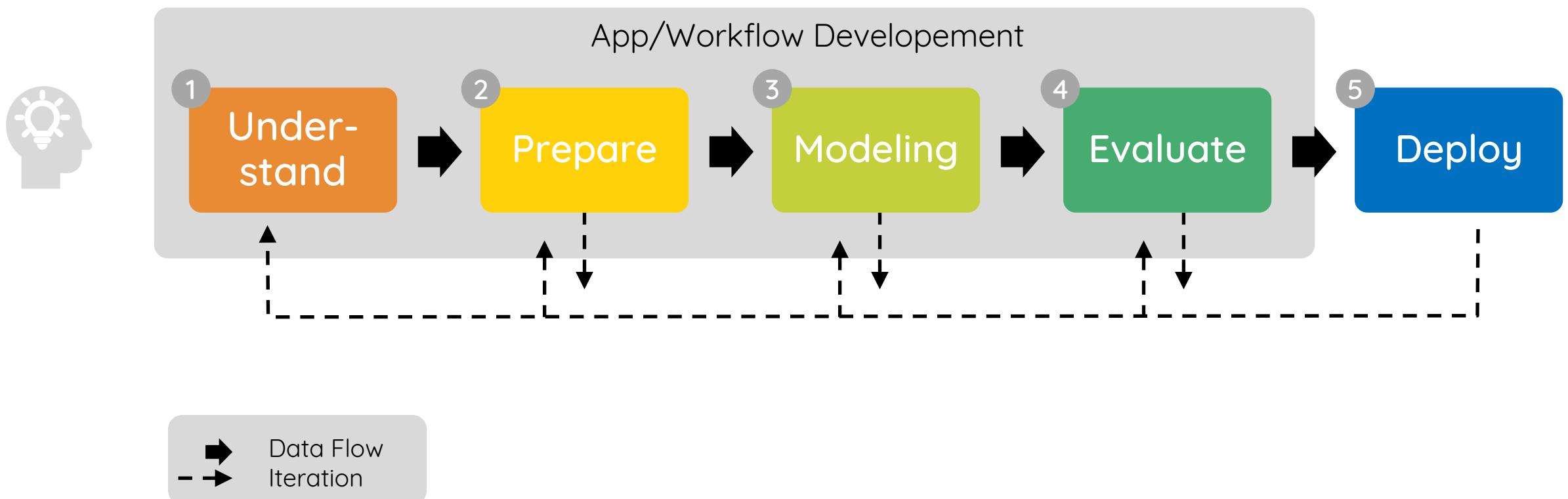
# Machine Learning Process

---



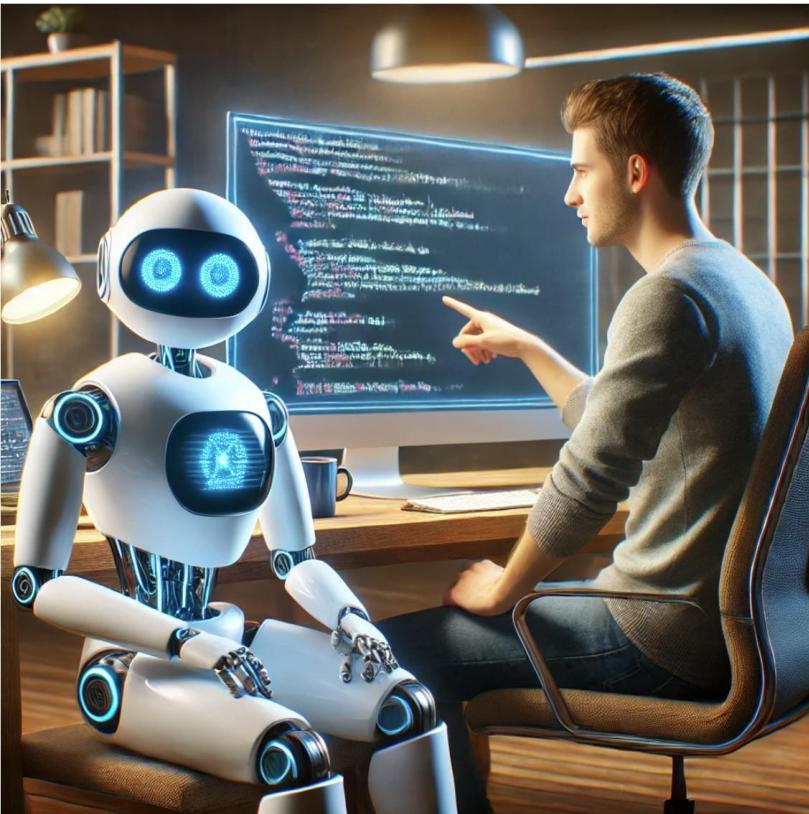
# Intro ML Process

# Machine Learning Process



Setzen  
wir ein

# Chatbot als Supporter



Ein Chatbot kann beim Coden unterstützend eingesetzt werden, wenn es darum geht, schnelle Antworten auf Fragen zu Syntax, Fehlerbehebungen oder Best Practices zu erhalten.

- Code-Erklärungen: Verständnis von Funktionen, Klassen oder Algorithmen.
- Bug-Fixes: Hilfe bei der Fehlersuche und -behebung.
- Code-Generierung: Vorschläge für Code-Snippets oder erste Entwürfe.
- Optimierung: Empfehlungen zur Verbesserung von Codequalität und Effizienz.
- Lernunterstützung: Erklären von Konzepten und Schritt-für-Schritt-Anleitungen.

Bild mit DALL·E erstellt

# Checklisten

---

Setzen  
wir ein

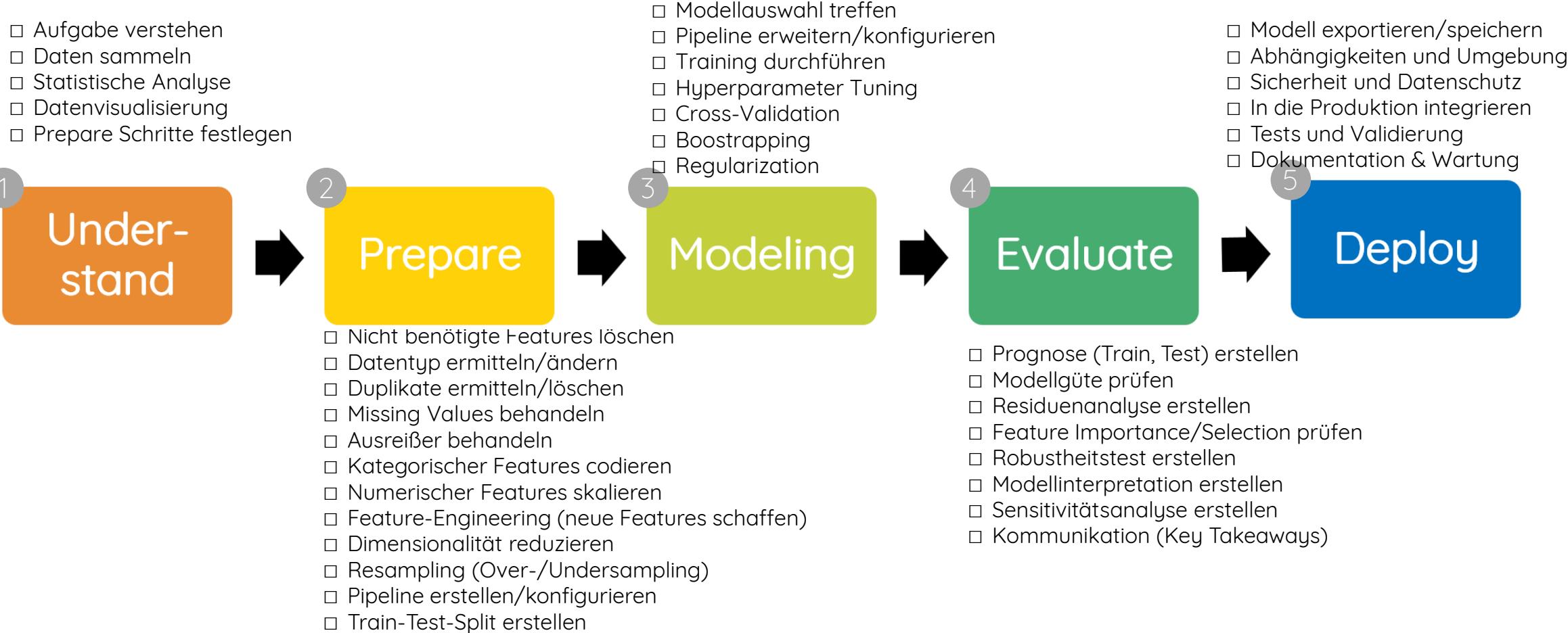


Bild von [Gerry](#) auf [Pixabay](#)

In der Fliegerei werden Checklisten verwendet, dass wichtige Schritte und Verfahren ordnungsgemäß durchgeführt werden und keine wichtigen Details übersehen werden.

- **Sicherheit:** Checklisten helfen dabei, sicherheitsrelevante Aufgaben und Verfahren zu standardisieren und sicherzustellen, dass sie in einer sicheren und konsistenten Art und Weise durchgeführt werden.
- **Komplexität:** Es gibt eine Vielzahl von Systemen, Verfahren und Protokollen, die beachtet werden müssen. Checklisten helfen den Piloten und dem Flugpersonal dabei, diese Komplexität zu bewältigen und nichts zu vergessen.
- **Standardisierung:** Checklisten ermöglichen eine Standardisierung der Abläufe und Verfahren. Dies ist besonders wichtig, wenn mehrere Piloten oder Crewmitglieder zusammenarbeiten, dass alle nach dem gleichen Prozess arbeiten und keine wichtigen Schritte auslassen.

# Checkliste ML Process



# Code-Snippets



Bild von [Hans](#) auf Pixabay

- Code-Snippets sind vorgefertigte Codefragmente, die in der Programmierung wiederverwendet werden können.
- Sie werden in verschiedenen IDEs (Integrated Development Environments) eingesetzt, um Zeit zu sparen und die Effizienz zu steigern.
- Einsatzmöglichkeiten u.a.:
  - Wiederverwendung von Code: Häufig wiederkehrende Codeblöcke, die in verschiedenen Projekten oder sogar innerhalb desselben Projekts verwendet werden sollen.
  - Schnelle Implementierung: Code-Snippets können helfen, häufig verwendete Funktionen oder Algorithmen schnell zu implementieren.
  - ...

# Understand

---



# Understand

---



- Das Einlesen und Analysieren der Daten ist der erste Schritt der Datenanalyse.
- Erste Erkenntnisse zu den Daten: Features, Labels sowie darin erkennbaren Mustern werden gewonnen.
- Hierbei können erste, einfache statistische Analysen hilfreich sein (Min, Max, Mittelwert, Korrelation, Standardabweichung, Varianz, ...).
- Mithilfe interaktiver Ansichten und Visuals kann das Gesamtbild weiter vervollständigt werden.
- Aus der Exploration ergeben sich erste Anhaltspunkte für den Folgeprozess der Datenaufbereitung.

# Prepare

---



We usually  
have to play  
with what's on  
the right

---

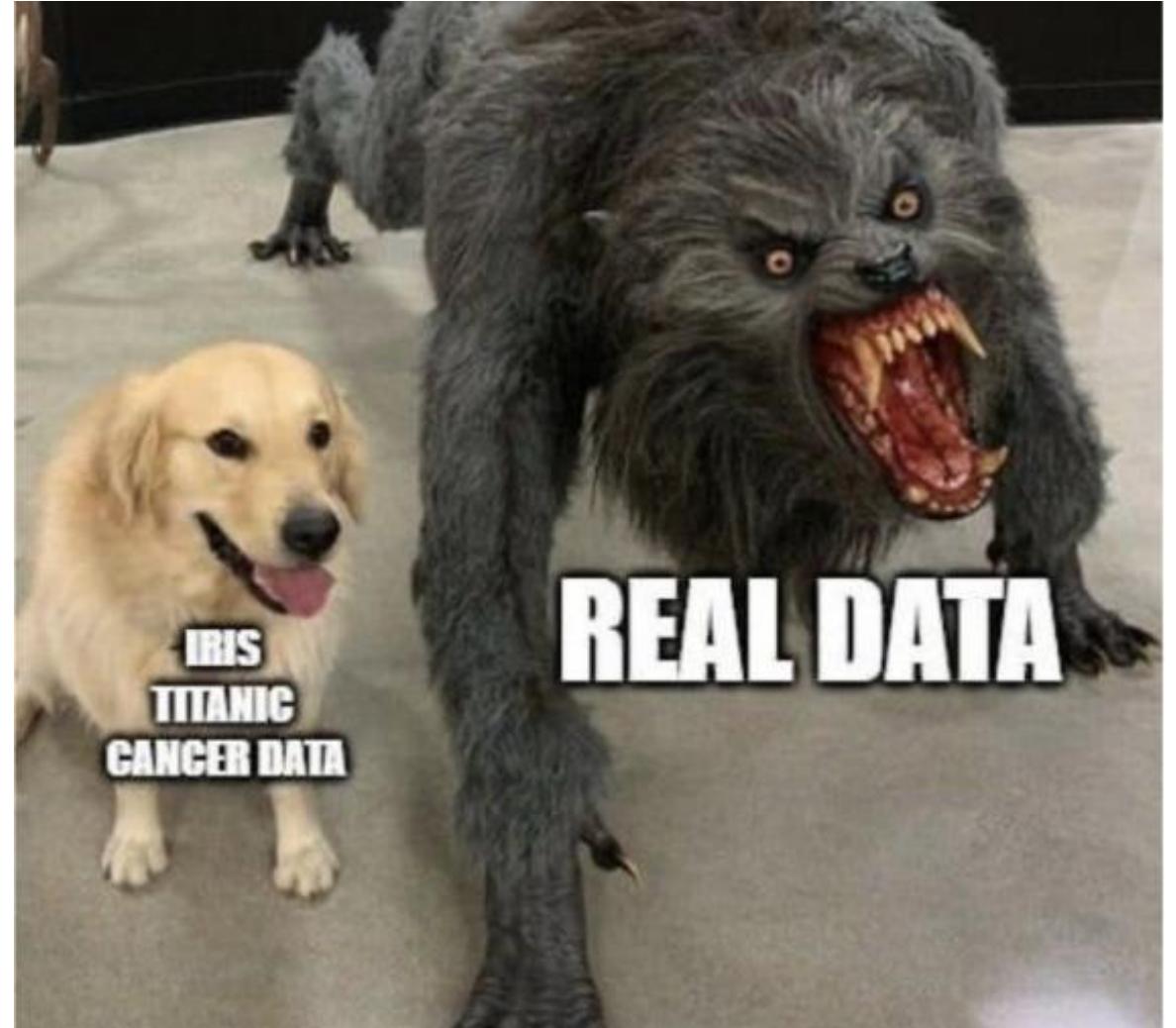
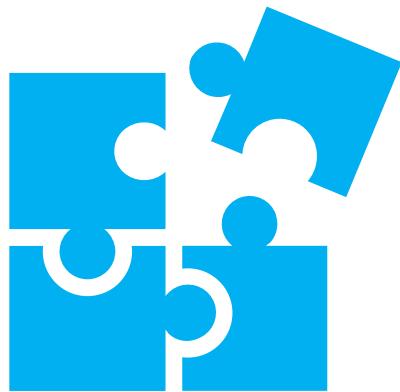


Bild-Quelle: [Clean AutoML for “Dirty” Data. How AutoML helps to work efficiently... | by Mikhail Sarafanov | Towards Data Science \(medium.com\)](#)

# Prepare

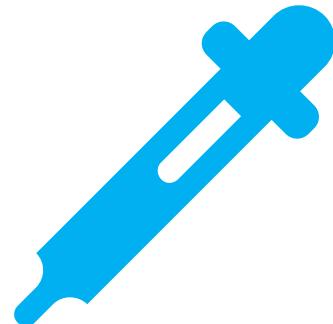
---



- Die Datenvorverarbeitung beim maschinellen Lernen ist ein entscheidender Schritt, die Datenqualität zu erkennen und zu verbessern
- Die Datenvorverarbeitung umfasst alle Schritte der Vorbereitung der Rohdaten für das anschließende Training von Modellen.
- Die wesentlichen Schritte sind:
  - Identifikation von fehlenden Werten/Dubletten
  - Identifikation von Ausreißern, unplausiblen Daten
  - Codierung & Skalierung
  - Umgang mit Ungleichgewichten

# Missing Values

---

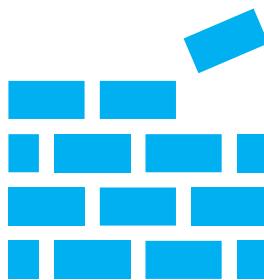


- Löschen von Zeilen und/oder Spalten mit fehlenden Werten:
  - Entfernen der Zeilen und/oder Spalten aus dem Dataset.
  - Konsequenz: Weniger Trainingsdatenproben und/oder weniger Features können zu Overfitting/Underfitting führen
- Ausfüllen (Imputation) der fehlenden Werte:
  - Ersetzen durch den Durchschnittswert in der Spalte
  - Ersetzen durch den gebräuchlichsten Wert für diese Spalte
  - Zuweisen eines gemeinsamen Werts (Platzhalters) für fehlenden Werte.
- Erweitertes Ausfüllen (Imputation):
  - Prognostizieren von fehlenden Werten aus Stichproben.

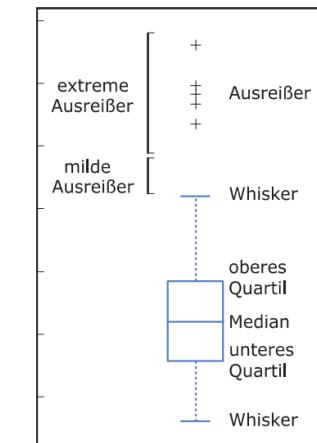


# Outlier

---



- Ein Ausreißer ist ein Datenpunkt (Outlier), dessen Ausprägung stark von der Norm abweicht.
- Um die Auswirkungen von Ausreißern auf die Analysen oder Modellbildungen zu reduzieren, müssen Ausreißer erkannt ggf. bereinigt werden.
- Vorgehensweisen zur Bereinigung können analog „fehlender Daten“ gewählt werden.
- Identifikation von Ausreißern:
  - wissensbasiert
  - statistikbasiert
- Kategorien von Ausreißern:

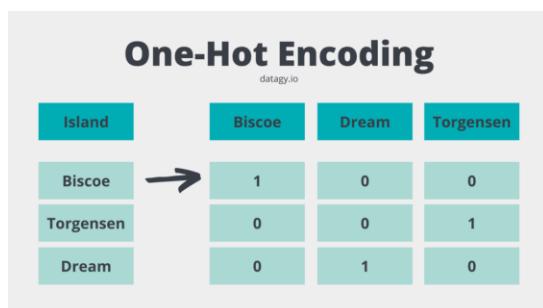


- [KNMIME Data Preprocessing for Machine Learning Part 1](#)
- [KNIME Advanced ETL Functionalities and Machine Learning](#)

# Coding (1/2)

---

Encoder	
primary	1
middle	2
high	3
college	4
university	5



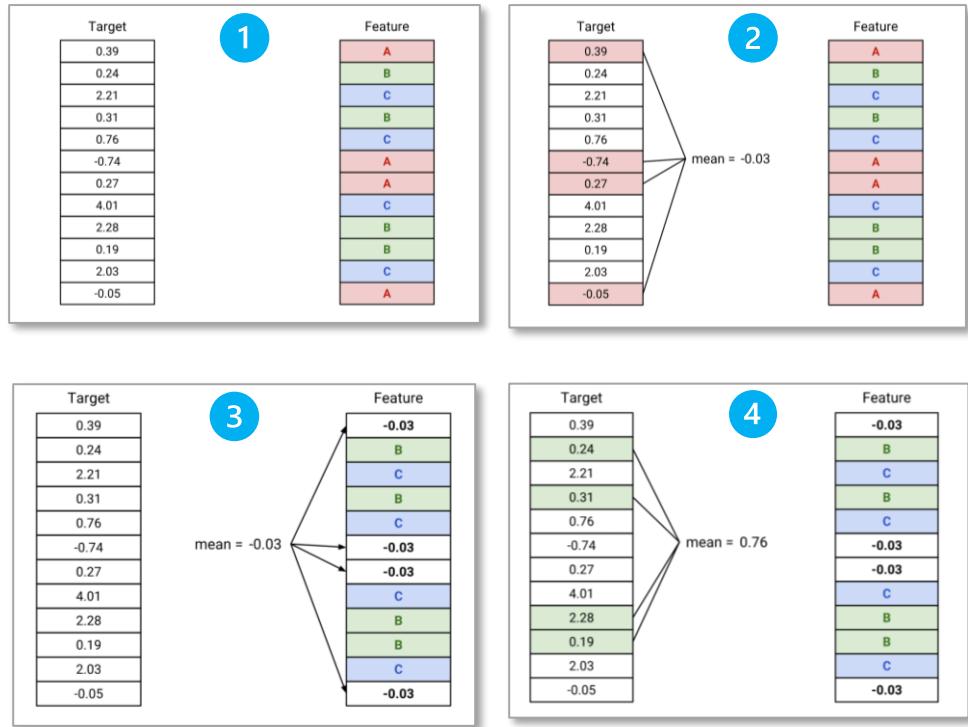
- ML-Algorithmen, die numerische Daten benötigen, brauchen ein Vorverarbeitung, die diskrete/ kategoriale Daten (z. B. Nominaldaten) in numerische Daten umgewandelt.
- Verschiedene Ansätze (Auswahl) können dies leisten:
  - **OrdinalEncoder (ordinal):** Kategorialen (diskreten) Merkmale werden in ordinale Ganzzahlen konvertiert. Dies führt zu einem Array mit ganzen Zahlen (0 bis n-categories - 1) pro Feature.
  - **OneHotEncoder (nominal):** Erstellt eine binäre Spalte für jede Kategorie und gibt eine Sparse-Matrix zurück.

Quelle: [One-Hot Encoding in Scikit-Learn with OneHotEncoder • datagu](#)

 [KNIME Data Preprocessing for Machine Learning Part 1](#)

# Coding (2/2)

---



- TargetEncoder (nominal):  
Der TargetEncoder ersetzt kategoriale Werte durch eine Zahl, die unter Berücksichtigung des Zielmerkmals (target) für diese Kategorie ermittelt wird.
- Bewertung der nominalen Encoder
  - OneHotEncoder: Besser für Daten mit wenigen Kategorien; vermeidet die Einführung einer ordinalen Beziehung.
  - TargetEncoder: Effektiver bei vielen Kategorien; bezieht Informationen über das Zielmerkmal ein, birgt aber Risiken wie Overfitting und Datenlecks.

Quelle: [Representing Categorical Data with Target Encoding | Brendan Hasz](#)

 [KNIME Data Preprocessing for Machine Learning Part 1](#)

# Skalierung (= Normalisierung & Standardisierung)

---

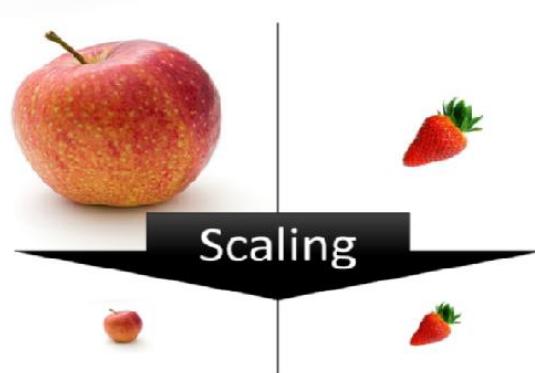


Bild-Quelle: [What, When and Why Feature Scaling for Machine Learning | by Mayank Gupta | TechnoFunnel | Medium](#)

- Eine Skalierung kann für eine korrekte Vorhersagen erforderlich sein.
- Falls die Werte einer Spalte im Vergleich zu anderen sehr hoch sind, ist die Auswirkung der Spalte mit dem höheren Wert viel höher als die, der niedrigwertiger Spalten.  
Beispiel: Alter vs. Einkommen
- Daher liefert die Vorhersage möglicherweise nicht die erwarteten Ergebnisse.
- Effekte der Skalierung:
  - Die Höhe der Regressionskoeffizienten werden direkt von der Skalierung der Features beeinflusst.
  - Einige der Algorithmen verkürzen die Ausführungszeit, wenn sie skaliert würden.
  - Verbesserung der Prognose-Ergebnisse durch Skalierung.

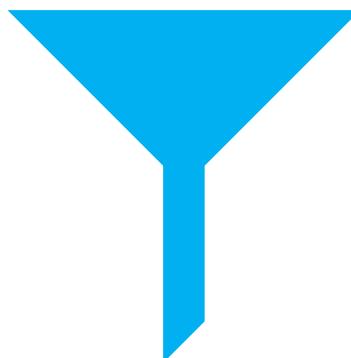
 [KNIME Data Preprocessing for Machine Learning Part 1](#)

# Normalisierung vs Standardisierung

Kriterium	Normalisierung (Min-Max)	Standardisierung
Definition	Skalierung der Werte in einen festen Bereich, üblicherweise [0,1]	Verschiebung der Verteilung auf Mittelwert = 0 und Standardabweichung = 1
Formel	$x_{\text{norm}} = (x - \min(x)) / (\max(x) - \min(x))$	$x_{\text{std}} = (x - \mu) / \sigma$
Vorteile	<ul style="list-style-type: none"><li>Einfache Interpretierbarkeit, da alle Werte zwischen 0 und 1 liegen</li><li>Sinnvoll, wenn ein definierter Wertebereich (z.B. [0,1] oder [-1,1]) gewünscht wird</li></ul>	<ul style="list-style-type: none"><li>Besserer Umgang mit Ausreißern</li><li>Häufig für lineare Modelle und neuronale Netze empfohlen</li></ul>
Nachteile	Sehr anfällig für Ausreißer: Ein extrem großer oder kleiner Wert beeinflusst das ganze Feature stark	Daten liegen nicht mehr in [0,1], was für manche Modelle oder Visualisierungen weniger intuitiv sein kann
Typische Verwendung	<ul style="list-style-type: none"><li>Distanzbasierte Algorithmen (k-NN, k-Means)</li><li>Neuronale Netze mit normalisierten Eingaben (z.B. Pixelwerte)</li></ul>	<ul style="list-style-type: none"><li>Lineare Modelle (Lineare/Logistische Regression)</li><li>SVM</li><li>Neuronale Netze</li><li>Standard für einheitliche Feature-Skalierung</li></ul>
Empfehlung	Verwenden, wenn alle Features den gleichen Bereich benötigen	Standardmäßig wählen, wenn nicht klar ist, welche Skalierung sinnvoll ist, oder wenn Modelle stark auf einheitliche Skalen angewiesen sind

# Dimension Reduction

---



- Eine (extrem) große Anzahl von Merkmalen/Features kann den Modellaufbau, den Modelleinsatz sowie die Modellinterpretation erschweren.
- Bei der Dimensionsreduktion werden nicht informative Merkmale/Features entfernt und damit das Datenmodell vereinfacht.
- Prinzipielle Vorgehensweisen:
  - Analyse & Ausschluss einzelner Merkmale/Features
  - Transformation von Merkmalen/Features  
(siehe Algorithmen & Modelle)

# Analyse & Ausschluss

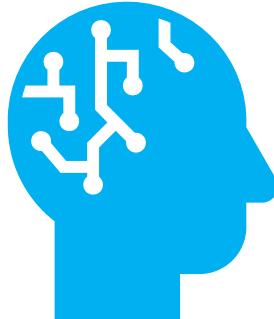
---

Ausschluss von Merkmalen mit ...:

- geringer Varianz  $\sigma^2$   
(Streuung von Werten um ihren Mittelwert, mittlere quadratische Abweichung)
- konstanten Werten
- fehlenden Werten
- hoher Korrelation

# Feature Engineering

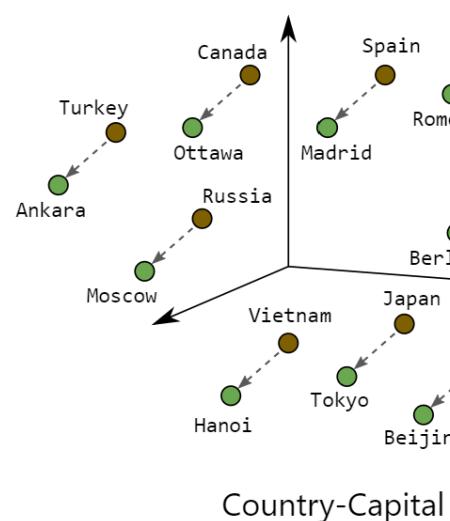
---



- Merkmalsgestaltung / Feature-Engineering verwendet Domänen- und Datenkenntnisse, um aus den bereitgestellten Rohdaten neuartige oder reduzierte Features als Eingaben für ML-Modelle zu erstellen.
- Ziel: Reduzierung Trainingszeit, bessere Interpretierbarkeit
- Das Training eines ML-Modells wird optimiert, wenn nur relevante, aussagekräftige und korrekte Attribute verwendet werden.
- Alle ungenutzten, unnötigen und redundanten Attribute werden entfernt. Typische Entfernung sind Features, die im Falle einer Operationalisierung nicht verfügbar oder hoch korrelative sind.
- Folgende - sich tw. überschneidende - Ansätze stehen zur Verfügung:
  - Merkmalskonstruktion (Multiplikation, Quadrierung usw.)
  - Merkmalsextraktion (Kodierung, Vektorisierung)
  - Merkmalsauswahl (Dimensionsreduktion)

# Merkmalskonstruktion

---



Quelle: [Embeddings: Translating to a Lower-Dimensional Space | Machine Learning Crash Course | Google Developers](#)

- Merkmalskonstruktionen sind vereinfacht Kombinationen aus zwei oder mehreren Features, um einen inhaltlichen Mehrwert zu schaffen.
- Merkmalskonstruktion durch arithmetisch/logischer Verknüpfung (Feature Crossing):
  - Multiplikation, Division, Summen, ... von Merkmalen.  
Beispiel: Customer-Live-Time-Value
  - Logische Verknüpfung  
Beispiel: Land & Hauptstadt
- Dummy Coding / Dummy Merkmale:
  - Merkmale, die Kategorien oder Merkmale ersetzen.
  - Beispiel: Ausprägungsreduktion Wohnorte zu einem Merkmal „Raum“ mit Ausprägung: Stadt, Umland, ländlicher Raum

# Resampling

---



Die Anzahl der Stichproben pro Klasse ist nicht gleichmäßig verteilt. Ein ML-Modell funktioniert möglicherweise nicht gut für die seltenen Klassen.

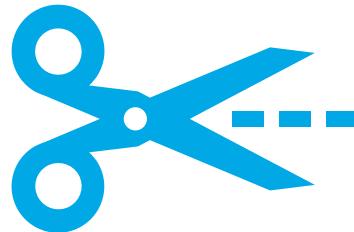
**Kernfrage:** Liegt es an den Daten oder an der Realität?

Strategien zur Handhabung:

- Analyse des Sachverhalts, z.B. anhand der Confusion Matrix.
- Down-Sampling: Reduktion die Zahl in der dominierenden Klasse.
- Up-Sampling: Erhöhen Zahl in der seltenen Klasse.
- Data Generation: Erstellen neuer Datensätze, ähnlich, aber nicht identisch.
- Sample Weights: seltenen Klassen erhalten höhere Gewichtungen und dominanten Klassen niedrigere Gewichtungen.
- Das Resampling erfolgt nur für die Trainingsdaten

# Train/Test-Split

---



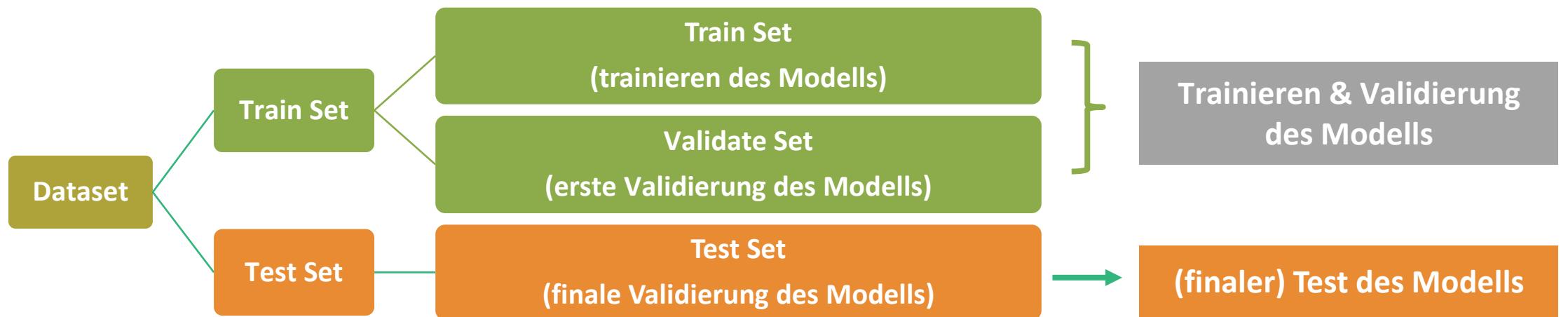
- Das Train-Test-Split-Verfahren wird verwendet, um die Leistung von ML-Algorithmen zu überprüfen.
- Mit den Trainingsdaten wird das Modell trainiert Vorhersagen zu treffen.
- Mit den Testdaten wird die Leistungsfähigkeit des Modells überprüft.
- Die Testdaten sind Daten, die das Modell vorher noch nicht gesehen hat.
- Es ist ein einfach anzuwendendes Verfahren, um das Modell zu evaluieren.
- Gängige Trennquoten Train/Test sind: 80-20, 70-30, 60-40



[KNIME Data Preprocessing for Machine Learning Part 1 & Part 2](#)

# Train-Validate-Test-Set

---



# Modeling

---



Die Masse an  
Auswahl  
garantiert nicht  
das Finden.

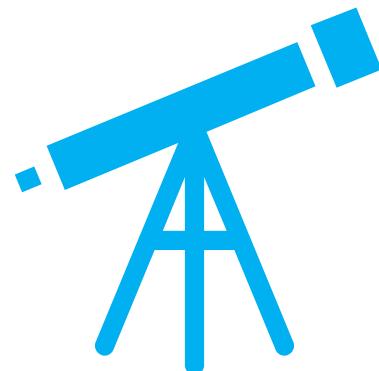
---

Damaris Wieser



Bild von [Nino Carè](#) auf [Pixabay](#)

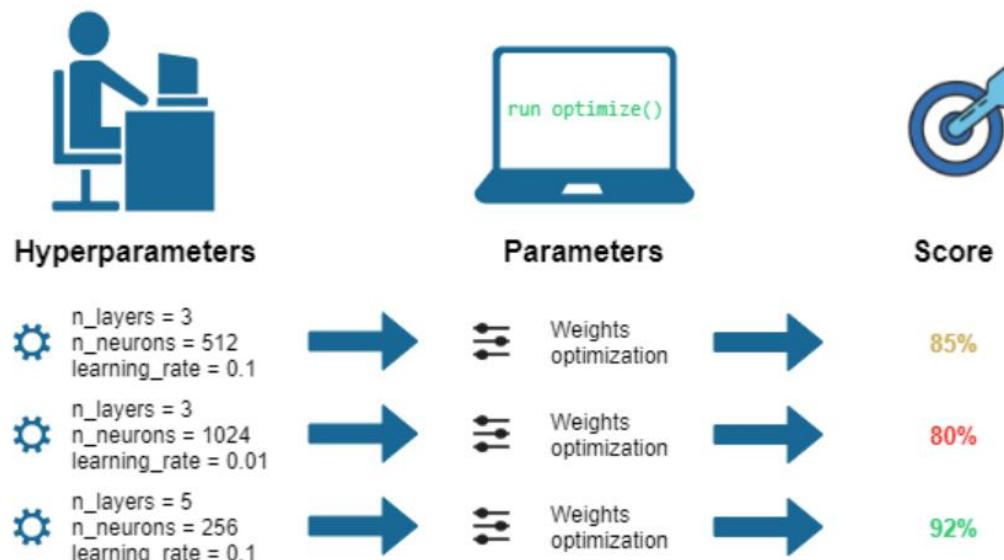
# Modellauswahl



- **Perspektive Daten:**
  - Target kategorial vs. numerisch
  - Lernstrategie: beaufsichtigt vs. unbeaufsichtigt
  - Umfang Datenbestand
  - Outlier, Missing Value, ...
- **Perspektive Use Case:**
  - Performance: Was ist akzeptabel?
  - Simplicity: Verwenden Sie kein komplexes Modell für ein einfaches Problem.
  - Interpretability: Wie gut muss ich den Entscheidungsprozess kennen?
  - Computation Costs: Es muss in angemessener Zeit trainierbar und mit angemessener Hardware anwendbar sei.
  - Competition: Vergleiche – wenn mehrere Modelle zu den o.a. Kriterien.

# Hyperparameter vs Modellparameter

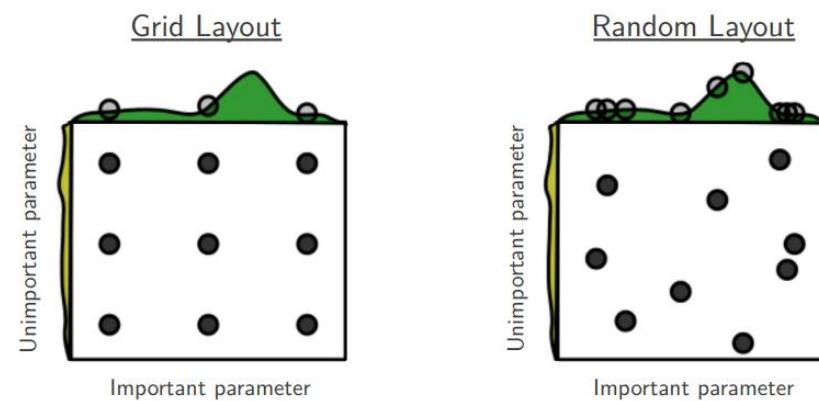
---



- **Hyperparameter**  
Alle Parameter, die vor Beginn des Trainings beliebig eingestellt werden können (z. B. Anzahl der Bäume im Random Forest).
- **Modellparameter**  
Parameter, die während des Modelltrainings gelernt werden (z. B. Weights (Gewichte) in neuronalen Netzen oder Intercept and Slope (Ursprung & Steigung) bei der linearen Regression).

Quelle: [Hyperparameters Optimization. An introduction on how to fine-tune... | by Pier Paolo Ippolito | Towards Data Science](#)

# Hyperparameter Tuning

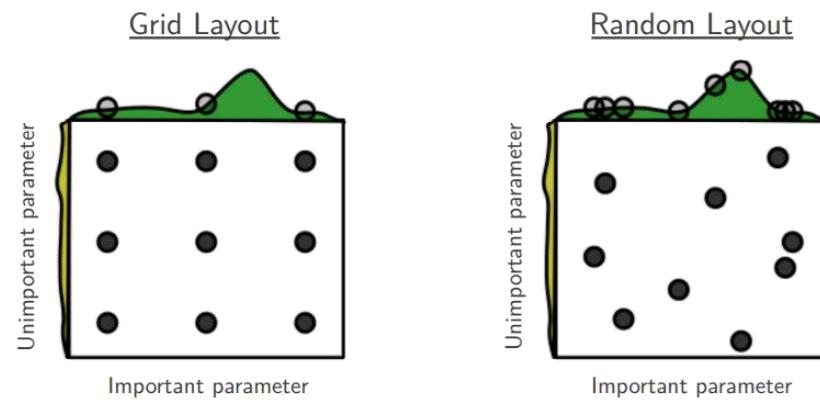


- Hyperparameter Tuning ist die Suche nach optimalen Hyperparametern.
- Ein Hyperparameter ist ein Parameter, der zur Steuerung eines ML-Algorithmus verwendet wird und dessen Wert vor dem eigentlichen Training des Modells festgelegt werden muss.
- Für das Hyperparameter Tuning gibt es verschiedene Ansätze.
- Näher betrachtet werden Raster- und Zufallssuche.
- Rastersuche
  - Es werden in einer erschöpfenden Suche die Hyperparameter in allen möglichen Kombinationen auf den ML-Algorithmus angewendet.
  - Die untersuchte Untermenge der Hyperparameter muss im Vorfeld festgelegt werden.

Quelle: [James Bergstra, Yoshua Bengio: Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research 13 \(2012\) 281-305](#)

# Hyperparameter Tuning (Forts.)

---



Quelle: [James Bergstra, Yoshua Bengio: Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research 13 \(2012\) 281-305](#)

- Zufallssuche (Random Grid)
  - Es wird mit einer zufälligen Auswahl von Werten der Hyperparameter der ML-Algorithmus angewendet.
  - Die Zufallsuche erfolgt in einem vorgegebenen Parameterraum.
  - der Hyperparameter muss im Vorfeld festgelegt werden.
- Die Zufallssuche kann die Rastersuche in Geschwindigkeit und Performance übertreffen.
  - Bei der Rastersuche werden für jeden Parameter nur wenige vorgegebene Werte ausprobiert.
  - Die zufällig ausgewählten Werte im Suchraum sind oft wesentlich besser verteilt.
- Guter Überblick siehe [Wikipedia](#)

# Modelle mit kleinen Datenmengen

---



Foto von Jenna Hamra: [Pexels](#)

- **Merkmalsauswahl:**  
Reduzierend der Dimensionalität der Daten, indem nur die relevantesten Merkmale für Ihre Vorhersagen verwenden. Methoden: Auswahl nach Wichtigkeit (feature importance), Korrelationsanalyse oder mit automatischen Feature-Auswahltechniken.
- **Merkmalskonstruktion:**  
Erstellen neuer Merkmale aus den vorhandenen Daten, die relevante Informationen zusammenfassen darstellen.
- **Modellwahl:**  
Einfache Modelle bevorzugen, komplexe Modelle neigen dazu, über kleine Datenmengen zu overfitten. Einfachere Modelle wie lineare Regression oder Entscheidungsbäume sind oft effektiver.
- **Regularisierungstechniken:** Verwenden von Modellen mit Regularisierung (z.B. Lasso, Ridge Regression), um Overfitting zu vermeiden.
- **Cross-Validation:** K-Fold Cross-Validation: Verwenden von Techniken wie K-Fold Cross-Validation, um sicherzustellen, dass das Modell nicht überoptimiert ist und gut generalisiert. Dies ist besonders wichtig bei kleinen Datenmengen.

# Modelle mit kleinen Datenmengen

---

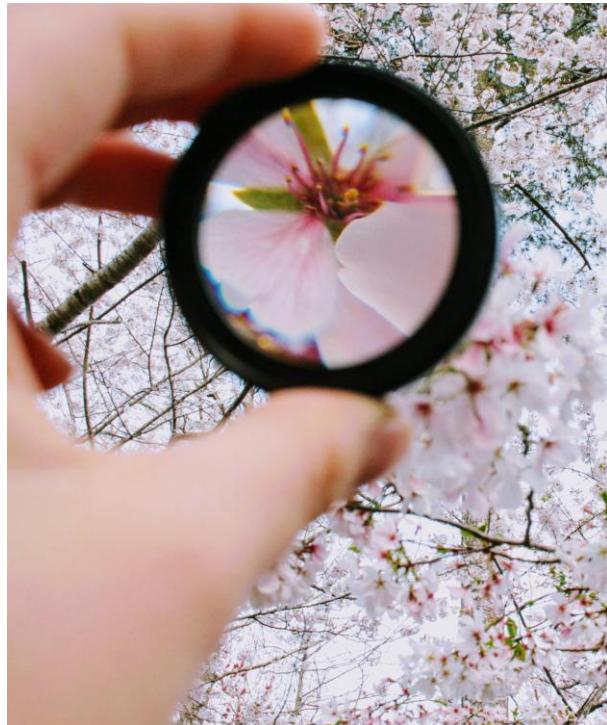


Foto von Jenna Hamra: [Pexels](#)

- **Bootstrapping und Resampling:**  
Erzeugen neuer Trainingssets durch Ziehen mit Zurücklegen aus Ihren vorhandenen Daten, um mehrere Modelle zu trainieren und ihre Vorhersagen zu mitteln.
- **Ensemble Methoden:**  
Verwenden Sie Ensemble-Methoden wie Random Forest oder XGBoost, die mehrere Lerner kombinieren, um die Vorhersageleistung zu verbessern.
- **Transfer Learning:**  
Hauptsächlich in Domänen wie Bild- und Sprachverarbeitung angewendet, wenn Modelle von ähnlichen Aufgaben übertragen und angepasst werden können.
- **Synthetische Datengenerierung:**  
Dataaugmentation, Techniken wie z.B. SMOTE (Synthetic Minority Over-sampling Technique) verwendet werden, um künstlich Beispiele für Minderheitenklassen zu generieren und so das Training zu verbessern.

# Evaluate

---



# Wie wenige Dinge beurteilen wir richtig!

---

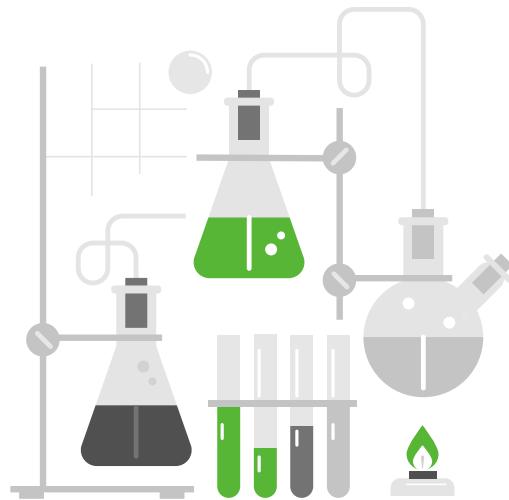
Luc de Clapiers  
Marquis de Vauvenargues



Bild von [Adrian](#) auf Pixabay

# Evaluate

---



- Modellbewertung ist der Teilprozess, der die Qualität der Vorhersagen eines Systems quantifizieren.
- Dazu wird die Leistung des trainierten Modells bzgl. der Qualität seiner Vorhersagen gemessen.
- Ziel der Bewertung ist es, die aktuelle Leistung zu quantifizieren und Möglichkeiten der Leistungsverbesserung aufzuzeigen.
- Fragen können sein:
  - Wie gut funktioniert das Modell?
  - Ist das Modell gut genug für den Produktivbetrieb?
  - Werden mehr Daten die Leistung des Modells verbessern?
  - ...

# Evaluation – good practice

---



Bild von [Konstantin Kolosov](#) auf [Pixabay](#)

- Bewertung der **Modellgüte**: Metriken wie den mittleren quadratischen Fehler (Mean Squared Error, MSE) oder den Bestimmtheitsmaß (Coefficient of Determination, R<sup>2</sup>), um die Genauigkeit des Modells (Training, Validierung, Test) zu bewerten.
- **Residuenanalyse**: Analyse der Residuen ( $\Delta$  real/predict) auf Muster oder systematische Fehler. Eine zufällige Verteilung der Residuen um Null deutet auf ein gutes Modell hin.
- **Feature Importance/Selection**: Überprüfung der Relevanz der verwendeten unabhängigen Feature im Modell, keine Verwendung potenziell irrelevante/redundante Feature. Interpretation der Koeffizienten der unabhängigen Feature, um Stärke/Richtung des Zusammenhangs zwischen Feature & Zielvariable zu gewinnen.
- **Robustheitstests**: Test auf Robustheit des Modells durch mit verschiedenen Teilmengen der Daten, um sicherzustellen, dass die Ergebnisse konsistent sind.
- **Sensitivitätsanalyse**: Analyse des Modells bei Veränderungen in den Eingabefeature / Vorhersagen des Modells, um die Robustheit und Stabilität der Ergebnisse zu bewerten.
- **Modellinterpretation**: Gesamtheitliche Analyse der Ergebnisse. Explorative Analyse der prognostizierten Zielvariablen.
- **Kommunikation der Ergebnisse**: Zusammenfassung der Ergebnisse in einer klaren und verständlichen Weise mit Schlussfolgerungen, Einschränkungen und möglichen Handlungsempfehlungen.

# Evaluation – good practice

Evaluationen	local	global
Modellgüte	Probability	Accuracy, F1-Score, Confusion Matrix, R <sup>2</sup> , MAE, Silhouette-Koeffizient, Hyperparameter-Tuning
Residuenanalyse	Δ real / predicted	Δ real / predicted, Residuals-Plots
Feature Importance / Selection	Break Down Analyse, Shapley Values	Feature Importance/Selection, Recursive Feature Elimination
Robustheitstest	Δ real / predicted, Ceteris Paribus Analyse	Δ real / predicted, Cross Validation, Bootstrapping, Learning Curve, Validation Curve, ROC, AUC
Modellinterpretation	Break Down Analyse, Shapley Values	Δ real / predicted, Histogramm, Box-Plot, Scattergramm, Trees, Feature Importance
Sensitivitätsanalyse	Ceteris Paribus Analyse	Ceteris Paribus Profile (CDP), Accumulated Local Dependence Profile (ALDP), Partial Dependence Plot
Kommunikation	best of above, keep it simple	best of above, keep it simple

# Classification

---



# Confusion Matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Die Konfusionsmatrix (Confusion Matrix) zeigt, ob die Prognose einer Klassifizierung richtiger- oder fälschlicherweise wahr oder falsch ist.

- True Positive (TP): Positives Ergebnis vorhergesagt; Positive Realität
- False Positive (FP): Positives Ergebnis vorhergesagt; Negative Realität
- True Negative (TN): Negativ Ergebnis vorhergesagt; Negative Realität
- False Negative (FN): Negatives Ergebnis vorhergesagt; Positive Realität

Bild: [Confusion Matrix for Your Multi-Class Machine Learning Model](#) | by Joydwip Mohajon | Towards Data Science

 [KNIME Scoring Metrics for Classification](#)  [KNIME Confusion Matrix](#)  
 [StatQuest Confusion Matrix](#)  [StatQuest Sensitivity and Specificity](#)

# Confusion Matrix - Metrics

		Metrik	Berechnung	Interpretation
Predicted Class	True Class	Accuracy (Genauigkeit)	$\frac{TP + TN}{\text{Summe aller Fälle}}$	Anteil der <b>richtig klassifizierten</b> Fälle Bewertung: 0 (schlecht) und 1 (sehr gut) Beispiel: $(20 + 15) / 43 = 0,81$
		Precision (Relevanz)	$\frac{TP}{TP + FP}$	%-Anteil der <b>richtig positiven</b> Fälle Bewertung: 0 (schlecht) und 1 (sehr gut) Beispiel: $20 / (20 + 5) = 0,80$
		Recall (Sensitivität)	$\frac{TP}{TP + FN}$	%-Anteil der richtig <b>erkannten</b> pos. Fälle Bewertung: 0 (schlecht) und 1 (sehr gut) Beispiel: $20 / (20 + 3) = 0.87$
		F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Maß für <b>Gesamtleistung</b> Bewertung: 0 (schlecht) und 1 (sehr gut) Beispiel: $2 \times (0.696 / 1,672) = 0,833$

KNIME Accuracy & Statistics

KNIME Cohen's Kappa

StatQuest Confusion Matrix

# Confusion Matrix – Multi Class

**Beispiel: Apple**

$$TP = 7$$

$$FN = (1 + 3) = 4$$

$$FP = (8 + 9) = 17$$

$$TN = (2+3+2+1) = 8$$

			True Class		
			Apple	Orange	Mango
Predicted Class	Apple	7	8	9	
	Orange	1	2	3	
	Mango	3	2	1	

- Anders als bei der binären Klassifizierung gibt es hier keine positiven oder negativen Klassen.
- Am Anfang mag es etwas schwierig sein, TP, TN, FP und FN zu finden, da es keine positiven oder negativen Klassen gibt, aber es ist eigentlich ziemlich einfach.
- Was wir hier tun müssen, ist TP, TN, FP und FN für jede einzelne Klasse zu finden. Wenn wir zum Beispiel die Klasse Apple nehmen, sehen wir uns an, was die Werte der Metriken aus der Konfusionsmatrix sind.

Bild-Quelle: [Confusion Matrix for Your Multi-Class Machine Learning Model | by Joydwip Mohajon | Towards Data Science](#)

 [KNIME Confusion Matrix](#)  [StatQuest Confusion Matrix](#)

 [StatQuest Sensitivity and Specificity](#)

# Cohen's Kappa

---



Bild von [Tom](#) auf [Pixabay](#)

- Cohen's Kappa ist eine statistische Metrik, die zur Bewertung der **Übereinstimmung zwischen zwei** Beobachtern oder **Klassifikatoren** verwendet wird.
- Cohen's Kappa liegt zwischen -1 und 1, wobei
  - negative Werte auf eine schlechte Übereinstimmung,
  - Werte nahe 0 auf eine zufällige Übereinstimmung und
  - Werte nahe 1 auf eine starke Übereinstimmung hinweisen.
- Es ist ein Maß, wie konsistent die Vorhersagen eines Modells im Vergleich zu den tatsächlichen Labels sind.
- Es ist ein robusteres Maß als die einfache Accuracy, besonders in Fällen, wo die Klassendistribution unausgewogen ist.

🎥 [Cohen's Kappa](#)

# Receiver Operating Characteristic (ROC)

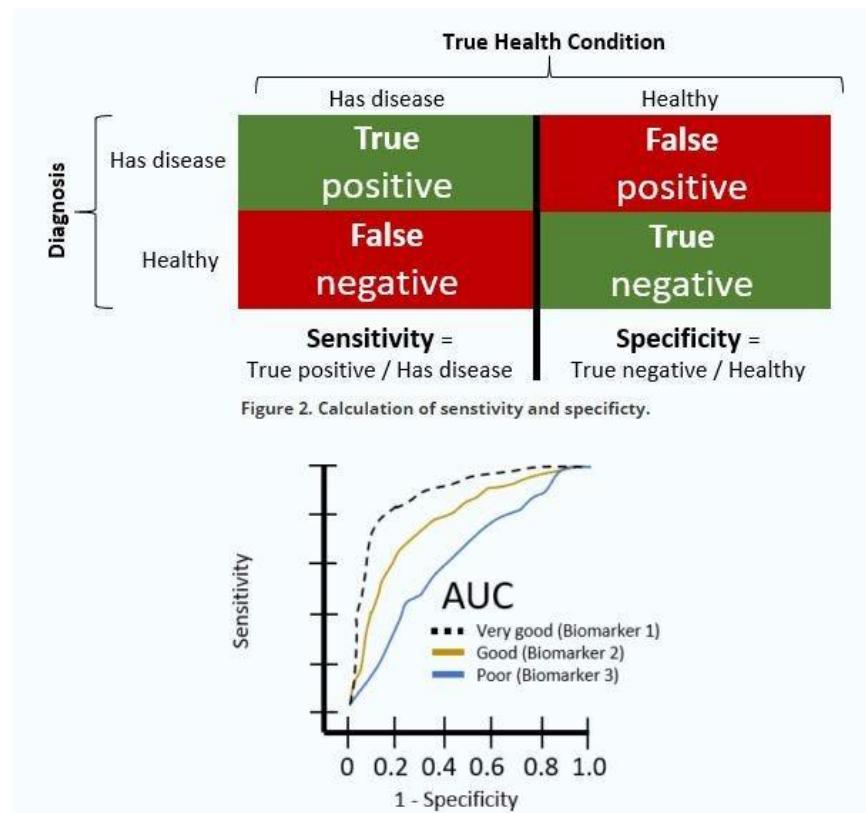


Bild: [What is the AUC — ROC Curve?. AUC-ROC CURVE | CONFUSION MATRIX \[...\] by Anuganti Suresh | Computer Architecture Club | Medium](#)

- ROC-Kurve: Ist ein Diagramm, das zeigt, wie gut ein Klassifizierungsmodell alle Klassen unterscheidet.
- Es wird die **True Positive Rate** (TPR, **Sensitivity**) und die **False Positive Rate**, (FPR, **1-Specificity**) bei verschiedenen Schwellenwerten (default: 0.5) ermittelt.
- Die Sensitivität gibt an, welcher Anteil der positiven Klasse korrekt klassifiziert wurde.
- Die Spezifität gibt an, welcher Anteil der negativen Klasse richtig klassifiziert wurde.
- Durch eine  $45^\circ$  Diagonale erfolgt ein Vergleich mit einer zufälligen Ausprägung.
- Eine Kurve, die sich nah an der linken oberen Ecke des Plots befindet, zeigt eine hohe Trennschärfe des Modells an.

- [KNIME ROC Curve](#)
- [KNIME ROC Curve Classification Problem](#)
- [StatQuest ROC and AUC](#)

# ROC - Optimaler Schwellenwert

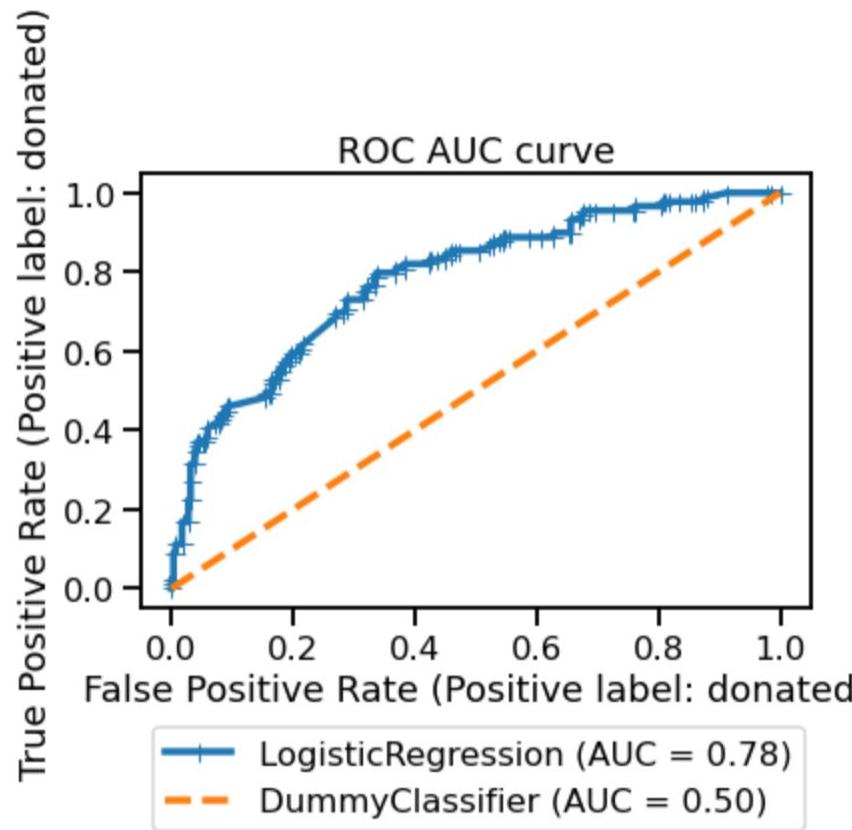
Beispiel: Test auf eine ernsthafte, aber behandelbare Krankheit wie z.B. Brust- oder Prostatakrebs:

- Hohe TPR (Sensitivität): Eine **hohe Sensitivität** bedeutet, dass fast alle tatsächlichen Fälle von Krebs erkannt werden. Bei einer ernsthaften Krankheit wie Krebs ist es entscheidend, dass der Test so viele tatsächliche Fälle wie möglich erkennt (True Positives), um die Behandlung frühzeitig beginnen zu können.
- Höhere FPR: Eine **höhere FPR** bedeutet, dass einige gesunde Personen fälschlicherweise als positiv diagnostiziert werden könnten. Während dies sicherlich Stress verursacht und zusätzliche Tests nach sich zieht, wird es als akzeptables Risiko angesehen, wenn es bedeutet, dass tatsächliche Fälle nicht übersehen werden.
- Es ist ersichtlich, dass eine ROC-Kurve verwendet werden kann, um einen **Schwellenwert** für einen Klassifikator auszuwählen, der die wahren Positivwerte maximiert und die falschen Positivwerte minimiert.



Bild von [PDPics](#) auf Pixabay

# Area Under the Curve (AUC)



- Area Under the Curve (AUC) ist ein Maß, das sich direkt auf die ROC-Kurve bezieht.
- Es ist die Fläche unter der ROC-Kurve.
- AUC ist ein numerischer Wert, der zwischen 0 und 1 liegt.
- Ein Wert von 1 zeigt eine perfekte Klassifizierung und ein Wert von 0,5 eine zufällige Klassifizierung (keine Unterscheidungsfähigkeit besser als Zufall).
- AUC eignet sich gut zum Methodenvergleich.
- Das Modell mit dem höchsten AUC-Wert ist im Allgemeinen das leistungsfähigste Modell.



[KNIME ROC Curve](#)



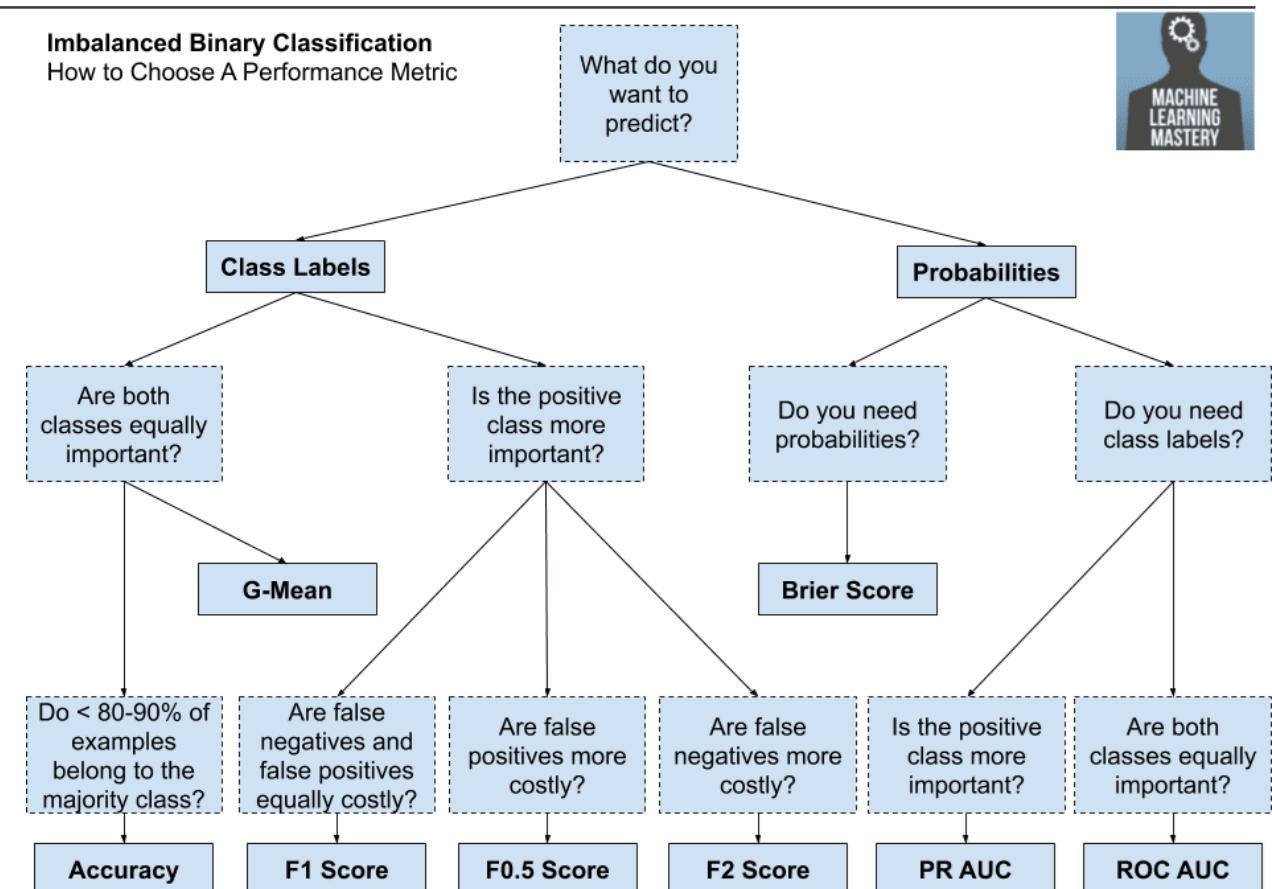
[KNIME ROC Curve Classification Problem](#)



[StatQuest ROC and AUC](#)

# Imbalanced Classification

- Wie wählt man eine Leistungskennzahl für eine binäre Klassifikation bei unausgewogenen Daten aus?
- Basierend auf dieser Entscheidung „Was möchte man vorhersagen“ und weiteren Kriterien empfiehlt das Diagramm spezifische Leistungskennzahlen, die am besten zu den spezifischen Anforderungen passen.



Quelle: Step-By-Step Framework for Imbalanced Classification Projects - MachineLearningMastery.com

© 2019 MachineLearningMastery.com All Rights Reserved.

# Regression

---



# Bestimmtheitsmaß ( $R^2$ )

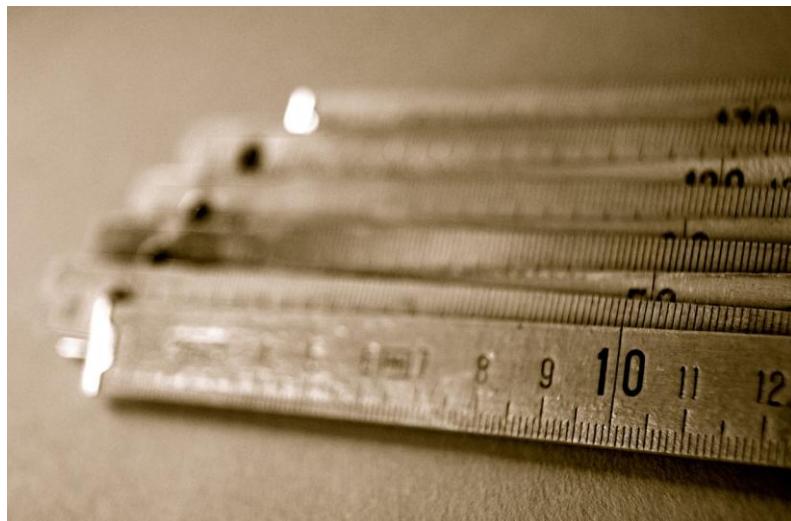


Bild von [Uwe Baumann](#) auf [Pixabay](#)

- Das Bestimmtheitsmaß ist ein statistisches Maß, das verwendet wird, um den Grad der Erklärungskraft eines Modells oder einer Regressionsanalyse zu quantifizieren.
- Es misst, wie gut die abhängige Variable durch die unabhängigen Variablen erklärt werden kann. In der Regel wird das Bestimmtheitsmaß als  $R^2$  bezeichnet und liegt zwischen 0 und 1.
- Ein Wert von 1 bedeutet, dass das Modell die abhängige Variable vollständig erklärt, während ein Wert von 0 darauf hindeutet, dass die unabhängigen Variablen keinerlei Vorhersagekraft haben.
- Es sollte immer in Verbindung mit anderen statistischen Maßen und einer sorgfältigen Interpretation der Ergebnisse verwendet werden.

 [StatQuest R-squared](#)

# Mean Absolut Error (MAE)

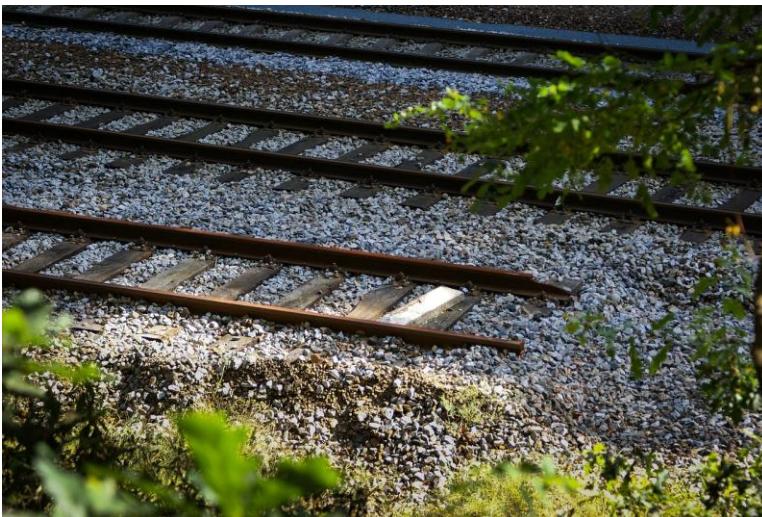
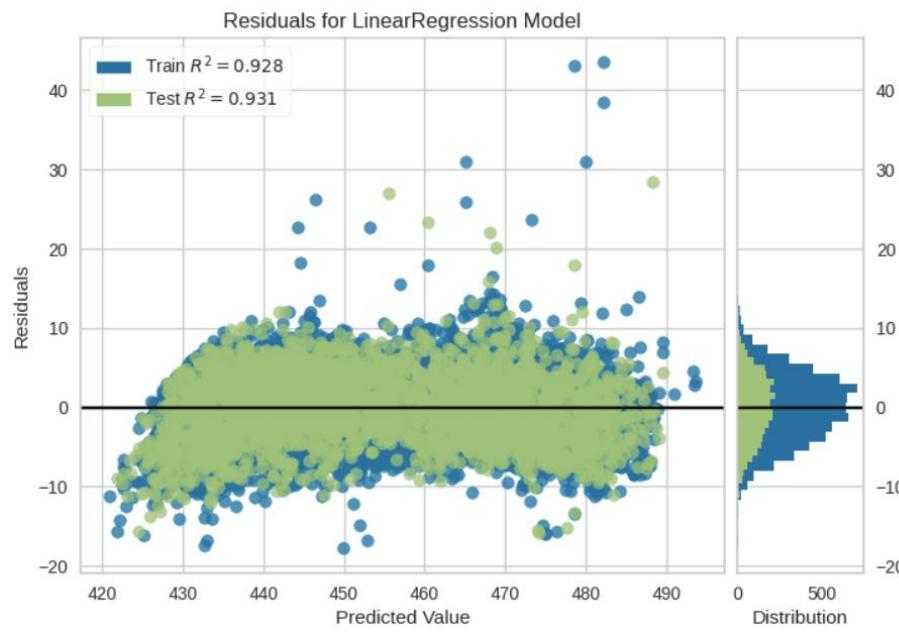


Bild von [Niek Verlaan](#) auf [Pixabay](#)

- Der Mean Absolute Error (MAE) ist eine häufig verwendete Metrik zur Bewertung der Genauigkeit von Vorhersagemodellen.
- Er misst den durchschnittlichen absoluten Unterschied zwischen den tatsächlichen Werten und den vorhergesagten Werten.
- Der MAE ist besonders nützlich, wenn Ausreißer in den Daten vorhanden sind, da er weniger empfindlich auf extreme Werte reagiert als andere Metriken.
- Es ist ratsam, den MAE mit anderen Metriken zu vergleichen, um ein umfassenderes Bild der Modellgenauigkeit zu erhalten.

# Residual Plot



- Ein Residualplot ist eine statistische Analyse, die Unterschiede zwischen beobachteten und modellvorhergesagten Werten visualisiert.
- Diese Unterschiede, oder Residuen, zeigen Modellfehler an.
- Residuen sollten zufällig verteilt sein, was auf eine gute Modellanpassung hindeutet.
- Systematische Muster in Residuen können auf nicht erfasste nichtlineare Beziehungen hinweisen.
- Große Abweichungen im Residualplot deuten auf Ausreißer hin, die das Modell stark beeinflussen können.
- Konstante Streuung der Residuen (Homoskedastizität) sind ideal; variierende Varianz kann Modellschätzprobleme anzeigen.

# Clustering

---



# Silhouette-Koeffizient



Bild von [Pfüderi](#) auf [Pixabay](#)

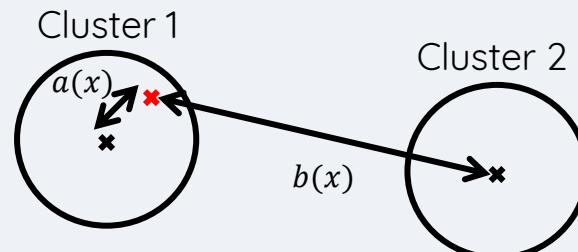
- Der Silhouette-Koeffizient ist ein Maß für die Qualität einer Clusterbildung in einer Clusteranalyse.
- Er bewertet, wie gut eine Beobachtung zu ihrem eigenen Cluster im Vergleich zu anderen Clustern passt.
- Der Silhouette-Koeffizient liegt zwischen -1 und 1.
- Ein hoher Silhouette-Koeffizient nahe 1 deutet darauf hin, dass die Beobachtungen gut in ihre eigenen Cluster passen und weit von anderen Clustern entfernt sind.
- Ein Wert nahe 0 deutet darauf hin, dass die Beobachtungen in der Nähe der Trennung zwischen Clustern liegen.
- Ein negativer Wert nahe -1 zeigt an, dass die Beobachtungen wahrscheinlich zu einem falschen Cluster zugeordnet wurden.

# Silhouette-Koeffizient

Berechnung:

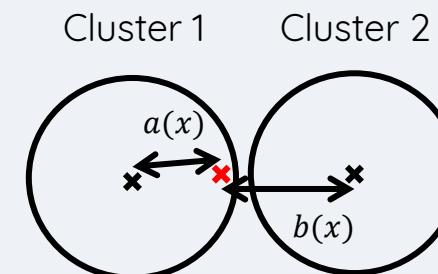
- $a(x)$ : Abstand des Objekts X zu seinem Cluster-Repräsentanten
- $b(x)$ : Abstand des Objekts x zum Repräsentanten des "zweitbesten" Clusters
- Silhouette  $s(x) = \frac{b(x)-a(x)}{\max\{a(x),b(x)\}}$

Good clustering...



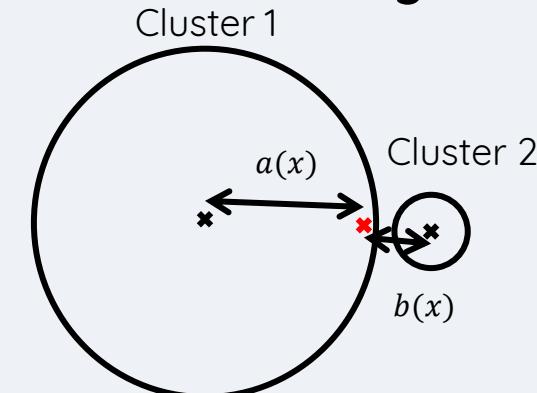
$\sim 1$

...not so good...



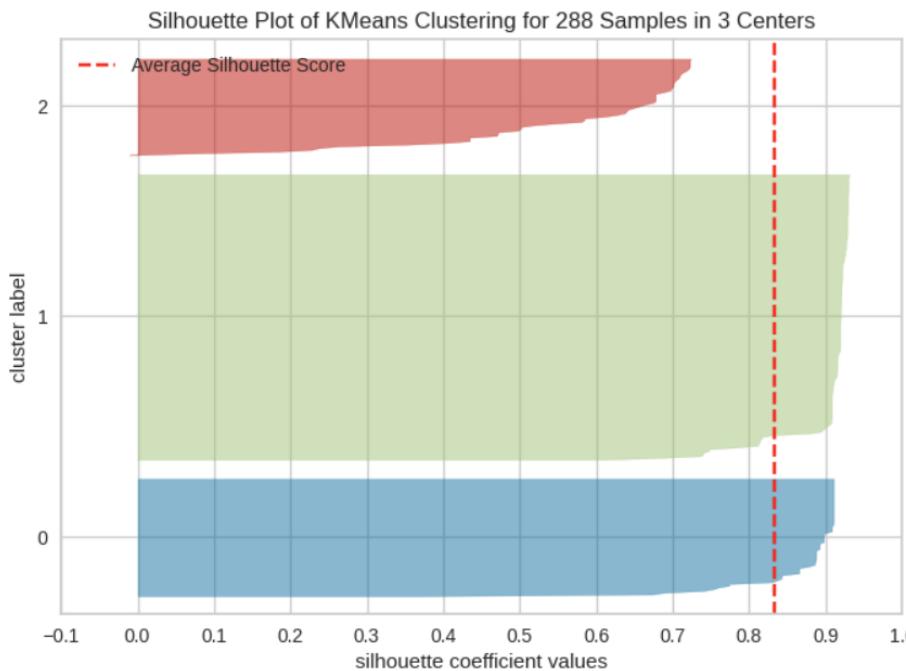
$\sim 0$

...bad clustering.



$\sim -1$

# Silhouette-Plot



- Ein Silhouettenplot ist eine Grafik, die die Qualität von Clustern in einem Datensatz zu bewerten.
- Im Silhouettenplot werden die Silhouettenwerte aller Datenpunkte als horizontale Balken dargestellt, die nach der Größe der Silhouettenwerte geordnet sind.
- Cluster mit vielen positiven Silhouettenwerten sind gut definiert.
- Cluster mit vielen negativen oder nahe null liegenden Werten sind schlecht definiert.
- Der Silhouettenplot kann verwendet werden, um die optimale Anzahl von Clustern in einem Datensatz zu bestimmen.
- Durch das Plotten der durchschnittlichen Silhouettenwerte für verschiedene Clusteranzahlen kann man die Anzahl der Cluster identifizieren, die die höchste durchschnittliche Silhouettenbreite aufweist, was auf eine gute Clusterstruktur hinweist.

# Anomalie

---



# Anomalie Score

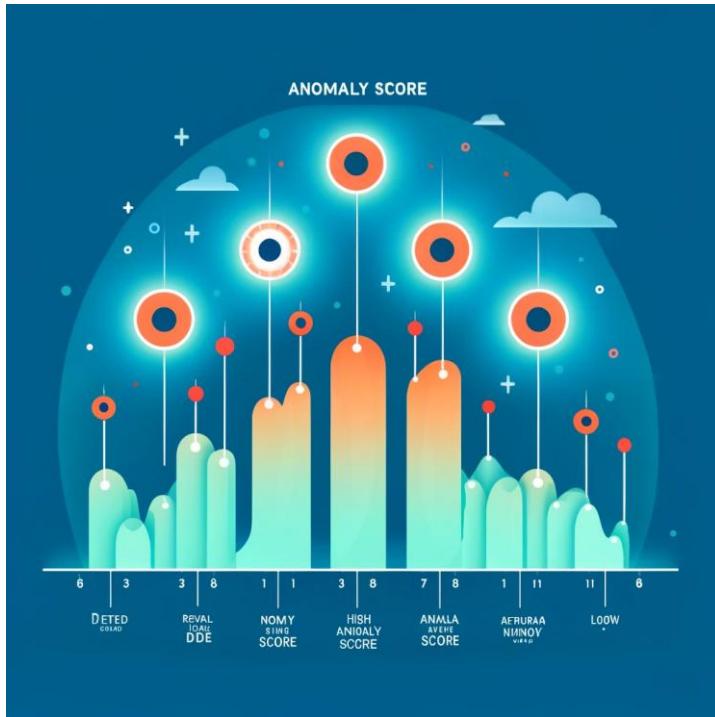


Bild wurde mit DALL-E erstellt

- Der Anomalie-Score bewertet, wie anomal oder atypisch ein bestimmter Datenpunkt im Vergleich zum Rest des Datensatzes ist.
- Werte nahe -1:  
Diese Werte deuten darauf hin, dass der Datenpunkt sehr wahrscheinlich anomali ist. Je näher der Wert an -1 liegt, desto anomaler ist der Datenpunkt im Vergleich zum Rest der Daten.
- Werte nahe 1:  
Diese Werte deuten darauf hin, dass der Datenpunkt sehr wahrscheinlich normal ist. Je näher der Wert an 1 liegt, desto typischer ist der Datenpunkt im Vergleich zum Rest der Daten.
- Anomalie-Scores werden von Algorithmen wie Isolation Forest oder Autoencodern berechnet, die darauf trainiert sind, das „Normalverhalten“ innerhalb eines Datensatzes zu erkennen.

# Cross Validation

---



# Cross Validation

Beispiel:

Kreuzvalidierung mit k=5 Stichproben (K-Fold)

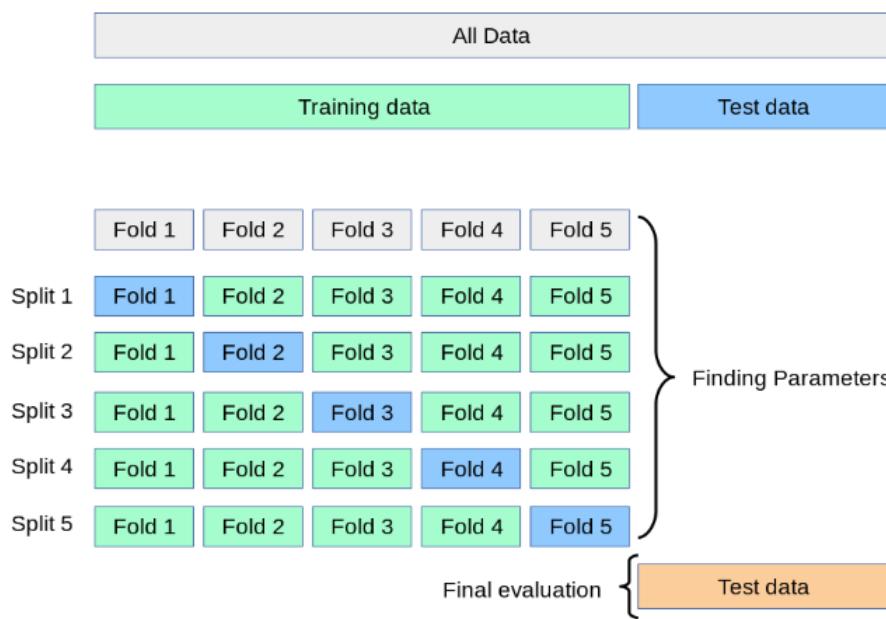


Bild: [3.1. Cross-validation: evaluating estimator performance](#)  
– scikit-learn 1.4.0 documentation

- Das Kreuzvalidierungverfahren (cross-validation) ist eine Methode zur Prüfung der Generalisierungsfähigkeit eines Vorhersagemodells.
- **Generalisierungsfähigkeit** = wie gut kann ein trainiertes Lernmodell neue, bisher ungesehene Daten vorhersagen.
- Zur Bewertung der Leistung eines ML-Modells, wird das Modell mehrfach mit einem veränderten Datenset trainiert & getestet (validiert).
- Hierbei wird der Datenbestand in k unabhängige Stichproben aufgeteilt.
- Bei jedem Lauf trainiert das Modell auf einer anderen Stichprobe der Trainings- und Testdaten.
- Die Ergebnisse der einzelnen Modell-Läufe geben Aufschluss, wie groß die Streuung der Ergebnisse des Modells sind.

[StatQuest Cross Validation](#)

# Strategien Cross Validation

---

Merkmal	K-Fold	Leave-One-Out (LOO)	Hold-Out <input checked="" type="checkbox"/>
Beschreibung	Teilt den Datensatz in K gleich große Teile (Folds). Bei jedem Durchlauf wird ein Fold als Testdatensatz und die restlichen als Trainingsdatensatz verwendet.	Jedes Datenbeispiel wird genau einmal als Testdatensatz verwendet, während die restlichen als Trainingsdatensatz dienen. Dies ist ein spezieller Fall von K-Fold, bei dem K gleich der Anzahl der Datenbeispiele ist.	Teilt den Datensatz in zwei Teile: einen für das Training und einen für das Testen, typischerweise im Verhältnis 70:30 oder 80:20.
Rechenaufwand	Moderat, abhängig von K.	Hoch, da für jedes Datenbeispiel ein Modell trainiert wird.	Niedrig, da das Modell nur einmal trainiert wird.
Varianz der Schätzungen	Kann durch die Wahl von K beeinflusst werden. Größeres K führt zu niedrigerem Bias, aber möglicherweise höherer Varianz.	Niedrige Bias, aber hohe Varianz, da die Trainingssätze sehr ähnlich sind.	Kann zu höherem Bias führen, besonders wenn der Datensatz nicht gut gemischt ist.
Anwendungsbereiche	Gut für mittelgroße bis große Datensätze. Ermöglicht eine gute Balance zwischen Rechenaufwand und Schätzgenauigkeit.	Empfehlenswert für sehr kleine Datensätze, bei denen jedes Datenbeispiel wertvoll ist.	Geeignet für große Datensätze oder als schneller, erster Ansatz zur Modellbewertung.
Überanpassung	Risiko kann durch Auswahl eines angemessenen K gemindert werden.	Höchstes Risiko der Überanpassung, da fast alle Daten zum Training verwendet werden.	Risiko hängt von der Größe des Testdatensatzes ab; kann durch Zufallsauswahl gemindert werden.
Effizienz	Effizient, wenn K sorgfältig gewählt wird.	Wenig effizient bei großen Datensätzen.	Sehr effizient, erfordert jedoch eine sorgfältige Aufteilung der Daten.

# Nested Cross Validation

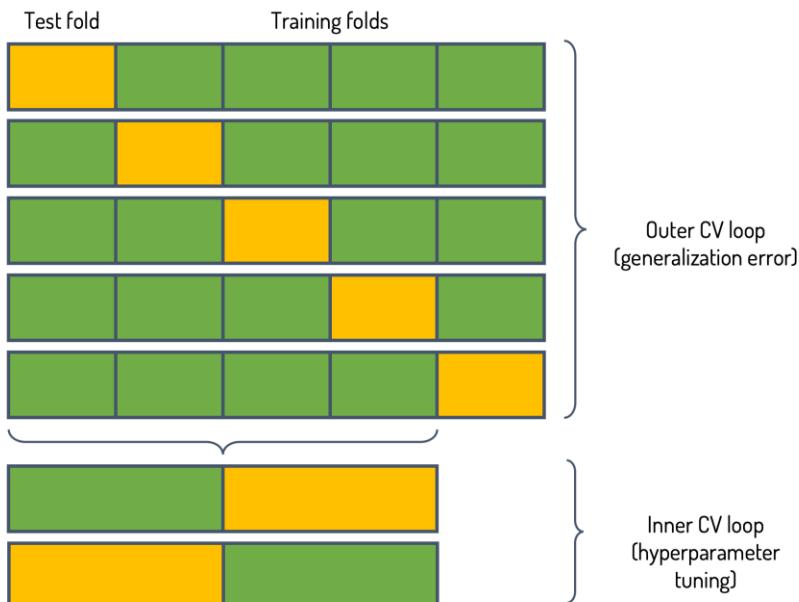


Bild: A “short” introduction to model selection |  
Moving Upwards ([dschoenleber.github.io](https://dschoenleber.github.io))

- **Nested Cross-Validation** ist eine erweiterte Form der Cross-Validation.
- Sie besteht aus zwei Ebenen der Cross-Validation:
- Äußere Schleife: Teilt den Datensatz in  $k$  Folds auf. Jeder Fold dient einmal als Testset (Validation), während die restlichen Folds als Trainingsset verwendet werden.
- Innere Schleife: Innerhalb jedes Trainingssets der äußeren Schleife wird eine weitere **Cross-Validation** durchgeführt, um die **Hyperparameter** zu optimieren. Dies bedeutet, dass das Training und die Validierung mehrmals innerhalb jedes Trainingssets der äußeren Schleife stattfinden.
- Dies verhindert Informationslecks und Überanpassung, die auftreten können, wenn dieselben Daten sowohl für die Hyperparameter-Abstimmung als auch für die Modellbewertung verwendet werden

# Bootstrapping

---



# Bootstrapping

---



Bild von [Лариса Мозговая](#) auf Pixabay

- Bootstrapping ist eine statistische Methode für die **Unsicherheitsquantifizierung** von Schätzern.
- Aus dem DataSet werden wiederholt Stichproben gezogen.
- Jede dieser neuen Stichproben wird durch Ziehen eines Datensatzes mit Zurücklegen erstellt.
- Das bedeutet, dass jeder Datensatz in der neuen Stichprobe mehrmals oder gar nicht vorkommen kann.
- Für jede dieser neuen Stichproben berechnet man die gewünschte Maßzahl (z.B. Accuracy oder  $R^2$ ).
- Nach einer ausreichenden Anzahl von Stichproben, verwendet man die Verteilung der Maßzahl, um die Unsicherheit oder das Konfidenzintervall der ursprünglichen Schätzung zu erfahren.
- Bootstrapping ist bei einigen Modellen bereits integriert, z.B. in den Bagging-/Boosting-Algorithmen.

# Cross Validation & Bootstrapping

---



Bild von [Paolo Trabattoni](#) auf [Pixabay](#)

Ansatz	Einsatz
Cross Validation	<b>Robustheit</b> eines maschinellen Lernmodells in Bezug auf seine Fähigkeit zur <b>Generalisierung</b> über verschiedene Datenstichproben hinweg.
Bootstrapping	<b>Robustheit</b> im bezogen auf die <b>Zuverlässigkeit</b> von <b>Schätzungen</b> sowie zur Berechnung von Standardfehlern und Konfidenzintervallen der Schätzer.

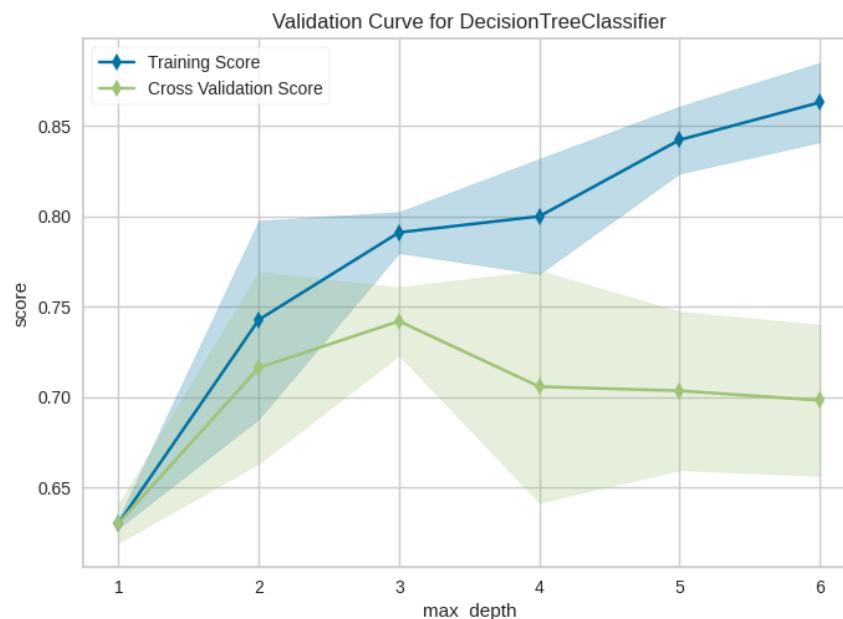
# Validation & Learning Curve

---



# Validation Curve

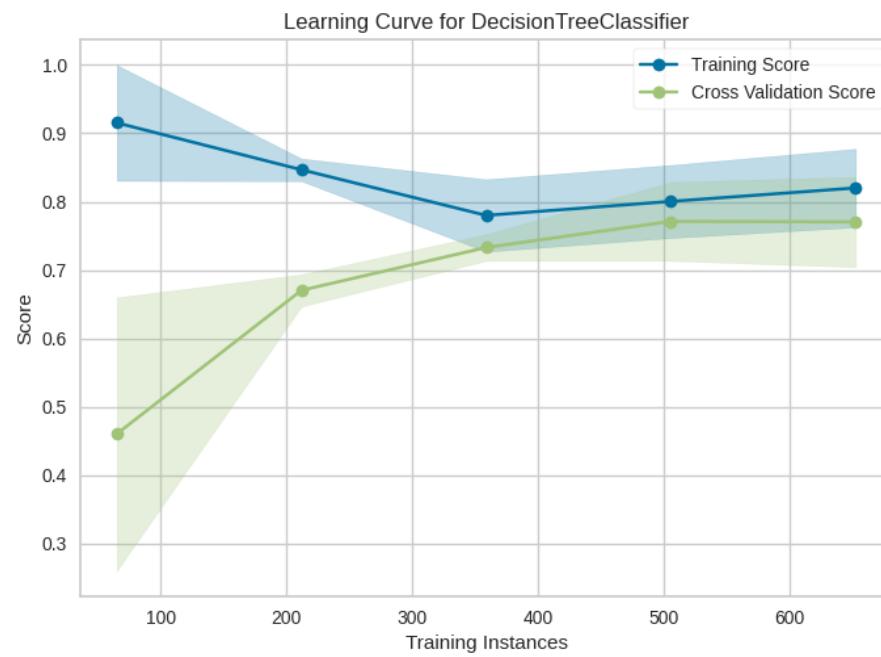
---



- Die Validierungskurve zeigt die Empfindlichkeit des ML-Modells bei Änderung von Parametern.
- Eine Validierungskurve wird zwischen einem Parameter des Modells und der Bewertung des Modells gezeichnet.
- In einer Validierungskurve werden die Bewertung für die Trainingsdaten und eine für die Kreuzvalidierung gezeigt.
- Beispiel: Wenn die Tiefe des Entscheidungsbaums zunimmt, nimmt die Genauigkeit des Trainingsergebnisses zu, die Kreuzvalidierungsergebnis ab.

# Learning Curve

---



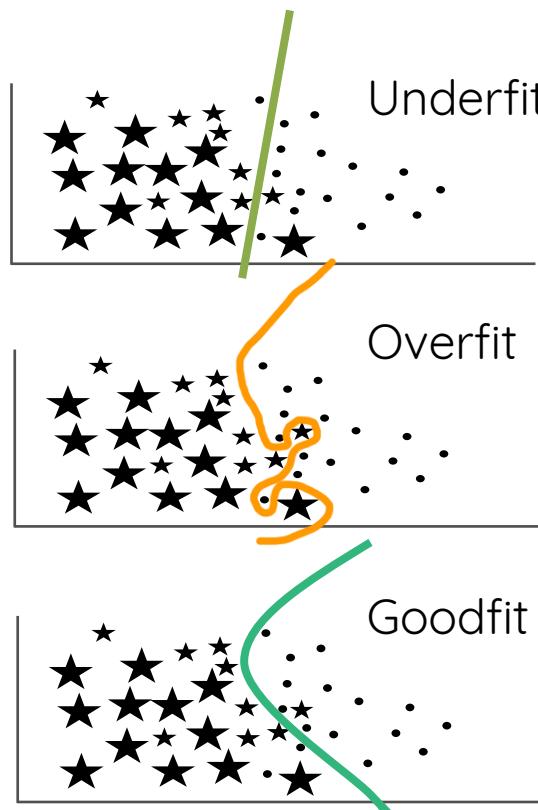
- Die Lernkurve (oder Trainingskurve) eines ML-Modell zeigt, wie sich die Qualität der Vorhersage ändert, wenn die Größe des Trainings-/Testdatensatzes zu- oder abnimmt.
- Eine Lernkurve kann helfen, die richtige Menge an Trainingsdaten zu finden.
- Im Idealfall sollte ein ML-Modell bei Änderung der Trainingssätze nicht zu stark variieren, d. h. das Modell sollte gut darin sein, wichtige Details über die Daten zu erfassen, unabhängig von den Daten selbst.

# Overfit

---



# Underfit/Overfit/Goodfit



- **Underfit:** Das Modell ist nicht gut genug, um die Beziehung zwischen den Eingabedaten  $x$  und der Ausgabe  $y$  zu beschreiben:
  - Das Modell ist zu einfach, um wichtige Muster in den Trainingsdaten zu erfassen.
  - Das Modell scheidet bei Training und Test schlecht ab.
- **Overfit:** Das Modell funktioniert bei den Trainingsdaten, es „versagt“ jedoch bei den „unbekannten“ Testdaten:
  - Das Modell ist zu komplex.
  - Das Modell ist zu stark auf die Trainingsdaten ausgerichtet.
- **Goodfit:** Das Modell erfasst die allgemeine Beziehung zwischen den Eingabedaten  $x$  und der Ausgabe  $y$ . Das Modell hat eine gute Performance bei Trainings- & Testdaten.

# Wie lässt sich Overfit vermeiden?

---

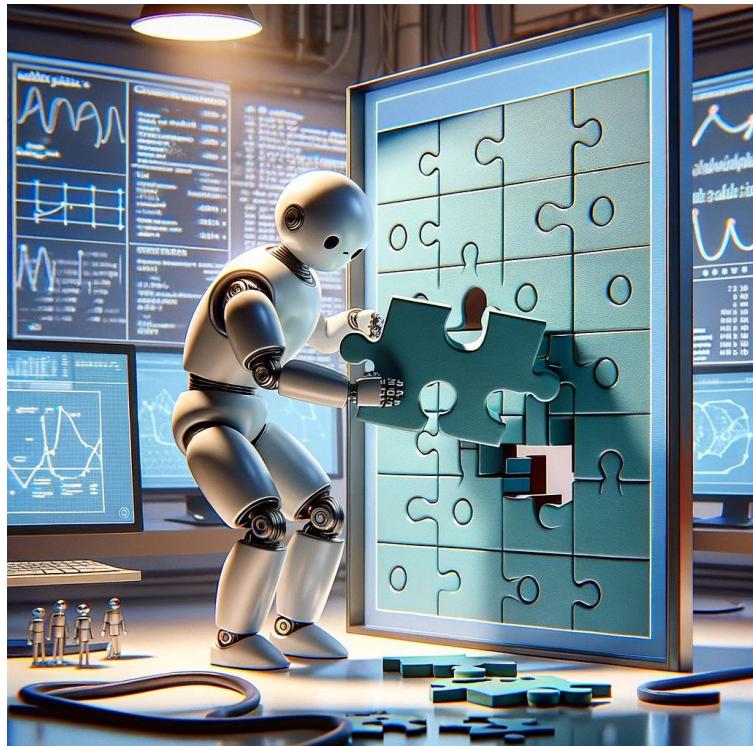


Bild mit DALL-E erstellt

- Train with more data
- Data augmentation
- Select/Remove features
- Early Stopping
- Regularization
- Ensembling
- Hyperparameter Tuning
- Cross-Validation
- Validation Curve
- Learning Curve
- Remove Layers (NN)
- Dropout (NN)

# Regularization

---



# Regularization



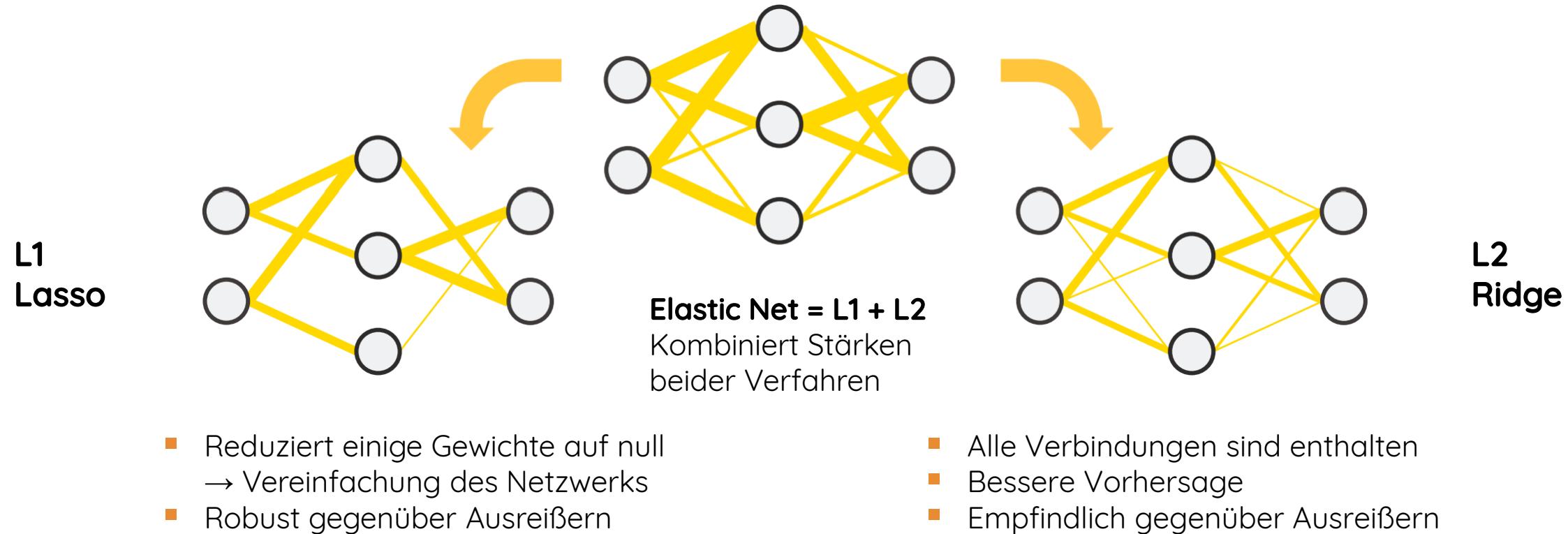
- Regularisierung ist eine Technik, die dazu dient, Überanpassung (Overfitting) zu verhindern, indem sie zusätzliche Informationen oder Einschränkungen in das Lernmodell einführt.
- Die Regularisierung erfolgt durch Anpassung der Gewichte.
- Ridge-Regularisierung (L2): **Reduzierung die Gewichte aller Merkmale**, um das Modell einfacher zu machen. Die Gewichte können klein werden, aber sie werden nie komplett auf null gesetzt. Anwendung bei kleinen Datenmengen.
- Lasso-Regularisierung (L1): Lasso kann **einige Merkmale ganz ausschließen**, indem die Gewichte auf null gesetzt werden. Anwendung bei großen Datenmengen.
- Die Intensität der Anpassung wird bei beiden Verfahren über einen Koeffizienten  $\alpha$  gesteuert.  $\alpha = 0 \rightarrow$  keine Anpassung,  $\alpha = \text{sehr groß} \rightarrow$  große Anpassung.

[KNIME Regularization](#)

[StatQuest Regularization #1 #2 #3 Ridge vs Lasso](#)

Bild mit DALL·E erstellt

# Regularization - Neural Network



# Regularisierung - Faustregeln

---



Foto von [Timothy Eberly](#) auf [Unsplash](#)

Situation	Regularisierung	Begründung
Viele Merkmale, einige vermutlich unwichtig	Lasso	Eliminiert unwichtige Merkmale
Multikollinearität	Ridge	Behält alle Merkmale bei, reduziert Gewichte
Sparsames Modell gewünscht	Lasso	Führt zu weniger Merkmalen im Modell
Mehr Merkmale als Beobachtungen	Lasso	Reduziert die Merkmalsanzahl
Robustheit gegenüber Ausreißern	Ridge	Weniger empfindlich als Lasso
Viele korrelierte Merkmale	Elastic Net	Kombiniert Lasso und Ridge Vorteile

# Deploy

---



# Geliefert in 10 Minuten!



Bild von [Kai Pilger](#) auf [Pixabay](#)

# Deployment

---

- Es bestehen große Unterschiede zwischen der Entwicklung eines ML-Modells und dessen Bereitstellung.
- Diese Unterschiede können zu enormen Diskrepanzen in Bezug auf Leistung, Geschwindigkeit und Ressourcenverbrauch führen.
- Einige der Herausforderungen sind:
  - Bereitstellung des ML-Modells an Kunden, deren Kenntnisse oder Verständnis für maschinelles Lernen begrenzt sind.
  - Transparenz & Nachvollziehbarkeit der ML-Modellergebnisse.
  - Veränderung Performance des ML-Modells ggf. unter den angegebenen Benchmark/Schwellenwert (z.B. durch Änderung der Daten-/Konzeptbasis).
  - Fortlaufende Überwachung der Modell-Performance.
  - Klare Verantwortlichkeit für den Produktionsbetrieb des Modells.
  - Versionsverwaltung.



[Deployment Options within KNIME](#)



Frameworks zur schnellen und einfach interaktiven Entwicklung von Apps, die Datenvisualisierungen, Dashboards und Machine-Learning-Modelle enthalten.

- Gradio ist ein weiteres Open-Source-Framework, das es Entwicklern ermöglicht, schnelle und einfache interaktive Anwendungen in Python und anderen Sprachen zu erstellen.
- Gradio verfügt über einen Drag-and-Drop-UI-Designer, der es Entwicklern erleichtert, interaktive Elemente wie Schieberegler, Textfelder, Dropdown-Menüs und Checkboxen hinzuzufügen.
- Es unterstützt auch mehrere Sprachen wie Python, Java, JavaScript, Ruby, R, Julia und C#.
- Gradio ermöglicht Entwicklern auch die Integration von Drittanbieter-Tools wie TensorFlow und PyTorch.
- Es kann einfach auf einem eigenen Server oder in der Cloud bereitgestellt werden.



# Model-Toolbox

---



# Intro Algorithmen & Modelle

# Überblick

Lernstrategien

Aufgabengruppen

Lernalgorithmen

Ergänzung:

- XGBoost Regression
- Stacking
- Isolation Forest



Quelle:  
[Online Graph Maker · Plotly Chart Studio](#)

# ShortCut Tool-Box (1/3)

Auswahl

Lernstrategie	Einsatzbereich	Algorithmus	Beschreibung	Bewertungsmetrik
Überwachtes Lernen	Klassifizierung Regression	Decision Tree	Ein Entscheidungsbaum teilt die Daten auf der Grundlage von Entscheidungsregeln auf. Er ist einfach zu verstehen und zu interpretieren.	Accuracy, F1-Score, AUC, Confusion-Matrix Cohens-Kappa $R^2$ , MAE, Residual Plot
	Klassifizierung Regression	Random Forest	Eine Ensemble-Methode, die mehrere Entscheidungsbäume kombiniert, um die Vorhersagegenauigkeit zu verbessern.	Accuracy, F1-Score, AUC, Confusion-Matrix Cohens-Kappa $R^2$ , MAE, Residual Plot
	Regression	Linear Regression	Modelliert die Beziehung zwischen einer abhängigen Variablen und einer oder mehreren unabhängigen Variablen durch lineare Ansätze.	$R^2$ , MAE, Residual Plot

# ShortCut Tool-Box (2/3)

Auswahl

Lernstrategie	Einsatzbereich	Algorithmus	Beschreibung	Bewertungsmetrik
Überwachtes Lernen	Klassifizierung	Logistic Regression	Modelliert die Wahrscheinlichkeit, dass eine Variable zu einer bestimmten Kategorie gehört, mit Ergebnissen zwischen 0 und 1.	Accuracy, F1-Score, AUC, Confusion-Matrix Cohens-Kappa
	Klassifizierung Regression	Neural Network	Bestehen aus Schichten von Knoten und können komplexe Beziehungen zwischen Eingaben und Ausgaben modellieren.	Accuracy, F1-Score, AUC, Confusion-Matrix Cohens-Kappa $R^2$ , MAE, Residual Plot
	Klassifizierung Regression	XGBoost	Eine optimierte Implementierung von Gradient Boosting, die für Geschwindigkeit und Leistung entwickelt wurde.	Accuracy, F1-Score, AUC, Confusion-Matrix Cohens-Kappa $R^2$ , MAE, Residual Plot
	Dimensionsreduktion	Linear Discriminant Analysis (LDA)	LDA reduziert die Dimensionalität, während es die Klassen separiert, indem es Merkmale findet, die die Klassentrennung maximieren.	Accuracy, F1-Score, AUC, Confusion-Matrix Erklärte Varianz

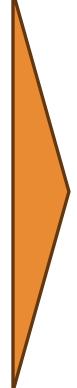
# ShortCut Tool-Box (3/3)

Auswahl

Lernstrategie	Einsatzbereich	Algorithmus	Beschreibung	Bewertungsmetrik
Unüberwachtes Lernen	Assoziations-analyse	Apriori	Identifiziert häufige Itemsets in Transaktionsdaten und erzeugt Assoziationsregeln zwischen ihnen.	Support, Confidence, Lift
	Clustering	K-Means	Teilt Daten in k Cluster auf, basierend auf der Nähe zu den k-Mittelpunkten.	Silhouetten-Koeffizient
	Clustering	DBSCAN	Ein dichtebasierter Clustering-Algorithmus, der Cluster basierend auf der Dichte der Datenpunkte identifiziert. Geeignet für Daten mit komplexen Strukturen.	Silhouetten-Koeffizient DB-Index
	Anomalie-erkennung	Isolation Forest	Ein Algorithmus zur Anomalieerkennung, der Datenpunkte isoliert, indem er sie zufällig in Entscheidungs-bäume unterteilt. Anomalien sind leichter zu isolieren.	Anomalie-Koeffizient
	Dimensions-reduktion	Principal Component Analysis (PCA)	PCA reduziert die Dimensionalität von Daten, indem es neue unkorrelierte Variablen findet, die die meiste Varianz erfassen.	Erklärte Varianz

# Algorithmus & Modell

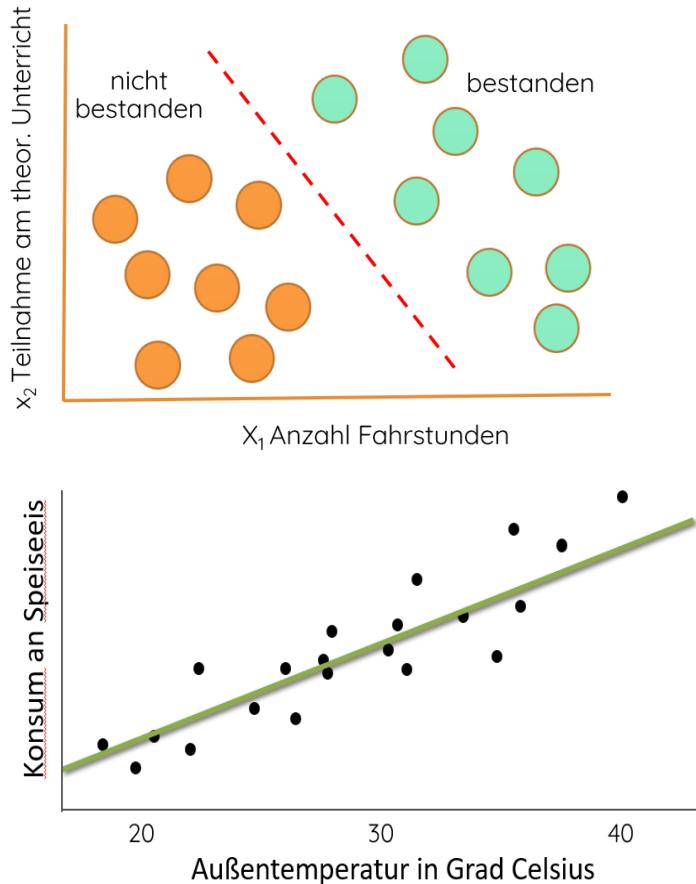
---

- Ein **Algorithmus** ist eine präzise, **wohldefinierte Prozedur** oder eine Reihe von Regeln und Schritten, die ausgeführt werden, um eine spezifische Aufgabe oder ein Problem zu lösen.
  - Algorithmen sind unabhängig von der Implementierung und können in natürlicher Sprache, Pseudocode oder jeder Programmiersprache ausgedrückt werden.
- 
- Ein **Modell** des maschinellen Lernens ist das **Ergebnis des Lernprozesses**, was ein Algorithmus aus den Daten gelernt hat.
  - Es repräsentiert die erkannten Muster, Beziehungen oder Strukturen basierend auf den Eingabedaten und kann für Vorhersagen oder Entscheidungen über neue, unbekannte Daten verwendet werden.
  - Ein Modell wird durch einen Algorithmus erstellt, der auf einem Datensatz trainiert wird, um spezifische Aufgaben wie Klassifizierung, Regression oder Clustering zu erfüllen.

# Supervised Learning

# Supervised Learning

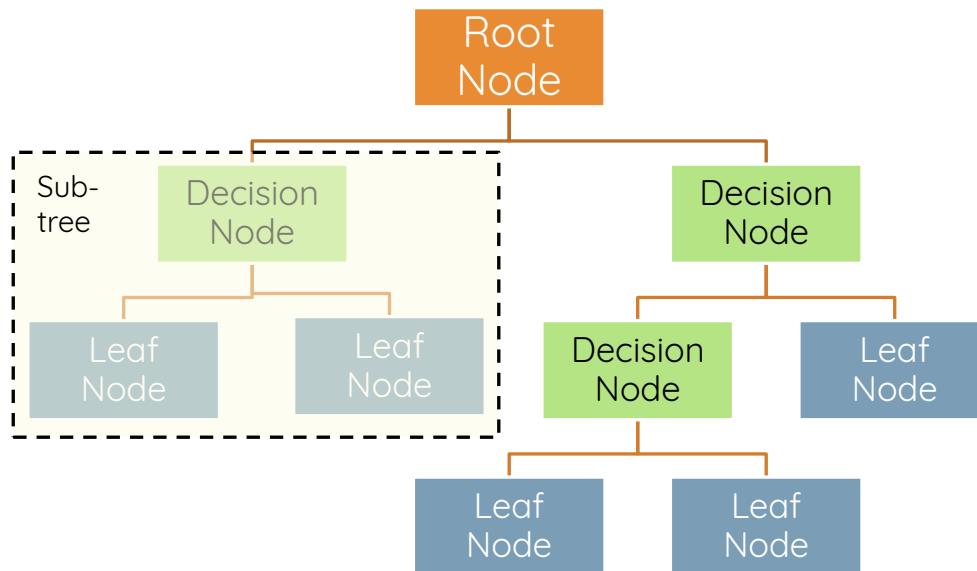
---



- Beim Supervised Learning (überwachtem Lernen) werden i.W. zwei Aufgaben unterschieden:
- **Klassifizierung:** Ein Klassifizierungsmodell sagt **kategorial Werte** voraus.
- Beispiel: Ist eine bestimmte E-Mail-Nachricht Spam oder kein Spam? Besteht ein Fahrschüler seine Prüfung?
- **Regression:** Ein Regressionsmodell sagt stetige, **numerisch Werte** voraus.
- Beispiel: Was ist der Wert einer Immobilie? Wie hoch ist der Konsum an Speiseeis bei einer Außentemperatur von 29°?

# Decision Tree

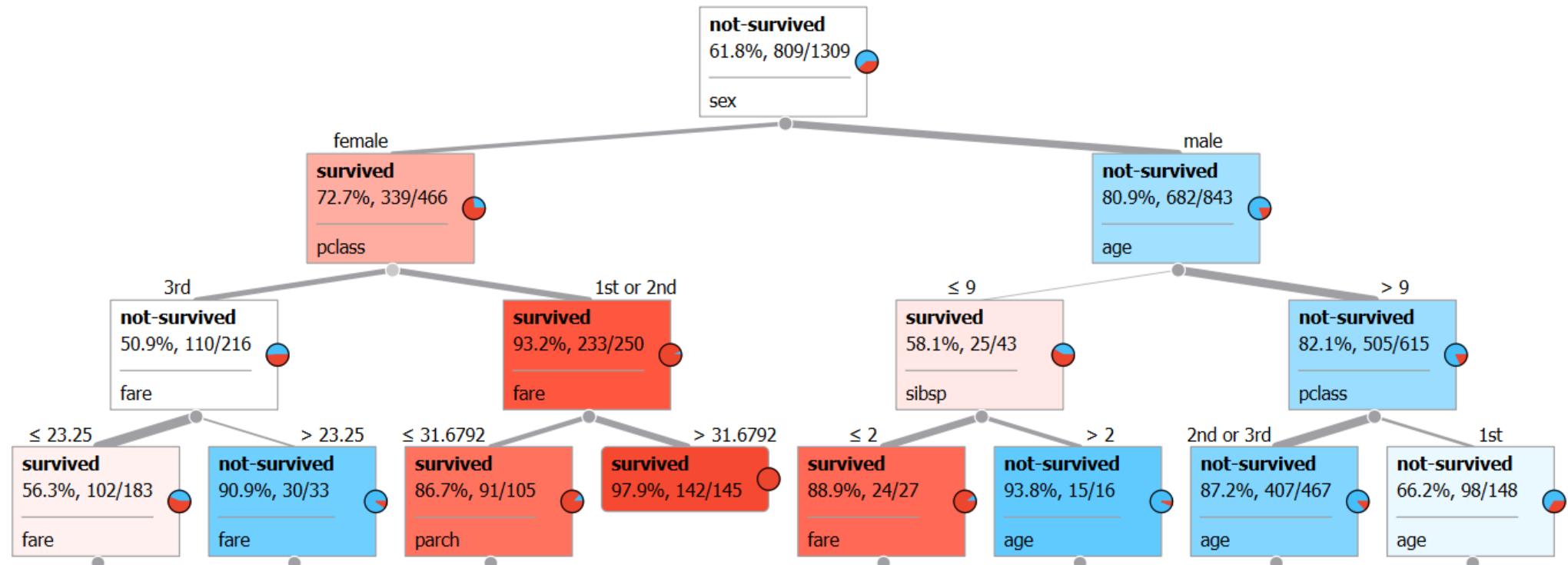
---



- Lernalgorithmus beim überwachten maschinellen Lernen
- Aufgabengruppe: Klassifizierung, Regression
- Aus dem DataSet (Training) wird eine hierarchische Struktur von möglichst wenigen Regeln abgeleitet
- Ausgehend von der „Wurzel“ erfolgt eine regelbasierte Ableitung der Lösung des Entscheidungsproblems.
- Auf jeder Hierarchieebene werden regelbasiert Verzweigungen anhand der verfügbaren Merkmale durchgeführt (Knoten).
- Für die Regel wird das Merkmal benutzt, dass zu der „best“ möglichen Aufteilung führt.
- Die best-mögliche-Aufteilung wird anhand der Entropie (Maß für die Unsicherheit, Zufälligkeit oder Unordnung) ermittelt.

# Decision Tree - Titanic

---



# Decision Trees – Split Numerical Attributes

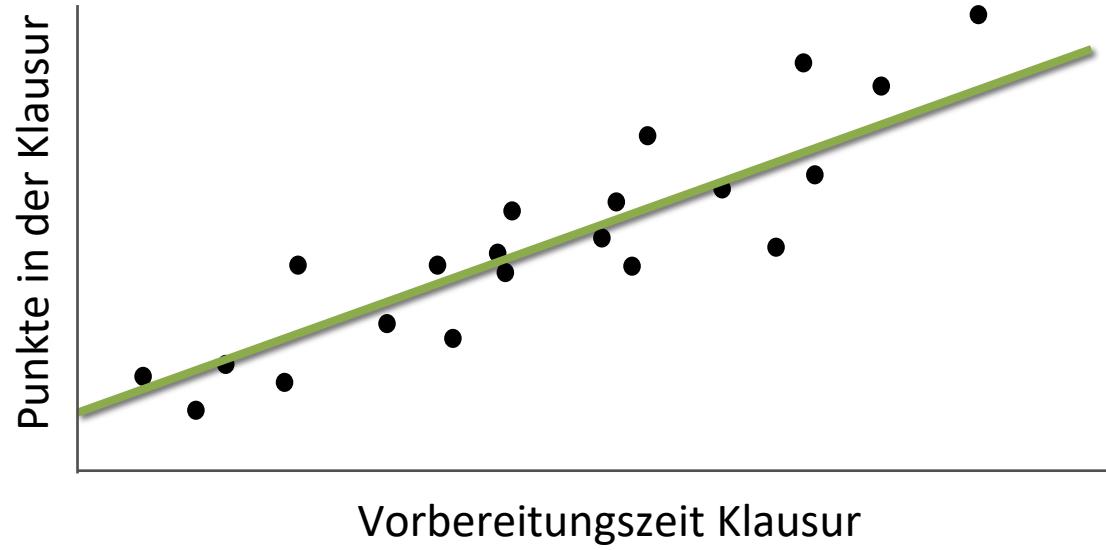
---

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No		
Taxable Income												
Sorted Values →	60	70	75	85	90	95	100	120	125	220		
Split Positions →	55	65	72	80	87	92	97	110	122	172	230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	1	2	2	1	3	0
No	0	7	1	6	2	5	3	4	3	4	4	3
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	0.420	

Quelle: [maschinelles Lernen - Entscheidungsbaum: Wo und wie wird ein Attribut in einem numerischen Datensatz aufgeteilt? - Kreuzvalidiert \(stackexchange.com\)](#), abgefragt: 10.12.2022

# Linear Regression

---



- Die lineare Regression ist ein statistisches Verfahren, mit dem eine beobachtete abhängige Variable (y) durch eine oder mehrere unabhängige Variablen (x) erklärt wird.
- Bei der linearen Regression wird dabei ein linearer Zusammenhang zwischen der abhängigen und den unabhängigen Variablen angenommen.
- Der lineare Zusammenhang kann als Gerade  $y = b + ax$  mit  $b$  = Ursprung,  $a$  = Steigung dargestellt werden.
- Beispiel: Zeit der Vorbereitung (y) und erzielte Punkte in der Klausur (x).



[KNIME Linear Regression](#)



[KNIME Behind the Scenes](#)

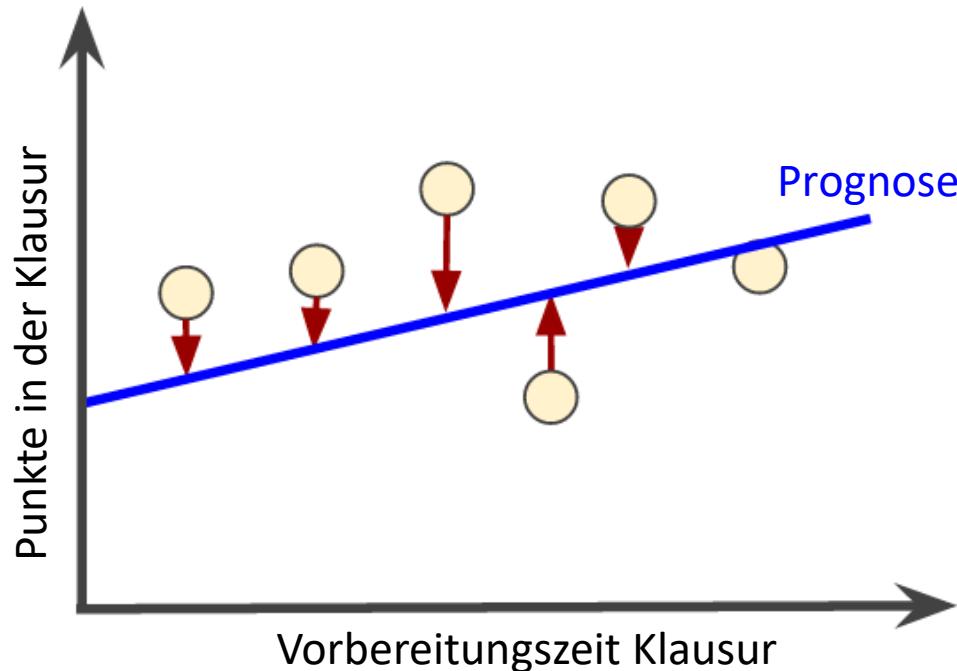


[StatQuest Linear Regression](#)

Nach: [Lineare Regression - Wikiwand](#)

# Prognosefehler / Loss

---



- Als Loss (Verlust) wird die Abweichung zwischen dem Ist-Wert ( $y$ ) und einer Vorhersage ( $y_{\text{pred}}$ ) bezeichnet.
- Das heißt, Loss ist eine Zahl, die angibt, wie schlecht die Vorhersage des Modells in einem einzelnen Beispiel war.
- Wenn die Vorhersage des Modells perfekt ist, beträgt der Verlust null; andernfalls ist der Verlust größer.
- Das Ziel beim Trainieren eines Modells besteht darin, eine Gerade zu finden, die in allen Beispielen im Durchschnitt einen geringen Loss (rote Pfeile) aufweisen.

Nach Bild-Quelle: [Descending into ML: Training and Loss | Machine Learning Crash Course \(google.com\)](#)

# Ansätze zum Trainieren von Modellen

Eigenschaften	Analytische Optimierung	Iterative Optimierung
Lösungsart	Direkte Ermittlung	Schrittweise Anpassung
Methode, Beispiel	Methode der kleinsten Quadrate	Gradient Descent
Algorithmus, Beispiel	Lineare Regression	Gradient Boosting
Geschwindigkeit	Schnell bei einfachen Modellen	Schnell bei großen und komplexen Modellen
Effizienz	Effizient bei einfachen Problemen	Effizienter bei großen und komplexen Problemen
Eignung für nichtlineare Probleme	Ungeeignet	Geeignet
Konvergenzgarantie	Garantierte Konvergenz zur optimalen Lösung bei richtiger Modellierung	Konvergenz zu lokalen Minima möglich, aber Möglichkeiten zur Verbesserung vorhanden
Abhängigkeit von der Wahl der Startwerte	Gering	Hoch
Anwendungsbereich	Wissenschaft, Mathematik, Ingenieurwissenschaften, Wirtschaft	Machine Learning, Data Science, Big Data

 [StatQuest The Chain Rule](#)  
 [StatQuest Gradient Descent](#)

# Gradient Descent

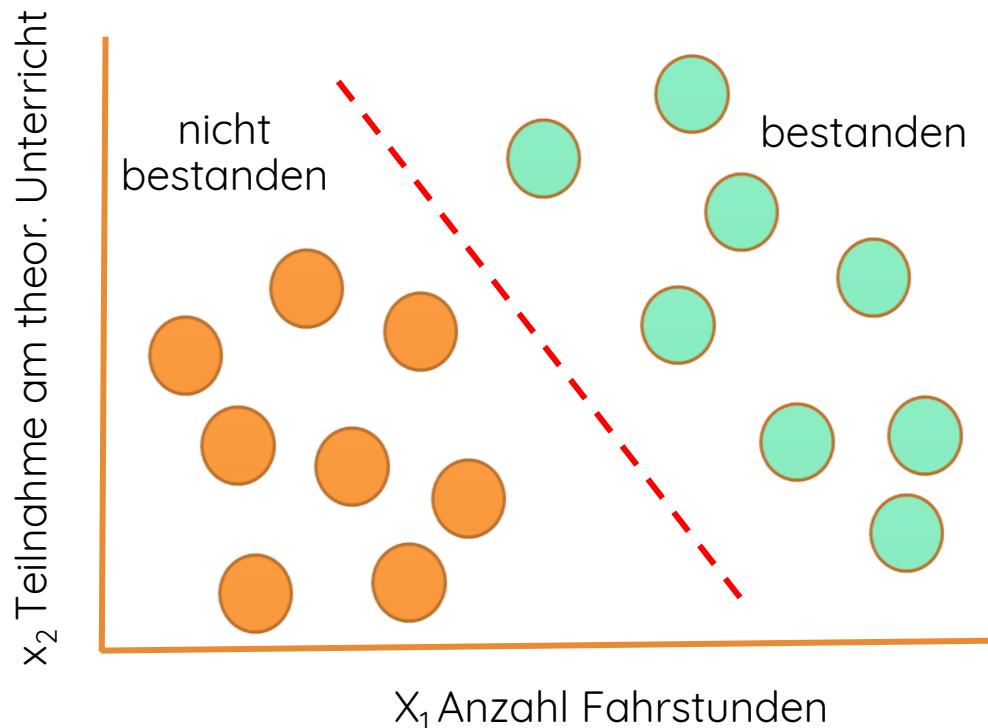


Stellen Sie sich vor, Sie sind auf einem großen Berg und möchten den tiefsten Punkt finden, also das Tal. Gradient Descent ist wie ein schlauer Plan, um dorthin zu gelangen. Sie schauen immer, in welche Richtung es am steilsten bergab geht, und machen dann einen Schritt in diese Richtung. Diesen Schritt wiederholen Sie immer wieder, bis Sie im Tal angekommen sind. So finden Sie den tiefsten Punkt des Berges! In der Mathematik benutzen wir das, um die beste Lösung für ein Problem zu finden.

-  [KNIME Gradient Descent](#)  [StatQuest Stochastic Gradient Descent](#)
-  [StatQuest Gradient Descent, Step by Step](#)

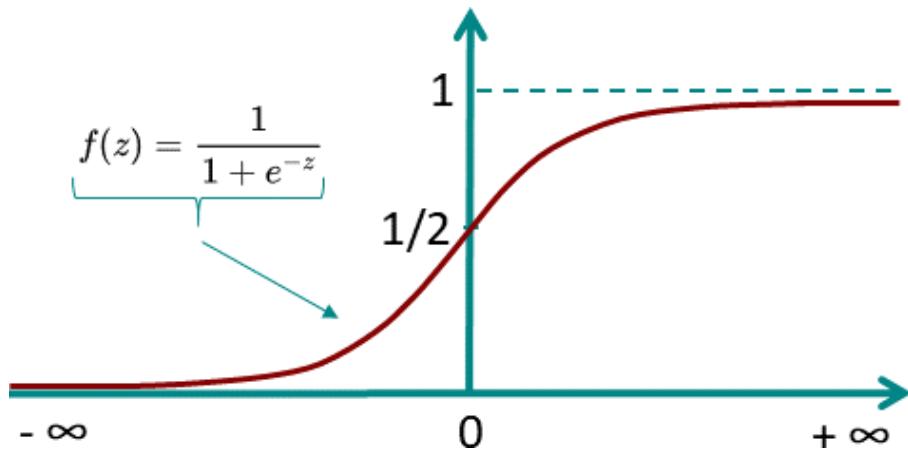
# Logistic Regression

---



# Von der linearen zur logistischen Regression

---



$$P(y = 1|x_1, \dots, x_n) = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \dots + b_k \cdot x_k + a)}}$$

$$P(y = 0|x_1, \dots, x_n) = 1 - \frac{1}{1 + e^{-(b_1 \cdot x_1 + \dots + b_k \cdot x_k + a)}}$$

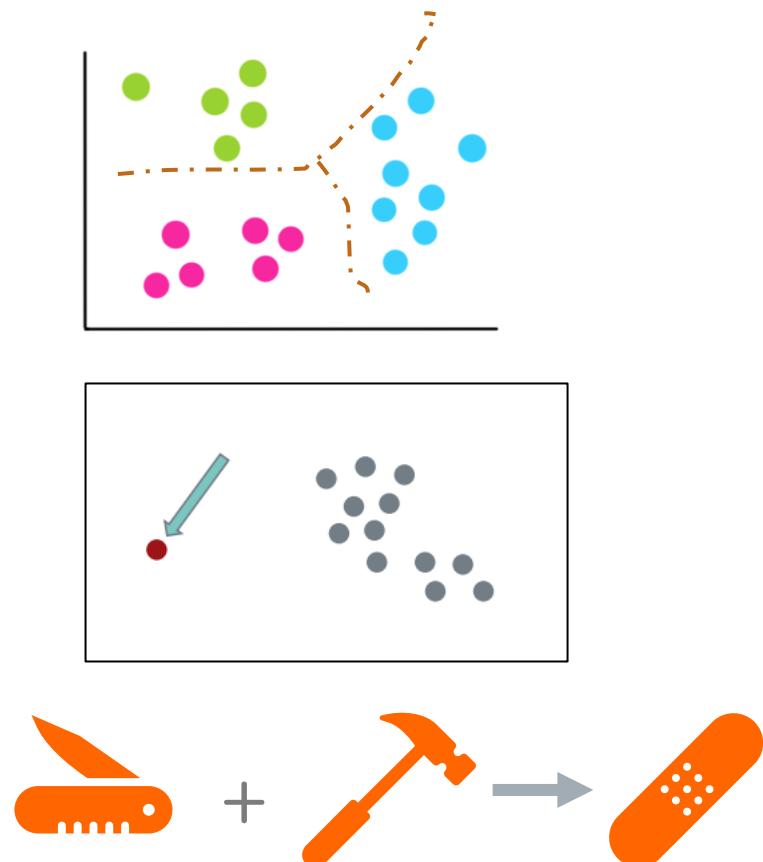
- Mit der linearen Regression können Zahlenwerte vorhergesagt werden.
- Sie kann jedoch keine binären Fragen abdecken.
- Die lineare Funktion muss zu einer S-Kurve umgewandelt werden, bei der nur Ergebnisse zwischen 0 und 1 erlaubt sind.
- Die Umwandlung kann z.B. durch die sog. Sigmoid-Funktion erfolgen.
- Die Sigmoid-Funktion wandelt die Ergebnisse so um, dass nur ein Ergebnis zwischen 0 und 1 herauskommen kann.

Quelle: [Logistische Regression – DATAtab](#)

# Unsupervised Learning

# Unsupervised Learning

---



- Das Ziel des unüberwachten Lernens ist es, versteckte Strukturen bzw. Muster in unmarkierten Daten zu finden.
- Im Regelfall werden drei Aufgabengruppen unterschieden:
  - **Clustering/Segmentierung** - Verfahren zur Entdeckung von Ähnlichkeitsstrukturen in (meist relativ großen) Datenbeständen. Die so gefundenen Gruppen von „ähnlichen“ Objekten werden als Cluster bezeichnet.
  - **Anomalierekennung** - soll Datensätze identifizieren, die für die gesamte Datenbasis untypisch sind.
  - **Assoziationsanalyse** - dient dem Auffinden von Zusammenhängen in transaktionsbasierten Datenbasen, die in Form sogenannter Assoziationsregeln dargestellt werden.

# Clustering

---

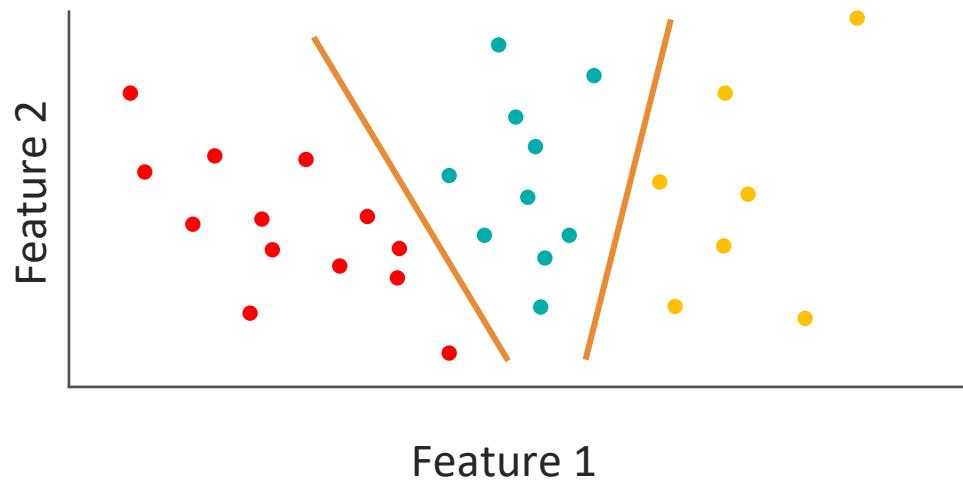


- Unter Clustering versteht man Verfahren zur Entdeckung von Ähnlichkeitsstrukturen in (meist relativ großen) Datenbeständen.
- Man kann u.a. folgende Verfahren unterscheiden:
- **Partitionierende Clusteranalyse** (z.B. KMeans): Es wird zunächst die Zahl der Cluster  $k$  festgelegt. Dann werden  $k$  Clusterzentren bestimmt und diese iterativ so lange verschoben, bis sich die Zuordnung der Beobachtungen zu den  $k$  Clusterzentren nicht mehr verändert.
- **Bei dichtebasierterem Clustering** (z.B. DBScan) werden Cluster als Objekte betrachtet, die dicht beieinander liegen, getrennt durch Gebiete mit geringerer Dichte.

Foto von [Eva Bronzini](#):

# Clustering - K-Means

---



- K-Means-Clustering ist ein einfacher Ansatz zum Partitionieren eines Datensatzes in K verschiedene, nicht überlappende Cluster.
- Der Algorithmus ist eine der am häufigsten verwendeten Techniken zur Gruppierung von Objekten, da er schnell die Zentren der Cluster findet.
- Um K-Means-Clustering durchzuführen, müssen zuerst die gewünschte Anzahl von Clustern K spezifizieren werden.
- Dabei bevorzugt der Algorithmus Gruppen mit geringer Varianz und ähnlicher Größe.



[KNIME Clustering](#)

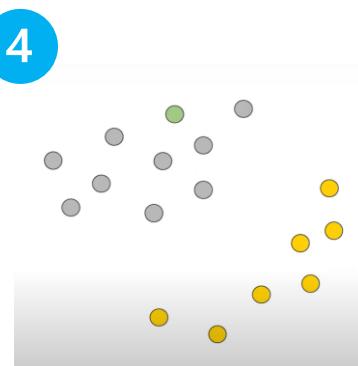
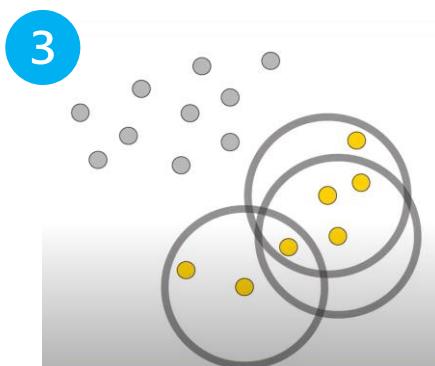
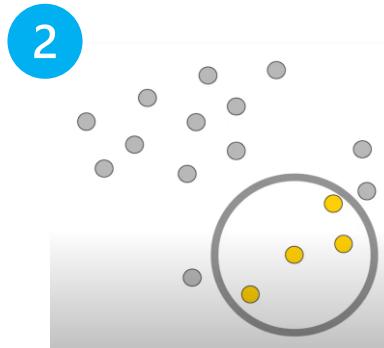
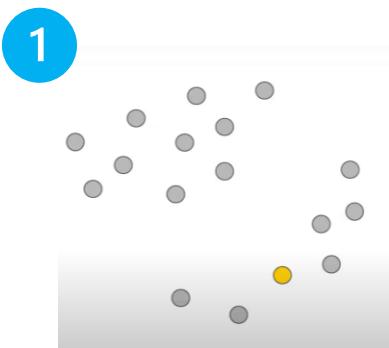


[KNIME Training Clustering Algorithm](#)



[StatQuest: K-means clustering](#)

# Clustering - DBSCAN



- DBSCAN (Density-Based Spatial Clustering of Applications with Noise, etwa: Dichtebasierter räumliche Clusteranalyse mit Rauschen) ist ein Algorithmus zur Clusteranalyse.
- Der Algorithmus arbeitet dichtebasiert und ist in der Lage, mehrere Cluster zu erkennen.
- Rauschpunkte werden dabei ignoriert und separat zurückgeliefert.
- Dabei bevorzugt der Algorithmus Gruppen mit geringer Varianz und ähnlicher Größe.

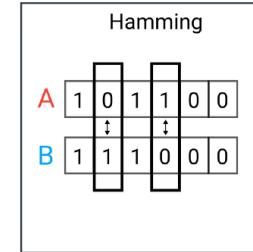
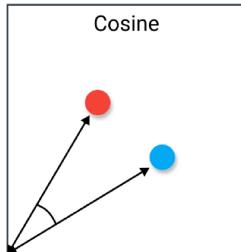
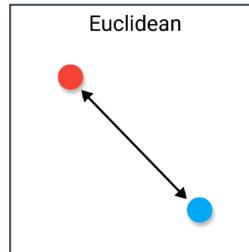
Bilder-Quelle: [KNIME Clustering](#) [KNIME Clustering](#) [KNIME Training Clustering Algorithm](#) [StatQuest DBScan](#)

# Distanzmessung

---

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Euclidean distance

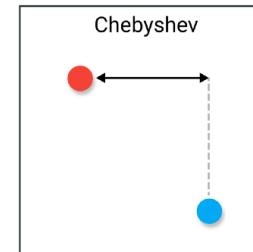
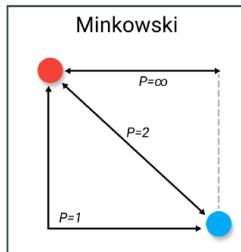
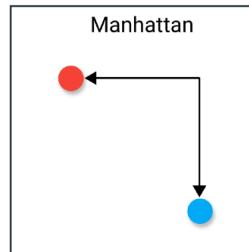


$$D(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|}$$

Cosine Similarity

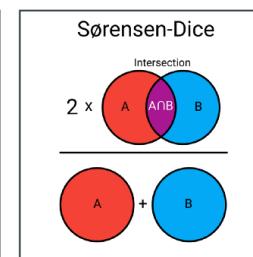
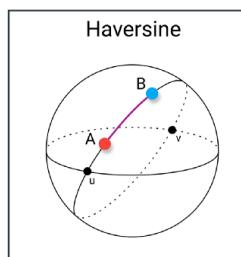
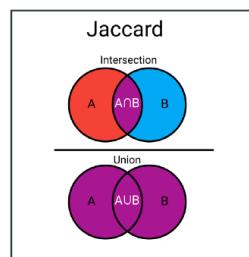
$$D(x, y) = \sum_{i=1}^k |x_i - y_i|$$

Manhattan distance



$$D(x, y) = \max_i (|x_i - y_i|)$$

Chebyshev distance



$$D(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Minkowski distance

Bilder-Quelle: [9 Distance Measures in Data Science | Towards Data Science \(medium.com\)](https://towardsdatascience.com/9-distance-measures-in-data-science-13a2f3a3e3d)

# Anomaly/Deviation – Typen

---

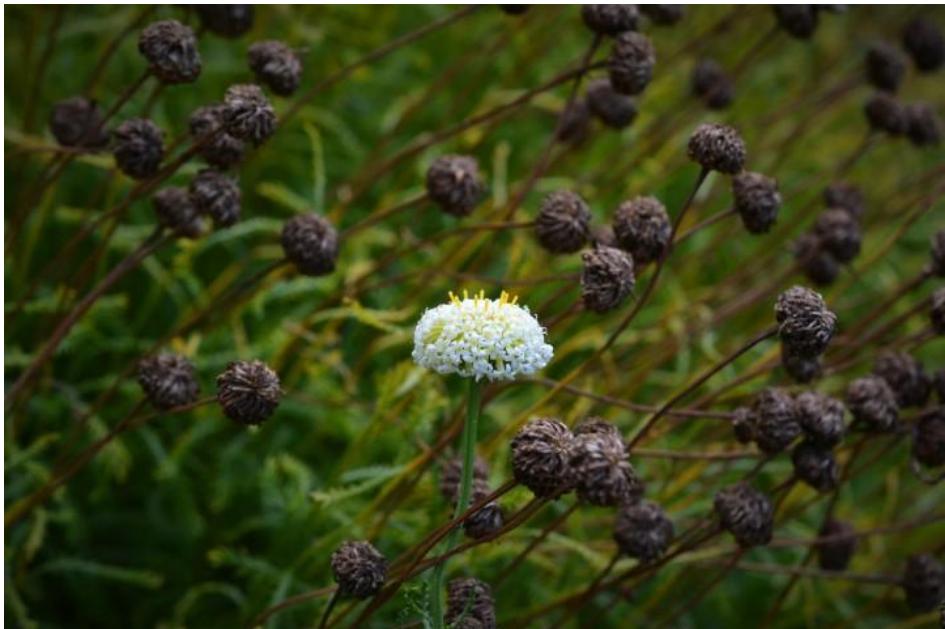
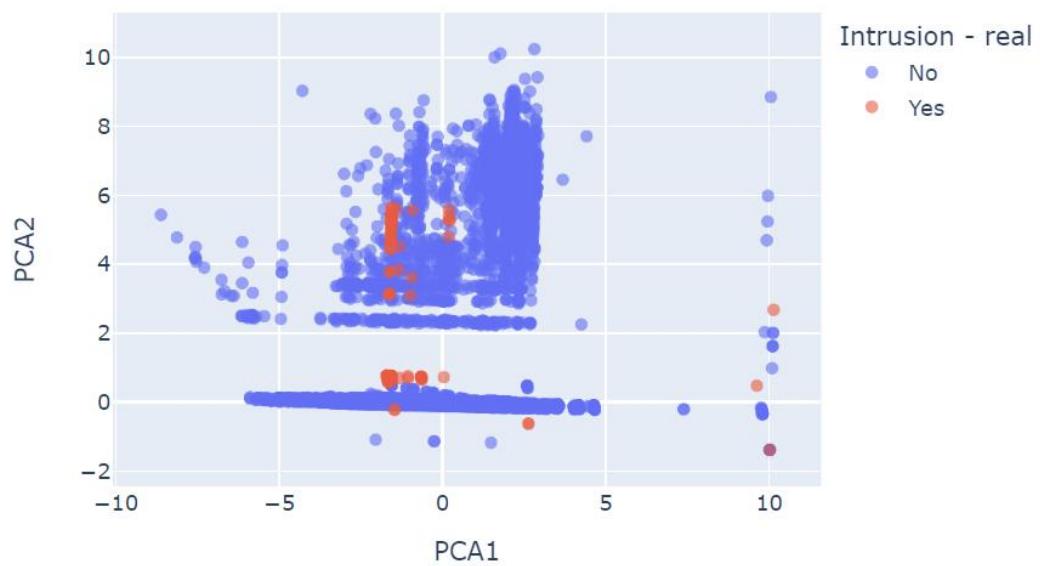


Bild von [Anemone123](#) auf [Pixabay](#)

- Anomalien sind Abweichungen von einem erwarteten oder normalen Zustand. Grundsätzlich lassen sich Anomalien in drei Arten einteilen:
- **Punkt/globale Anomalien**  
ein einzelner Datenpunkt, der in Bezug auf den Rest der Daten als anomali zu klassifizieren ist
- **Kontextuelle Anomalien**  
ein Datenpunkt, der nur in einem bestimmten Kontext anomali erscheint, Beispiel: Temperatur von +30° im Winter.
- **Kollektive Anomalien**  
eine Menge verwandter Datenpunkte ist anomali zum Rest, Beispiel: Daten einer Kreditkarte zeigen, dass ein Kauf in den USA zur gleichen Zeit in Frankreich getätigt wurde.

# Anomalies/Deviation - Isolation Forest

Network-Intrusion-Detektor - Real

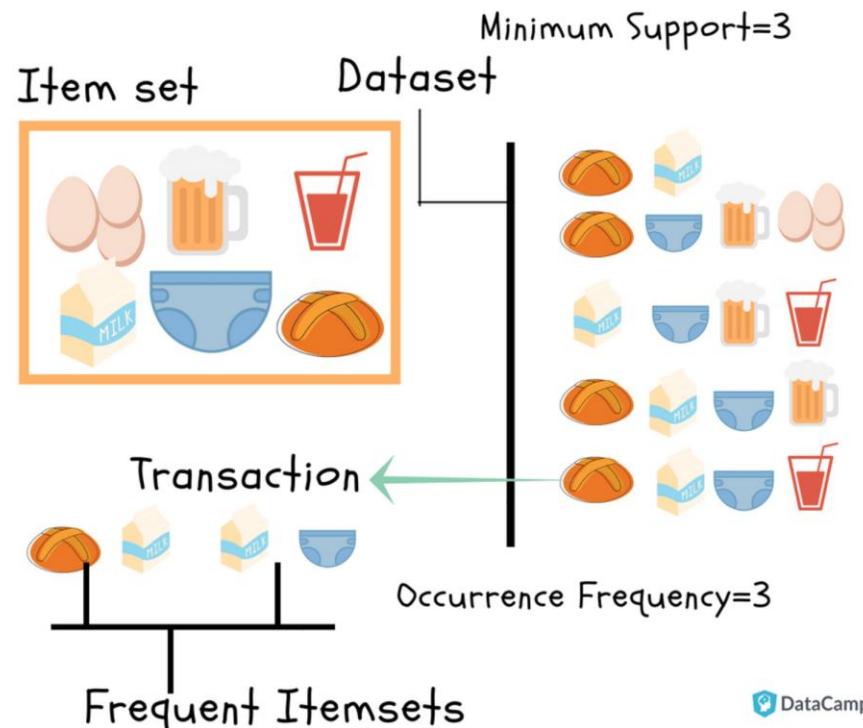


- Isolation Forest ist ein unbeaufsichtigter Algorithmus für maschinelles Lernen zur Erkennung von Anomalien.
- Wie der Name schon sagt, ist Isolation Forest eine Ensemble-Methode (ähnlich wie Random Forest), es verwendet den Durchschnitt der Vorhersagen mehrerer Entscheidungsbäume.
- Annahmen von Isolation Forest :
  - Die Anomalien gehören zu einer Minderheitsklasse im Vergleich zu den normalen Daten, die zahlreicher sind.
  - Die Anomalien werden tendenziell schnell mit dem kürzesten Durchschnittsweg (Entfernung Wurzel-Blatt) gefunden.



[Isolation Forest for Outlier Detection within Python](#)

# Association: Apriori

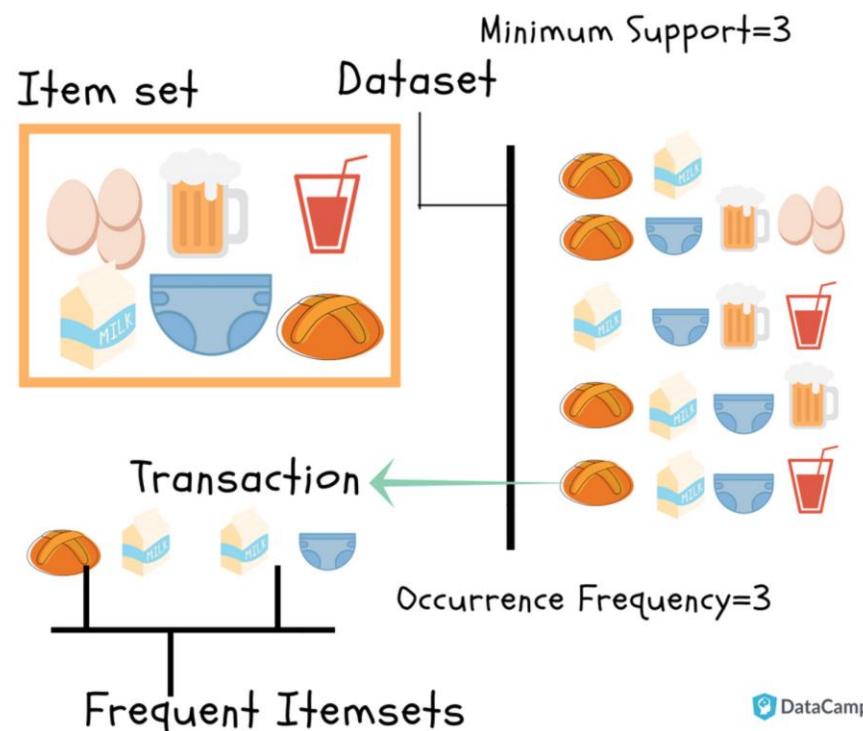


Quelle: <https://www.datacamp.com/>

- Der Apriori-Algorithmus ist ein Verfahren zur Assoziationsanalyse.
- Er dient der Auffindung von Zusammenhängen in transaktionsbasierten Datenbasen, die in Form sogenannter Assoziationsregeln dargestellt werden.
- Eine häufige Anwendung des Apriori-Algorithmus ist die Warenkorbanalyse.
- Items sind hierbei angebotene Produkte und ein Einkauf stellt eine Transaktion dar, welche die gekauften Items enthält.
- Beispiel: Liste der von Kunden gekauften Artikel, Angaben zu häufig besuchten Websites usw.

[KNIME Association – Apriori](#)

# Apriori – Weitere Anwendungsbiete



Quelle: <https://www.datacamp.com/>

- Ausbildung  
Data Mining kann „individuelle“ Vorschläge zum Kursprogramm machen. Basis absolvierte Kurse und aktuelle Performance.
- Forstwirtschaft  
Häufigkeit und Intensität von Waldbrandanalysen anhand von Waldbranddaten.
- Autocomplete  
Apriori wird von einer Reihe von Firmen eingesetzt, darunter das Empfehlungssystem von Amazon und das Autocomplete-Tool von Google.

 [KNIME Association – Apriori](#)

# Support, Confidence and Lift

{A, B, F} → H

■ support  $s = \frac{freq(A,B,F,H)}{N}$

■ confidence  $c = \frac{freq(A,B,F,H)}{freq(A,B,F)}$

■ lift  $I = \frac{support(\{A,B,F\} \Rightarrow H)}{support(A,B,F) \times support(H)}$



Wahrscheinlichkeit, dass das Itemset {A, B, F, H} in allen Transaktionen vorkommt.

relative Häufigkeit des Itemsets {A, B, F, H}, wenn man das letzte Item (consequenz) aus dem Set entfernt

Wahrscheinlichkeit des Vorkommens der Itemmenge im Vergleich zum Zufall.  
Lift = 1: Itemsets sind unabhängig;  
Lift >1: Stärke der Abhängigkeit von H

Support, confidence und lift können als Schwellwerte genutzt werden

# A-Priori Algorithm: Beispiel

- Betrachten wir Milch, Windel und Bier: {milk, diaper}⇒beer
- Wie oft werden sie zusammen in allen Warenkörben gefunden?
- Wie oft findet man sie zusammen in allen Einkaufskörben mit den Vorläufern?

TID	Transactions
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

support

$$s \frac{P(\text{milk, diaper, beer})}{|T|} = \frac{2}{5} = 0.4$$

confidence

$$c = \frac{P(\text{milk, diaper, beer})}{P(\text{milk, diaper})} = \frac{2}{3} = 0.67$$

# A-priori Algorithm: Beispiel

---

- Betrachten wir Milch, Windel und Bier:  $\{milk, diaper\} \Rightarrow beer$
- Wie oft sind sie zusammen in allen Shopping Baskets zu finden?
- Wie oft findet man *beer* zusammen in allen Einkaufskörben mit den Vorläufern *milk, diaper*?

TID	Transactions
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

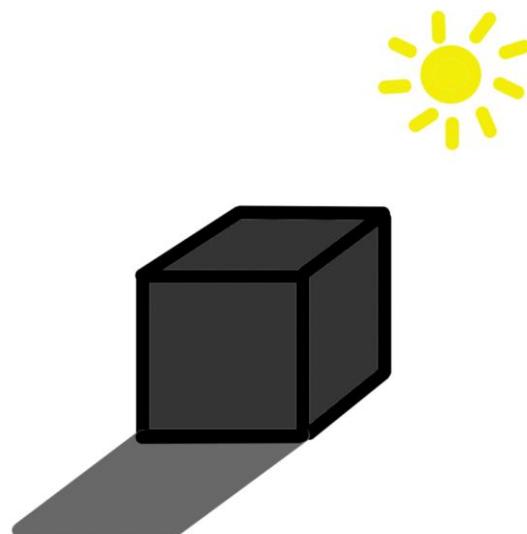
$$s(milk, diaper) = \frac{P(milk, diaper)}{|T|} = \frac{3}{5} = 0.6$$

$$s(beer) = \frac{P(beer)}{|T|} = \frac{3}{5} = 0.6$$

$$lift = \frac{s(milk, diaper, beer)}{s(milk, diaper) \times s(beer)} = \frac{0.4}{0.6 \times 0.6} = 1.11$$

# Dimensionsreduktion durch Projektion

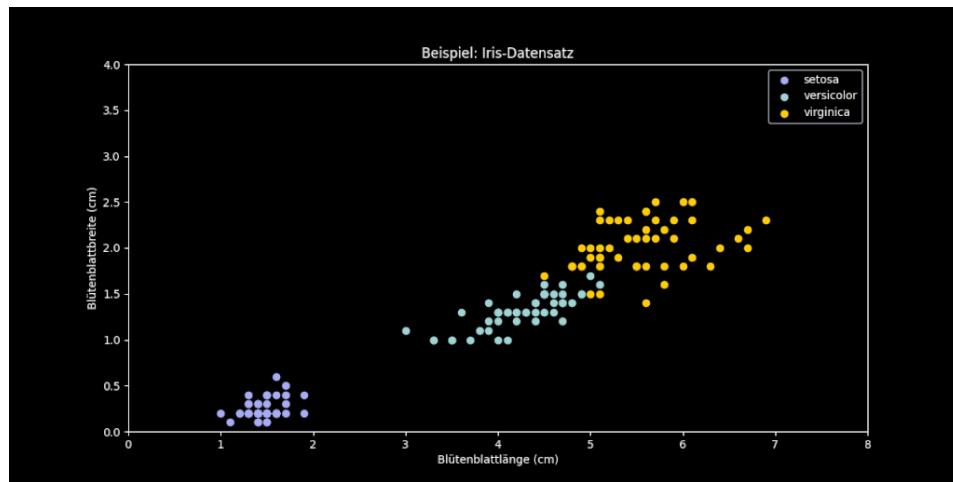
---



- Die Projektion ist eine Funktion, die Datenpunkte so verändert, dass sie mit weniger Komponenten beschreibbar sind, ähnlich einem Schattenwurf.
- Ein dreidimensionaler Punkt kann als zweidimensionaler Punkt in der Ebene dargestellt werden, auf die der Schatten geworfen wird.
- Dieses Konzept wird in der Hauptkomponentenanalyse angewendet, um die Dimensionalität von Daten zu reduzieren.
- Ein Beispiel ist der Schatten eines Würfels, der durch Projektion der dreidimensionalen Punkte des Würfels auf eine zweidimensionale Ebene entsteht.

Quelle: [PCA - Eine Methode zur Dimensionsreduktion »](#)  
[Lamarr-Institut \(lamarr-institute.org\)](http://lamarr-institute.org)

# Principal Component Analysis (PCA)



Quelle: PCA - Eine Methode zur Dimensionsreduktion » Lamarr-Institut ([lamarr-institute.org](http://lamarr-institute.org))

- Die Hauptkomponentenanalyse (PCA) ist eine Methode zur Dimensionsreduktion, die Datenpunkte in einen Unterraum mit weniger Dimensionen projiziert.
- Dieser Unterraum wird so gewählt, dass die Varianz der Datenpunkte maximal ist, was eine gute Separierbarkeit der verschiedenen Klassen ermöglicht.
- Bei einem Unterraum mit der Dimension 1 wird eine Gerade durch die Daten gelegt, welche die Varianz der (orthogonalen, rechtwinkligen) projizierten Datenpunkte maximiert.
- Diese Gerade repräsentiert den eindimensionalen Unterraum, auf den die PCA die Datenpunkte projiziert.
- Dimensionen des Unterraums lassen sich erweitern.
- Die  $k$  Koordinatenachsen (auch Hauptkomponenten genannt) werden analog gebildet.
- Die Summe der Varianz aller Hauptkomponenten liefert die im Unterraum repräsentierte Varianz.

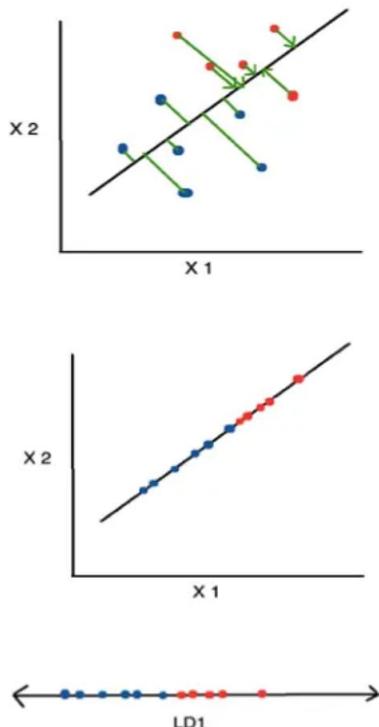
# Vergleich LDA vs PCA

---

LDA	PCA
Supervised learning	Unsupervised learning
Ziel: Maximierung der Klassenunterschiede	Ziel: Maximierung der Varianz
Verwendet Label-Informationen (Klassen)	Verwendet keine Label-Informationen
Reduziert die Dimensionen von Features	Reduziert die Dimensionen von Features
Kann für die Klassifizierung verwendet werden	Kann für die Datenkompression verwendet werden
Kann Overfitting verursachen	Vermeidet Overfitting
Bietet keine Interpretierbarkeit für die Dimensionen	Bietet Interpretierbarkeit für die Dimensionen

# Transformation - LDA

---



- Die **Linear Discriminant Analysis** (lineare Diskriminanzanalyse) ist eine Technik zur Reduzierung der Datendimensionalität.
- LDA ist wie PCA, konzentriert sich jedoch auf die Maximierung der Trennbarkeit zwischen bekannten Kategorien.
- Die grundsätzliche Idee besteht darin, einen linearen Entscheidungsgrenzwert zwischen verschiedenen Klassen von Datenpunkten zu finden, um sie optimal voneinander zu trennen.
- Die Methode sucht nach einer linearen Kombination der Merkmale, die die größte Varianz zwischen den Klassen und die kleinste Varianz innerhalb jeder Klasse aufweist. Mit anderen Worten, LDA maximiert den Abstand zwischen den Klassen und minimiert den Abstand innerhalb jeder Klasse.
- Sobald der Entscheidungsgrenzwert gefunden wurde, können neue Datenpunkte anhand dieser Grenzwerte klassifiziert werden.

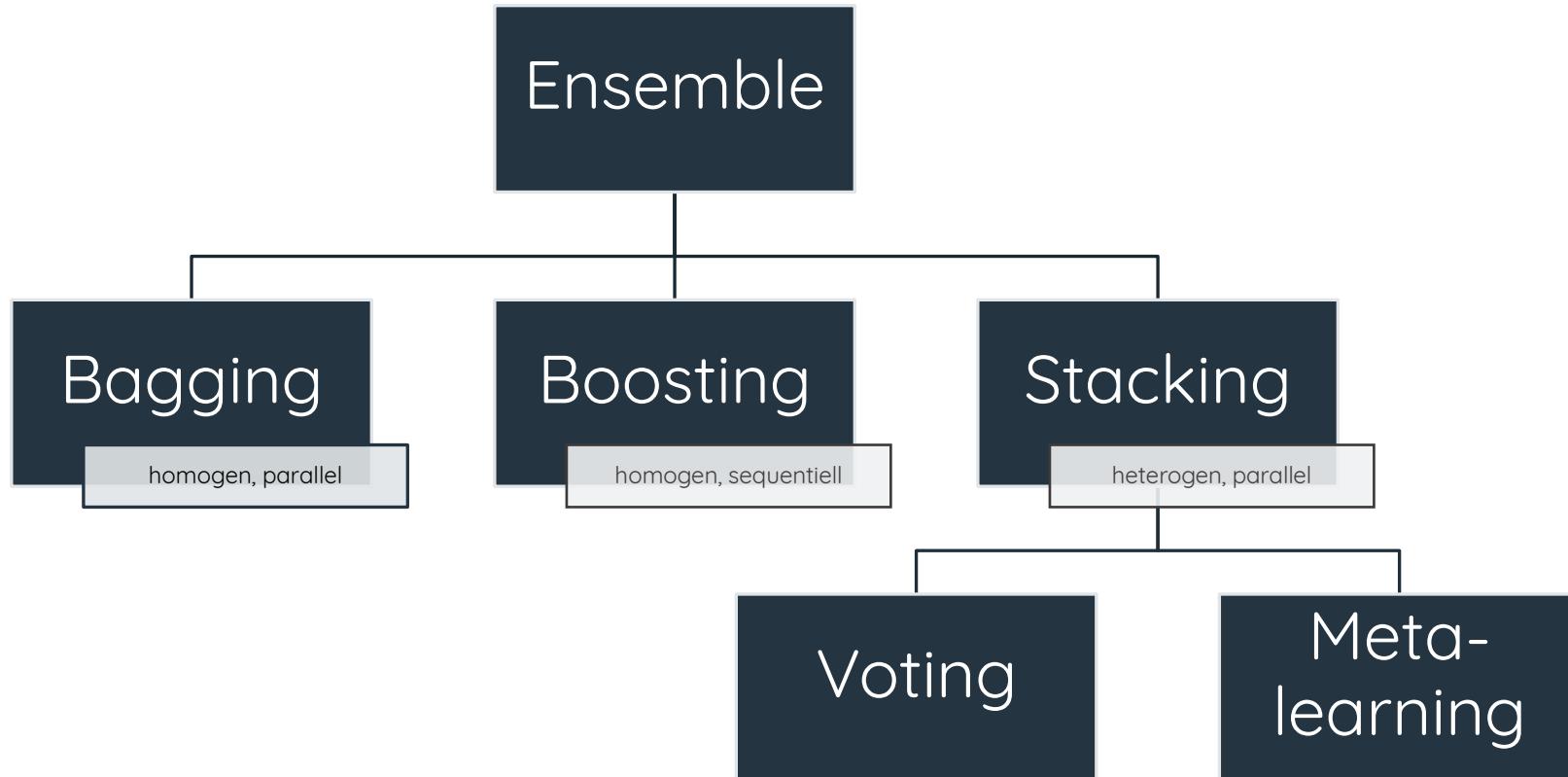


Quelle: [Was ist die lineare Diskriminanzanalyse \(LDA\)?](#)  
[\(knowledgehut.com\)](#)

# Ensemble

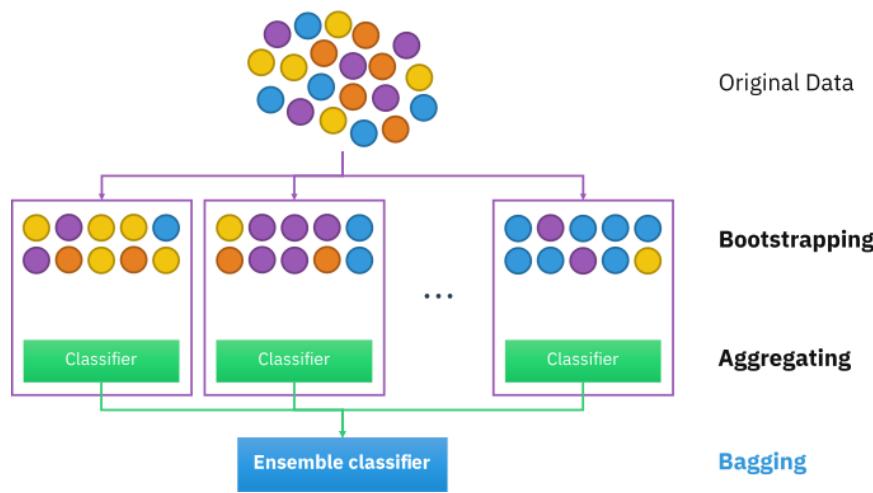
# Ensemble - Übersicht

**INFO BOX**  
Ensemble können auch im unsupervised Learning erstellt werden



# Ensemble - Bagging

---



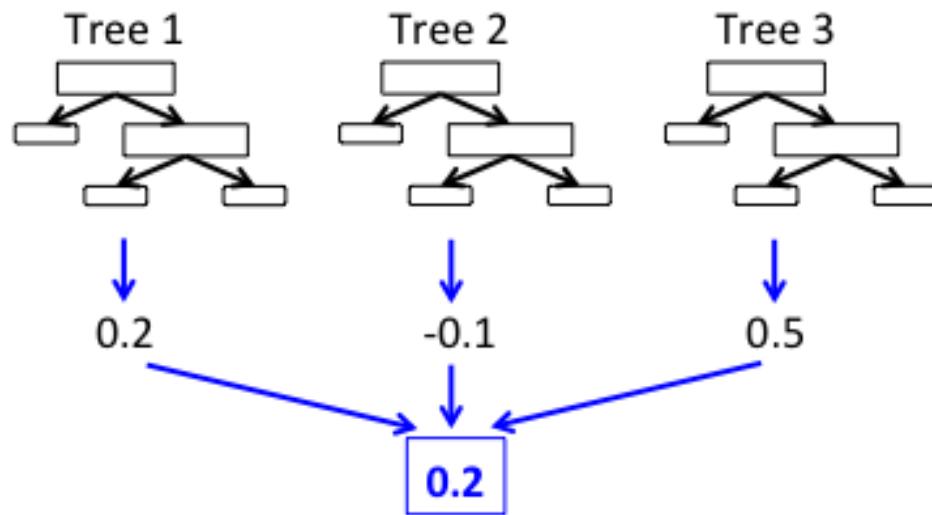
- Ensemble-Lernen ist ein Ansatz, bei dem mehrere Modelle miteinander kombiniert werden.
- Bagging (Bootstrap aggregating)
  - Es werden zufällige Stichproben von Daten für mehrere Modelle gebildet.
  - Anschließend werden diese Modelle unabhängig voneinander trainiert.
  - Regression/Klassifizierung ergibt sich als Mittelwert oder Median dieser Prognosen.

Bild-Quelle: [Bootstrap aggregating – Wikipedia](#)

 [KNIME Bagging & Boosting](#)

# Random Forest

---



- Random Forest ist eine Gruppe (Ensemble) von Entscheidungsbäumen.
- Mehrere Bäume werden in „zufälliger“ Weise aufgebaut und bilden einen Random Forest.
- Jeder Baum wird aus einer anderen Stichprobe von Daten (Bootstrap-Stichprobe) und Merkmalen erstellt werden.
- An jedem Knoten kann ein anderes Merkmal zum Teilen ausgewählt werden.
- Das Teilen kann anhand von kategorialen und numerischen Merkmalen erfolgen.
- Jeder der Bäume macht seine eigene individuelle Vorhersage.
- Diese Vorhersagen werden dann gemittelt, um ein einzelnes Ergebnis zu erzeugen.

Bild-Quelle: [Random Forests and Boosting in MLlib - The Databricks Blog](#)

[KNIME Random Forest](#) [KNIME Learner & Predictor](#) [StatQuest Random Forest #1 #2](#)

# Ensemble - Boosting

---



- Es werden mehrere, nacheinander geschaltete Modelle berechnet.
- Schrittweise wird die Leistung insgesamt gesteigert, indem die Fehler des vorherigen Modells reduziert werden.
- Vor jedem Durchlauf wird beurteilt, welche bislang erstellten Muster gut eingeordnet werden können.
- Diejenigen Muster, die noch nicht gut klassifiziert werden, werden im nächsten Durchlauf besonders stark berücksichtigt.
- Die unerkannten Muster bekommen ein höheres, die erkannten Muster ein geringeres Gewicht.

Bild-Quelle: [Bootstrap aggregating - Wikipedia](#)

 [KNIME Bagging & Boosting](#)

# Boosting

---

Das Grundprinzip: Mehrere schwache Modelle arbeiten nacheinander zusammen, wobei jedes neue Modell die Fehler der vorherigen korrigiert.

Ablauf in 4 Schritten:

- Modell 1 macht erste Vorhersagen (oft ungenau)
- Modell 2 lernt, wo Modell 1 falsch lag und korrigiert diese Fehler
- Modell 3 korrigiert die verbleibenden Fehler von Modell 1+2
- Finale Vorhersage = Summe aller Modellbeiträge

Beispiel:

- Modell 1: "Dieses Haus kostet 300.000€"  
(tatsächlich: 250.000€)
- Modell 2: "Modell 1 schätzt 50.000€ zu hoch, ich ziehe 45.000€ ab"
- Modell 3: "Immer noch 5.000€ zu hoch, ich ziehe 4.000€ ab"
- Ergebnis:  $300.000 - 45.000 - 4.000 = 251.000\text{€}$

Jedes neue Modell spezialisiert sich auf die verbleibenden Fehler, wodurch die Vorhersagen mit jedem Schritt präziser werden.

# XGBoost

---

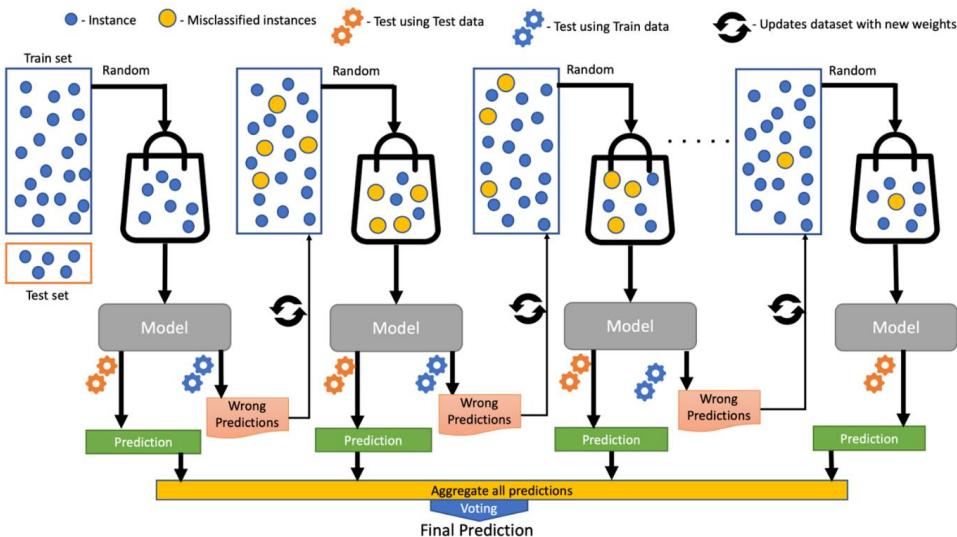


Bild-Quelle: [XGBoost: A Deep Dive into Boosting | by Rohan Harode | SFU Professional Computer Science | Medium](#)

- XGBoost (Extreme Gradient Boosting) ist ein Modell, dass auf der Boosting-Idee basiert.
- Beim Boosting werden mehrere Modelle zu einem Vorhersagemodell kombiniert.
- Bei der „Gradientenverstärkung“ wird ein einzelnes schwaches Modell „verstärkt“/ verbessert, indem es mit einer Reihe anderer Modelle kombiniert wird.
- Der Prozess der additiven Generierung von Modellen zielt darauf ab, dass die „Schwächen“ des aktuellen Modells beim nächsten Modell minimiert werden.
- XGBoost kann für Regressions- und Klassifizierungsmodelle verwendet werden.



[StatQuest XGBoost #1 Regression](#) [#2 Classification](#) [#3 Math Details](#) [#4 Crazy Cool Optimizations](#)

# XGBoost

---

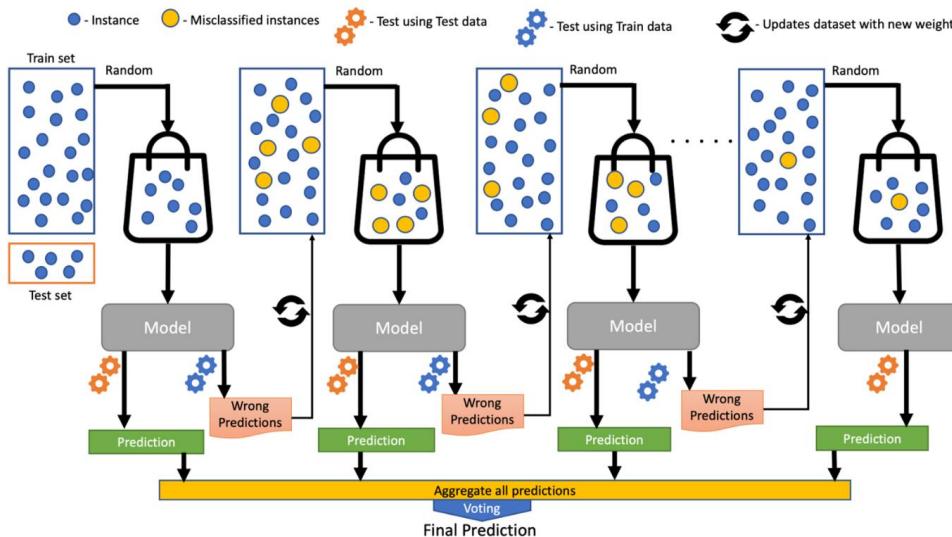


Bild-Quelle: [XGBoost: A Deep Dive into Boosting](#) | by Rohan Harode | SFU Professional Computer Science | Medium

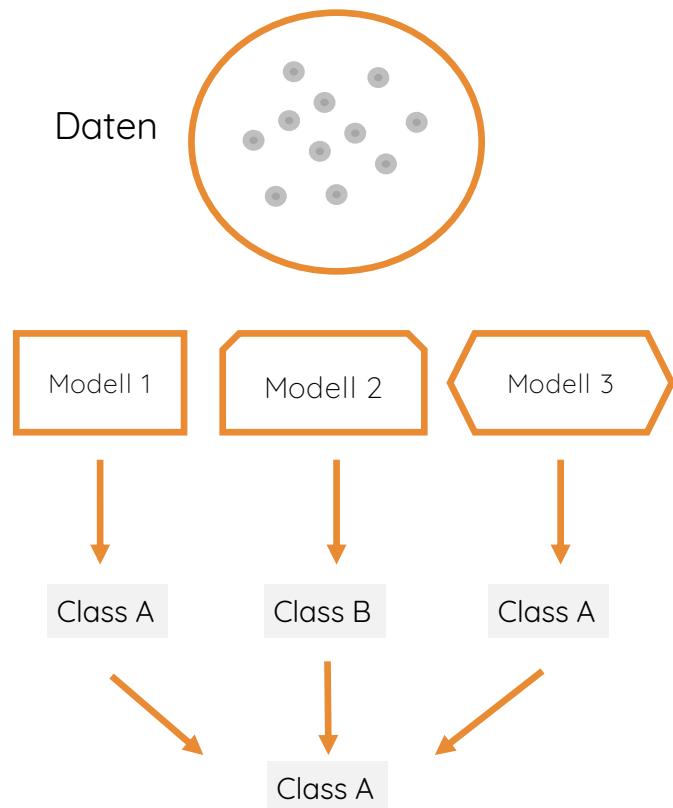
- XGBoost ist die Abkürzung für „eXtreme Gradient Boosting“.
- Das „eXtreme“ bezieht sich auf...
  - Geschwindigkeitsverbesserungen
  - paralleles Computing und
  - Cache-Awareness,
- die XGBoost etwa 10-mal schneller machen als ein herkömmliches Gradient Boosting.
- Darüber hinaus enthält XGBoost einen einzigartigen Split-Finding-Algorithmus zur Optimierung von Bäumen sowie eine integrierte Regularisierung, die eine Überanpassung reduziert.

[StatQuest XGBoost #1 Regression](#) [#2 Classification](#) [#3 Math Details](#) [#4 Crazy Cool Optimizations](#)

Quelle: [Getting Started with XGBoost in scikit-learn](#) | by Corey Wade | Towards Data Science

# Stacking - Voting

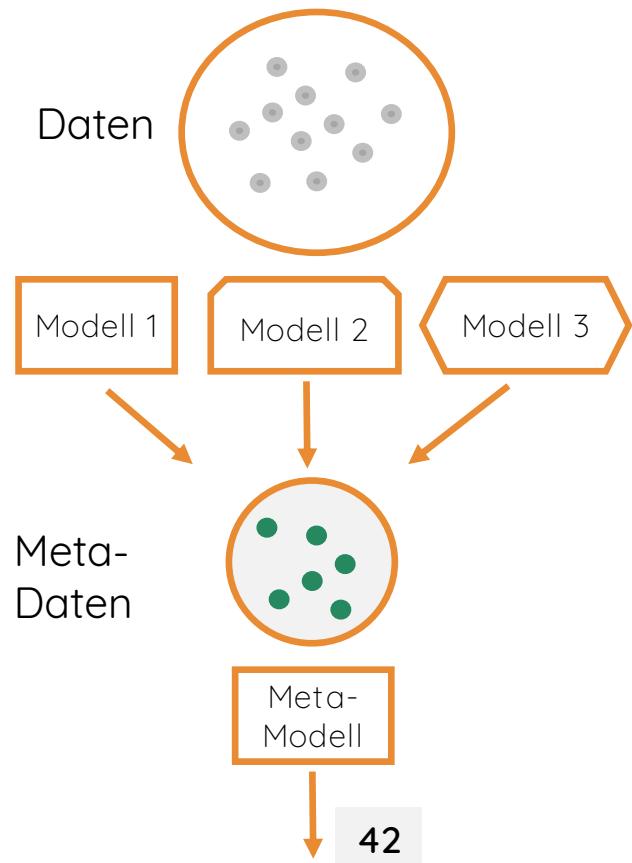
---



- Beim Voting sind Ensemble-Mitglieder (base learner) oft eine vielfältige Sammlung von Modelltypen, wie z. B. Entscheidungsbaum, Regression oder Neuronales Netz.
- Die base learner werden individuell trainiert.
- Ensemble-Vorhersagen werden erstellt, indem die Vorhersagen der einzelnen Modelle (base learner) aggregiert werden.
  - Klassifizierung: z.B. häufigste / gewichtete Werte
  - Regression: z.B. Mittelwert, Median

# Stacking - Metalearning

---



- Beim Metalearning sind die Ensemble-Mitglieder (base learner) ebenfalls eine vielfältige Sammlung von Modelltypen.
- Die base learner werden individuell trainiert.
- Die base learner generieren Metadaten (erste Vorhersagen), die an ein nachgelagertes Modell, dem meta learner, übergeben werden.
- Der meta learner erstellt auf Basis dieser Metadaten eine „eigene“ Vorhersage.

# Neural Networks

# Neural Network

---

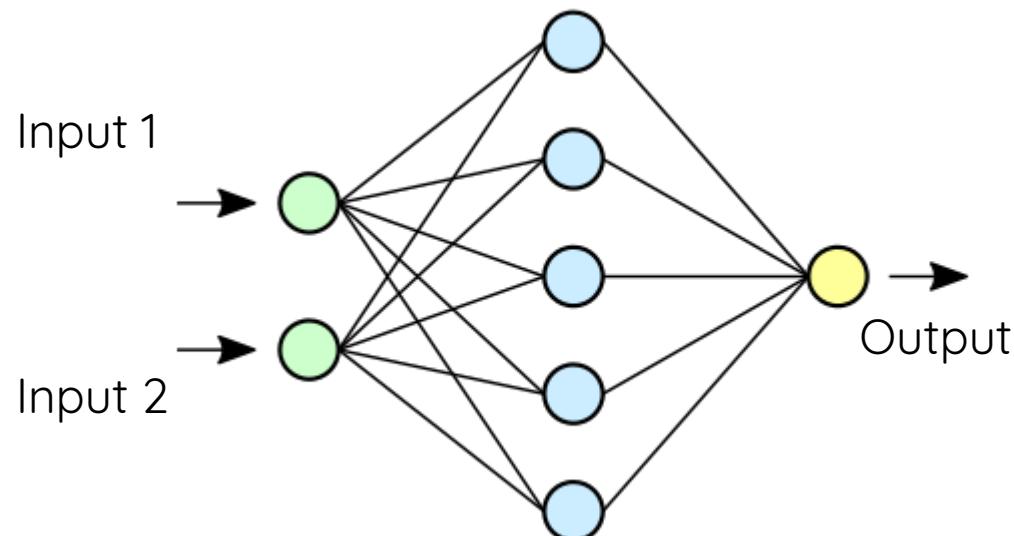


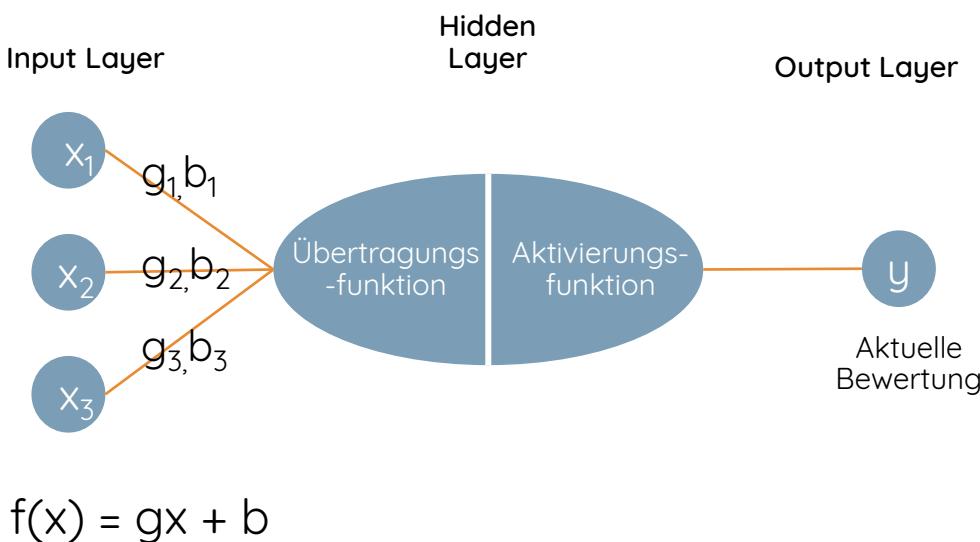
Bild: [Künstliches neuronales Netz – Wikipedia](#)

- Künstliche neuronale Netze, kurz: KNN (englisch artificial neural network, ANN), sind Netze aus künstlichen Neuronen.
- Nach dem biologischen Vorbild werden Modelle als eine Vernetzung von Neuronen im Nervensystem eines Lebewesens darstellen.
- Das Netzwerk besteht aus:
  - Neuronen bzw. Knoten &
  - Verbindungen zwischen den Neuronen

[KNIME Friendly Introduction](#) [KNIME Einführung ... Deep Learning](#)  
 [StatQuest Neural Networks Playlist](#)

# Neural Network: Neuron/Knoten

Beispiel  
Marke  
Modell  
Baujahr



Ein künstliches Neuron kann im Kern durch 3 Basiselemente beschrieben werden:

- **Gewichtung:** Jeder Eingang bekommt ein Gewicht. Die Gewichte bestimmen den Grad des Einflusses, den die Eingaben des Neurons in der Berechnung einnehmen.
- **Bias:** konstante, additive Komponente je Eingang.
- **Übertragungsfunktion:** Die Übertragungsfunktion berechnet anhand der gewichteten Eingaben die Netzeingabe des Neurons.
- **Aktivierungsfunktion:** Die Ausgabe des Neurons wird schließlich durch die Aktivierungsfunktion berechnet.

# Aktivierungs-funktionen

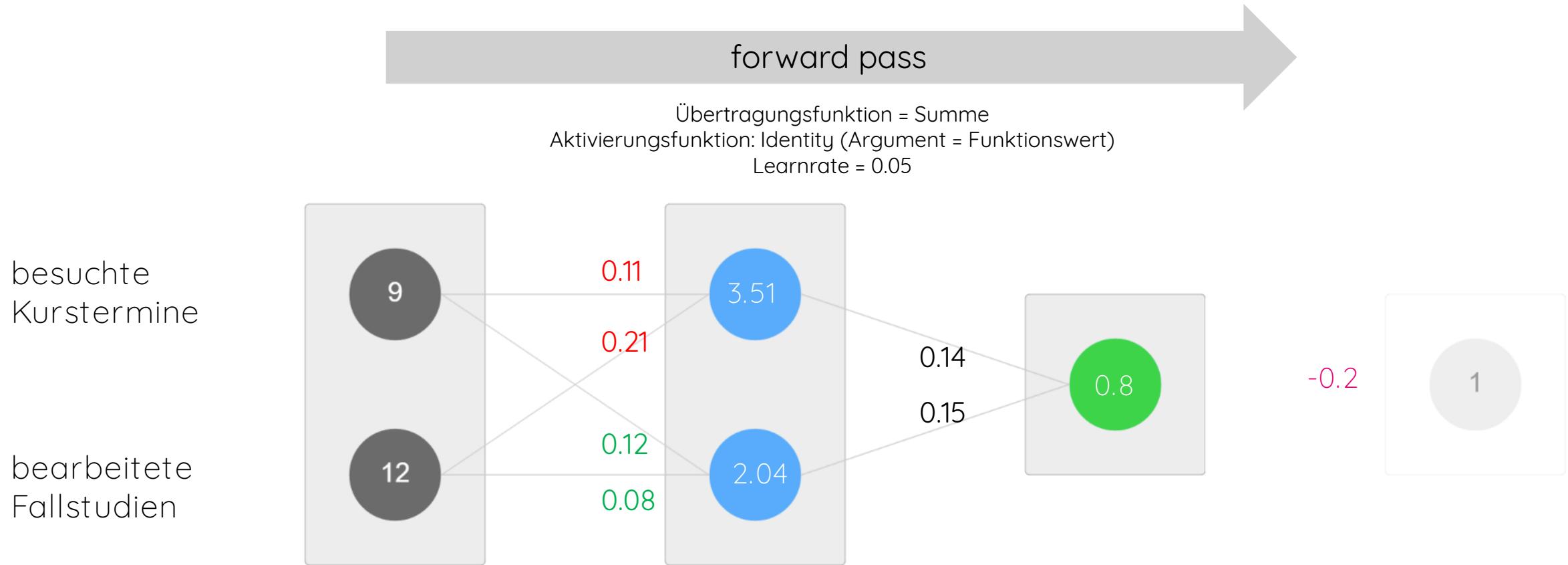
- Eine Aktivierungsfunktion definiert/bestimmt den möglichen Ausgabewert, also den Wertebereich der Ausgabe.
- Beispielsweise bildet eine lineare Funktion den Wert auf sich selbst ab, eine Sigmoid-Funktion bewirkt, dass die Ausgabe von Neuronen zwischen 0 und 1 liegen.

Bild: Activation Functions for Artificial Neural Networks  
[\(\[sebastianraschka.com\]\(http://sebastianraschka.com\)\)](http://sebastianraschka.com)

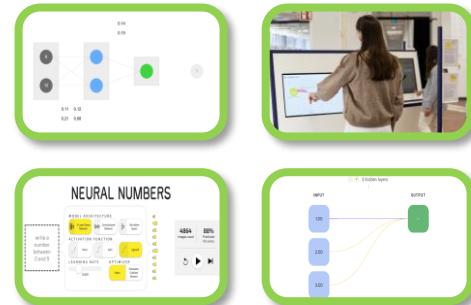
Activation function	Equation	Example	Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear (Identity)	$\phi(z) = z$	Adaline, linear regression $(-\infty, \infty)$	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic <u>(sigmoid)</u>	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN $(0, 1)$	
Hyperbolic tangent <u>tanh</u>	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks $(-1, 1)$	
Rectifier, <u>ReLU</u> (Rectified Linear Unit)	$\phi(z) = \max(0, z)$	Multi-layer Neural Networks $(0, \infty)$	
Rectifier, <u>softplus</u>	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

Copyright © Sebastian Raschka 2016  
[\(<http://sebastianraschka.com>\)](http://sebastianraschka.com)

# Neural Network: Beispiel



# Neural Network: Beispiel



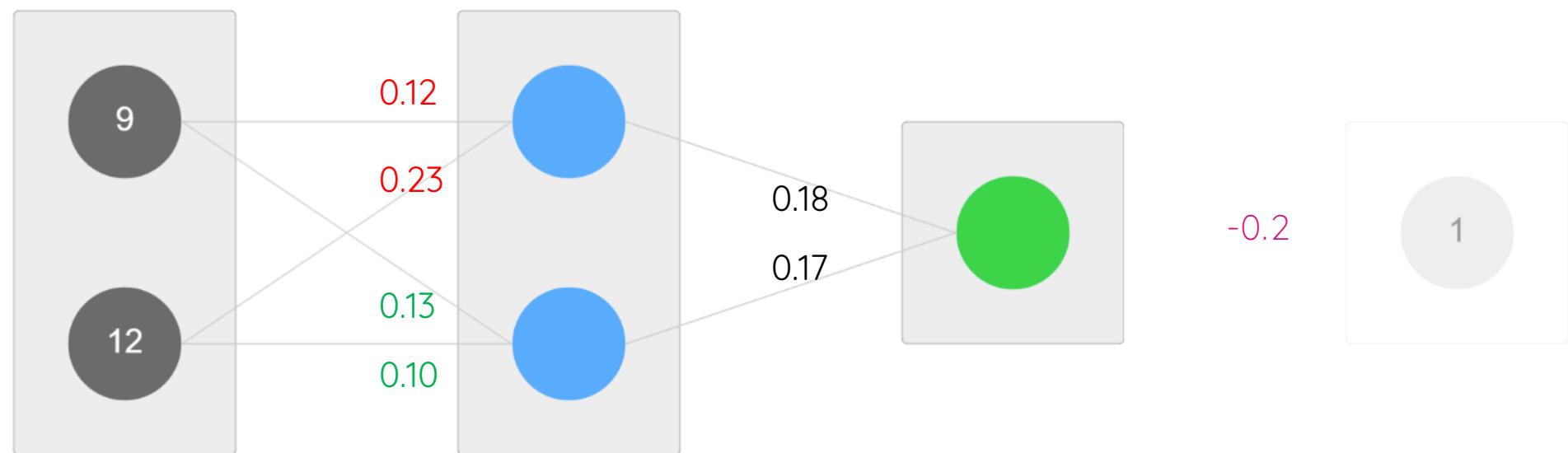
backward pass

$$W1 \text{ neu} = 0,12 = 0,11 - 0,05 * (9 * -0,04)$$

$$W5 \text{ neu} = 0,18 = 0,14 - 0,05 * (3,51 * -0,2)$$

besuchte  
Kurstermine

bearbeitete  
Fallstudien

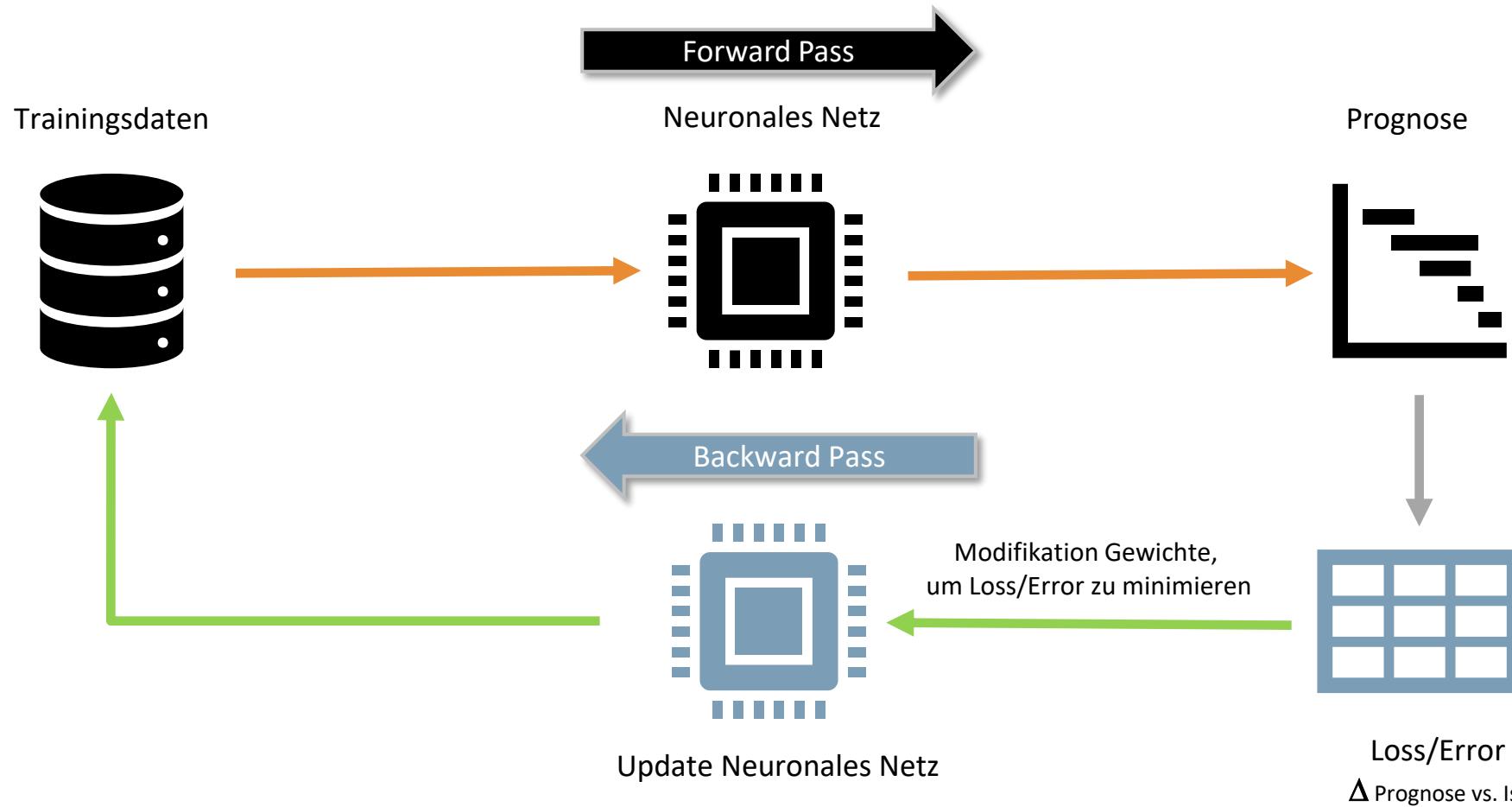


$$W4 \text{ neu} = 0,10 = 0,08 - 0,05 * (12 * -0,02)$$

$$W6 \text{ neu} = 0,17 = 0,15 - 0,05 * (2,04 * -0,2)$$

Quelle: [netflow-sample-nn](#)

# Training eines neuronalen Netzes



# Initialisierung der Gewichte

---

- Initiale Gewichte:
  - Zufällige Initialisierung, aber nicht völlig zufällig:
  - Xavier/Glorot: Berücksichtigt Input/Output-Dimensionen
  - He-Initialisierung: Optimiert für ReLU-Aktivierungen
- Vermeidung von:
  - Nullgewichten: Neuron wird inaktiv
  - Zu großen Werten: Sättigung der Aktivierungsfunktion
  - Symmetrischen Gewichten: Verhindert unterschiedliches Lernen
- Beispiel: Xavier/Glorot Initialisierung:
  - Gewichte werden aus einer Normalverteilung mit Varianz =  $2/(n_{in} + n_{out})$  gezogen
  - $n_{in}$ : Anzahl Eingabeneuronen,  $n_{out}$ : Anzahl Ausgabeneuronen
  - Verhindert, dass Signale zu stark oder zu schwach werden
  - Besonders geeignet für Sigmoid/Tanh-Aktivierungen

# Activation & Loss Function (1/2)

		Activation Functions		Loss Functions											
		Hidden Layers	Output Layer												
Problems		Sigmoid	Tanh	ReLU	Sigmoid	Tanh	Linear	ReLU	Softmax	Binary CE	Hinge CE	Categorical CE	MSE	MSLE	MAE
Classification	Binary classification (0 vs 1)	✓	✓	✓	✓					✓					
	Binary classification (-1 vs 1)	✓	✓	✓		✓					✓				
	Multi-class classification	✓	✓	✓					✓			✓			
Regression	Regression	✓	✓	✓	Δ	Δ	✓	Δ				✓			
	Regression (wide range)	✓	✓	✓			✓						✓		
	Regression (possible outliers)	✓	✓	✓			✓							✓	

# Activation & Loss Function (2/2)

Use Case	Beispiel	Hidden Layer Aktivierungsfunktion	Output Layer Aktivierungsfunktion	Lossfunktion
Clustering	k-means mit neuronalen Ansätzen	ReLU, Leaky ReLU	Softmax	Kullback-Leibler-Divergenz (KL-Divergence), Cosine Similarity Loss
Anomalieerkennung	Autoencoder zur Erkennung von Defekten in Bildern	ReLU, Tanh, Sigmoid	Sigmoid (bei normalisierten Daten)	Mean Squared Error (MSE), Binary Cross-Entropy
Dimensionality Reduction	Autoencoder für die Reduktion auf 2D oder 3D	ReLU, Leaky ReLU, Tanh	Linear, Sigmoid (bei normalisierten Daten)	Mean Squared Error (MSE)
Generative Modellierung	Generative Adversarial Networks (GANs) zur Bildgenerierung	Leaky ReLU, SELU	Tanh (Generator), Sigmoid (Discriminator)	Binary Cross-Entropy, Minimax Loss
Feature Learning	Contrastive Learning (z. B. SimCLR) zur Erstellung von embeddings	ReLU, Leaky ReLU	Normalisiert (z. B. mit L2-Norm)	Contrastive Loss (z. B. NT-Xent Loss), Triplet Loss
Dichteschätzung	Variational Autoencoder (VAE) zur Modellierung von Datenverteilungen	ReLU, Tanh, Leaky ReLU	Linear (Rekonstruktion), Sigmoid (bei normalisierten Daten)	Negative Log-Likelihood, Kullback-Leibler-Divergenz
Outlier Detection	Rekonstruktion mit Autoencodern zur Identifikation von Ausreißern	ReLU, Leaky ReLU	Sigmoid	Reconstruction Error (MSE), Cosine Similarity
Zeitreihenanomalien	RNN-Autoencoder zur Anomalieerkennung in Zeitreihen	Tanh, ReLU	Sigmoid, Linear	Mean Squared Error (MSE), Binary Cross-Entropy

# Auto ML

# AutoML

---



>>> pip install pycaret|

PyCaret is a simple, easy to learn, low-code machine learning library in Python. With PyCaret, you spend less time coding and more time on analysis.



Exploratory Data Analysis



Data Preprocessing



Model Training



Model Explainability

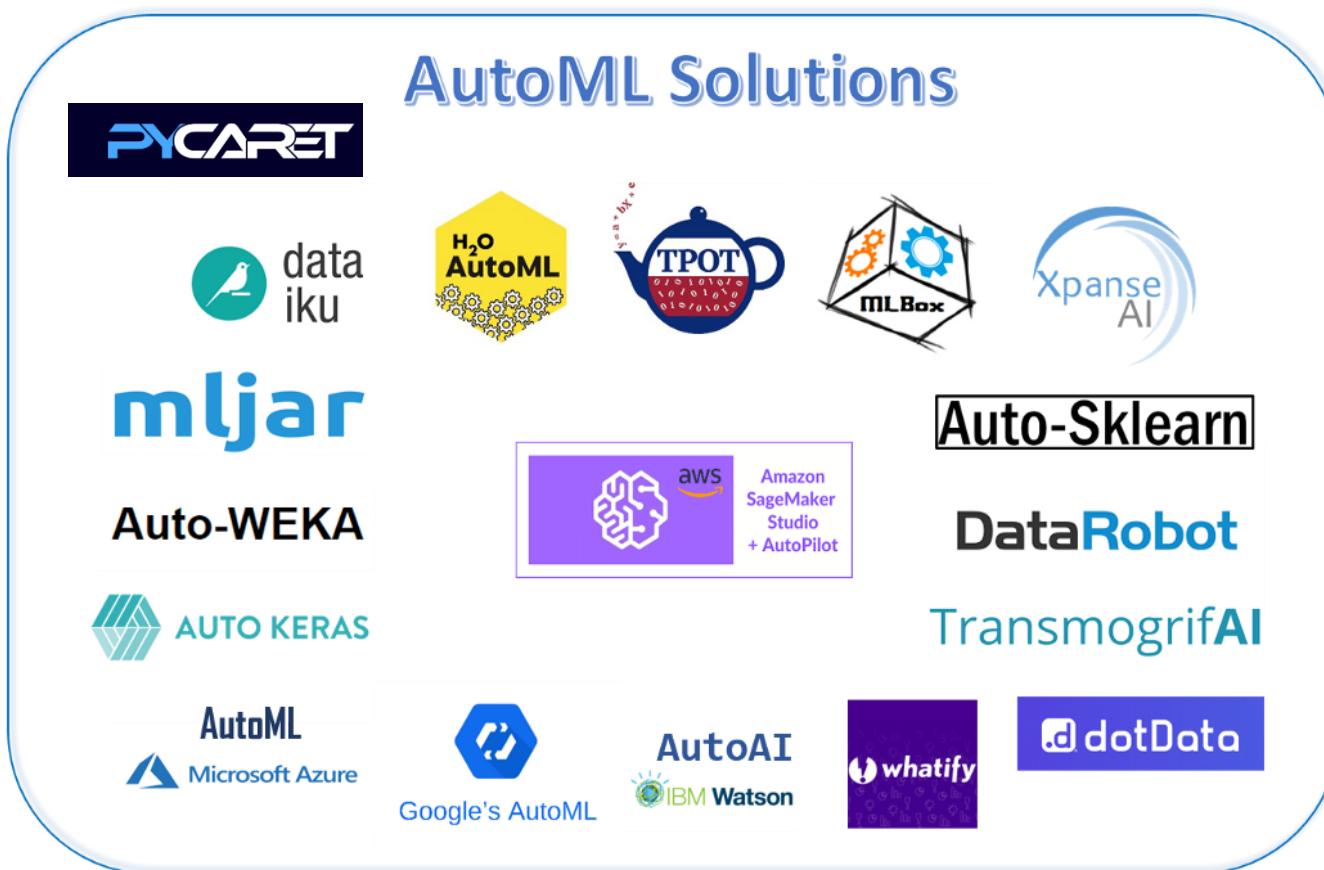


MLOps

Quelle: [Home - PyCaret](#)

- AutoML, kurz für "Automated Machine Learning", ist ein Bereich der künstlichen Intelligenz, der darauf abzielt, den Prozess des Anwendens von maschinellem Lernen auf reale Probleme zu automatisieren.
- Kernfunktionen:
  - Automatische **Datenvorbereitung**: AutoML-Tools können automatisch fehlende Daten behandeln, Kategorien kodieren und die besten Methoden zur Datentransformation und -normalisierung auswählen.
  - **Feature Engineering**: AutoML kann automatisch wichtige Merkmale (Features) aus einem Datensatz identifizieren und erstellen, was normalerweise ein zeitaufwändiger und komplexer Prozess ist.
  - Auswahl von **Algorithmen und Modellen**: AutoML-Systeme können aus einer Vielzahl von maschinellen Lernalgorithmen auswählen und diejenigen identifizieren, die am besten zu den spezifischen Daten und dem Problem passen.
  - **Hyperparameter-Tuning**: Sie optimieren automatisch die Einstellungen der Algorithmen (Hyperparameter), um die beste Leistung zu erzielen.
  - **Kreuzvalidierung und Modellbewertung**: AutoML-Tools führen eine gründliche Validierung durch, um die Effektivität verschiedener Modelle zu bewerten und Overfitting zu vermeiden.

# AutoML



Quelle: [AutoPilot: The Amazon Web Services \(AWS\) AutoML solution | by Saul Ventura | AutoPilot: The Amazon Web Services \(AWS\) AutoML solution | Medium](#)

# Reinforcement Learning

# Reinforcement Learning

---

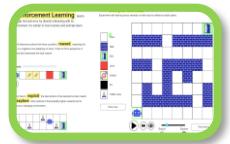
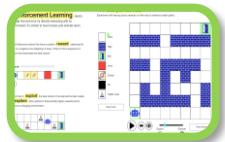


Bild mit DALL·E erstellt

Bausteine:

- **Agent**: Das ist der Lernende oder das Programm, das die Entscheidungen trifft.
- **Umgebung**: Das ist die Welt oder das System, in dem der Agent handelt. Die Umgebung reagiert auf die Aktionen des Agenten.
- **Aktionen**: Das sind die verschiedenen Entscheidungen oder Schritte, die der Agent in der Umgebung unternehmen kann.
- **Zustände**: Das sind die Situationen oder Positionen, in denen sich der Agent nach jeder Aktion befindet.



# Umgebung - Ziel - Aktionen

---



Bild mit DALL·E erstellt

- **Belohnungen:** Jedes Mal, wenn der Agent eine Aktion ausführt, erhält er eine Belohnung oder Strafe (auch Feedback genannt) von der Umgebung. Diese Belohnungen helfen dem Agenten zu verstehen, ob die Entscheidung gut oder schlecht war.
- **Ziel:** Das Ziel des Agenten ist es, eine Strategie zu entwickeln, die ihm hilft, die höchstmögliche Gesamtsumme an Belohnungen zu erhalten. Er lernt, welche Aktionen in welchen Zuständen die besten Belohnungen bringen.
- **Erkunden & Ausschöpfen:** Gleichgewicht finden zwischen der Suche nach neuen Informationen und der Nutzung des bereits vorhandenen Wissens.

# Special Neural Networks

---



Nur wo du zu Fuß  
warst, bist du  
auch wirklich  
gewesen.

---

Johann Wolfgang von Goethe



Bild von [aatlas](#) auf [Pixabay](#)

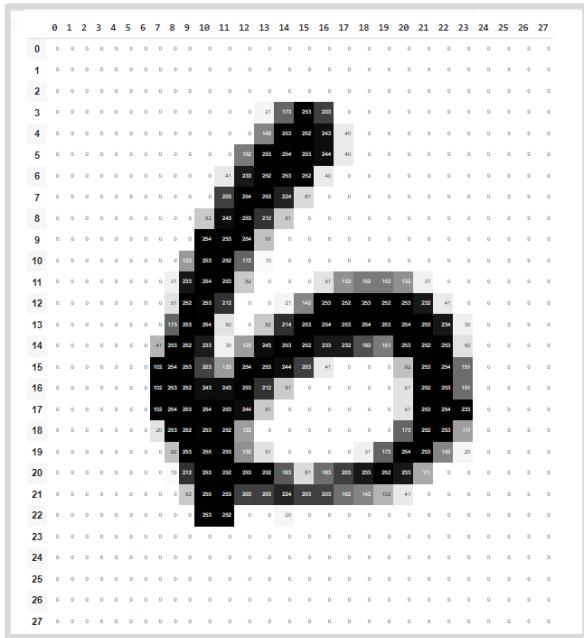
# Computer Vision

---



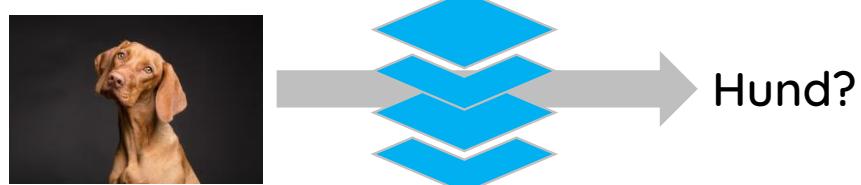
# Computer Vision

---



- Computer Vision analysiert Bilder/Videos, um deren Inhalt zu verstehen oder geometrische Informationen zu extrahieren.
- Typische Aufgaben sind die Objekterkennung.
- Technisch gesehen interpretiert das CV-Modell Bilder als eine Reihe von Pixeln.
- Mit Pixel oder Bildpunkt werden die einzelnen Farbwerte einer digitalen Rastergrafik bezeichnet sowie die zur Erfassung oder Darstellung eines Farbwerts nötigen Flächenelemente im Raster.
- Beispiel: Der MNIST-Datensatz, ein monochrones Bild mit einer Größe von  $28 \times 28$  Pixel.

# Convolutional Neural Network



- Convolutional Neural Network (CNN) sind mehrschichtige neuronale Netze, die wirklich gut darin sind, die Merkmale aus Daten herauszuholen. Sie funktionieren gut mit Bildern und benötigen nicht viel Vorverarbeitung.
- Mithilfe von Faltungen (convolutions) & Pooling (pooling) zum Reduzieren eines Bildes auf seine grundlegenden Merkmale können sie Bilder identifizieren.
- Diese Art von Architektur ist dominant, um Objekte aus einem Bild oder Video zu erkennen.
- Es wird in Anwendungen wie Bild- oder Videoerkennung, neuronaler Sprachverarbeitung usw. verwendet.

Bild von Péter Göblyös auf [Pixabay](#)

🎥 [StatQuest CNN](#) 🎥 [KNIME Break into Deep Learning for Image Data without Code, What are Convolutional Neural Networks?](#)

# CNN - Architektur

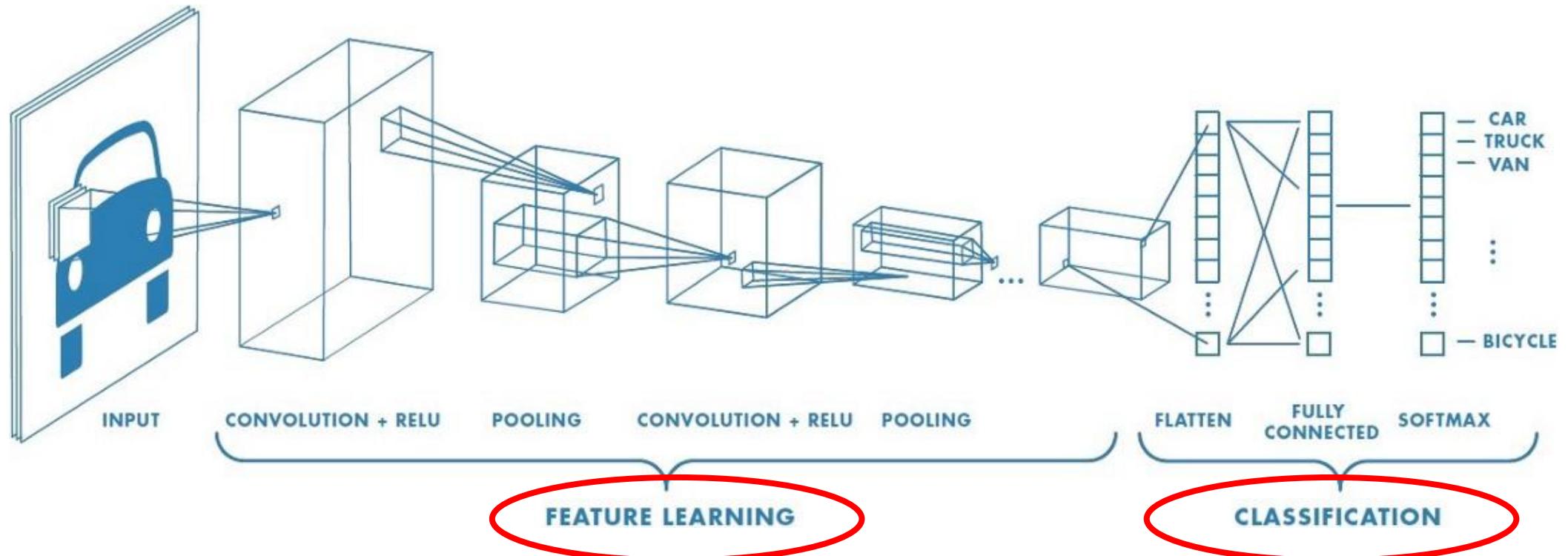
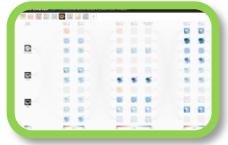
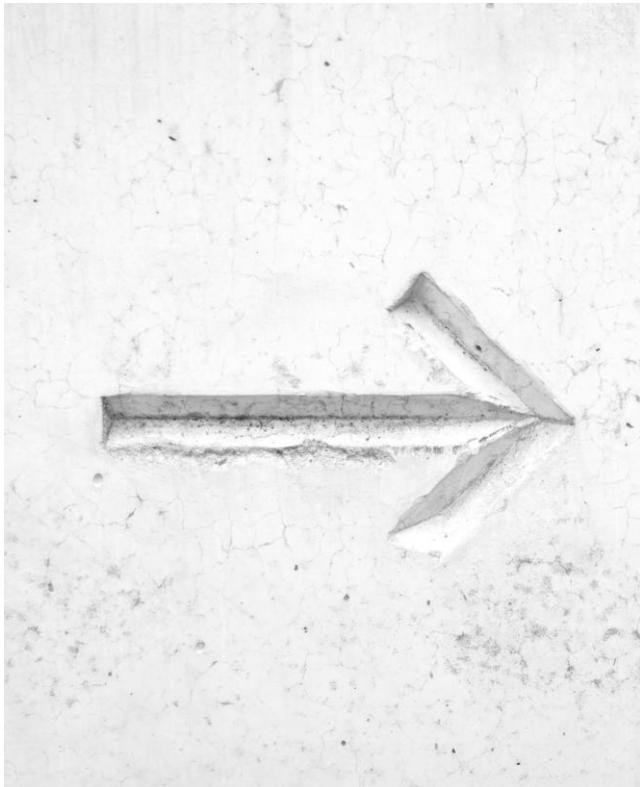


Bild-Quelle: [A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way | by Sumit Saha | Towards Data Science](#)

# CNN - Layer



- **Input Layer:**

Eingabeschicht, die Merkmale der Bilder, die im Argument der Funktion definiert sind (Höhe, Breite, Farbe).

- **Convolutional Layer:**

Es werden n Filter auf die Bild-Daten angewandt.

- **Pooling Layer:**

Der nächste Schritt nach der Faltung ist das Downampling der Bild-Daten. Der Zweck besteht darin, die Dimensionalität zu reduzieren. Bsp.: MaxPooling – Maximalwerte werden behalten.

- **Dense/Output Layer:**

In einem Dense-Layer werden die Eingangsdaten durch eine Multiplikation mit den Gewichtungen der Neuronen transformiert und dann durch eine Aktivierungsfunktion geleitet, um die Ausgabe des Neurons zu berechnen. Die Gewichte und Schwellenwerte der Neuronen werden im Laufe des gelernt.

Foto von [Hello I'm Nik](#) auf [Unsplash](#)

# Convolutional Layer – Kernel/Filter



Beispiel: 3 x 3 Filter  
auf ein 5 x 5 Pixelmatrix.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

Filter

1	0	1
0	1	0
1	0	1

4		

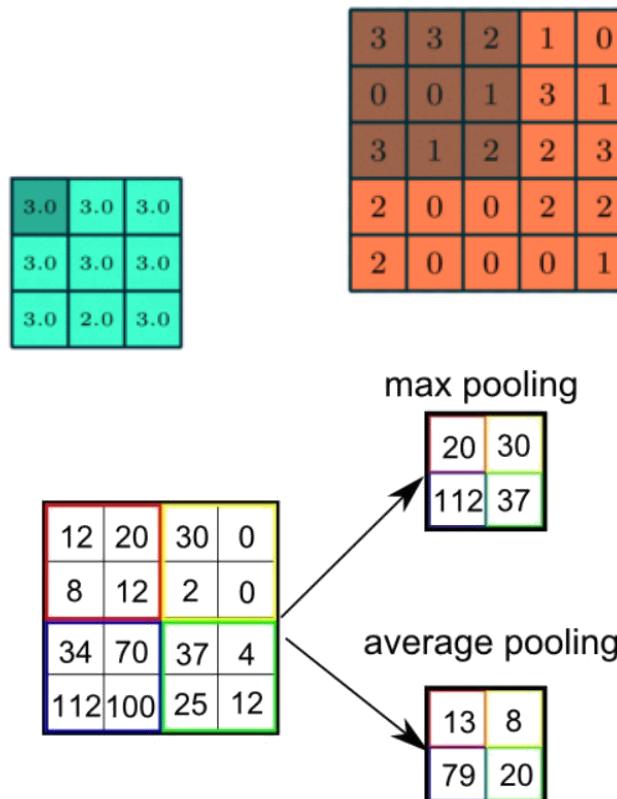
Convolved  
Feature

- Die Faltung (convolution) wird eingesetzt, um **Bildmerkmale zu erkennen**.
- Bildmerkmale sind z.B. vertikale/horizontale Linien, Kreise, Dreiecke, Kanten, Augen, Nase, ...
- Ein Filter ist eine Verkettung von x Kernel, wobei jeder Kernel einem bestimmten Eingangskanal zugeordnet ist.
- Die 2D-Faltung ist im Kern eine ziemlich einfache Operation: Der Kernel ist eine kleine Matrix von Gewichten.
- Dieser Kern „gleitet“ über die 2D-Eingabedaten, führt eine elementweise Multiplikation mit dem Teil der Eingabe durch, auf dem er sich gerade befindet, und summiert dann die Ergebnisse in einem einzigen Ausgabepixel.

Bild-Quelle: [A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way | by Sumit Saha | Towards Data Science](#)



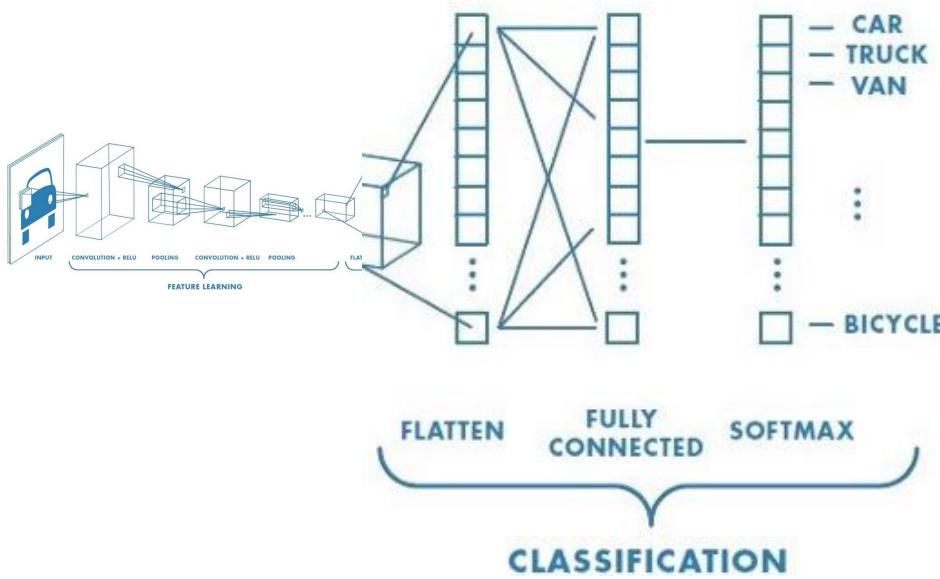
# Pooling Layer



- Ähnlich wie der Convolutional Layer ist der Pooling Layer dafür verantwortlich, die räumliche **Größe des Features** zu reduzieren.
- Darüber hinaus ist es nützlich, dominante Merkmale zu extrahieren, wodurch der Prozess des effektiven Trainierens des Modells aufrechterhalten wird.
- Es gibt zwei Arten von Pooling: Max Pooling und Average Pooling.
- Max Pooling gibt den maximalen Wert zurück.
- Average Pooling gibt den  $\varnothing$  aller Werte zurück.
- Max Pooling wirkt auch als Noise Suppressant (Entrauschung).
- Max Pooling wird daher bevorzugt eingesetzt.

Bild-Quelle: [A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way | by Sumit Saha | Towards Data Science](#)

# Flatten & FC Layer



- Nachdem der Prozess des Feature Learning durchlaufen wurde, wird die daraus resultierende Ausgabe geglättet.
- Ein Flatten Layer wird verwendet, um die Eingabe zu reduzieren. Wenn beispielsweise das Reduzieren auf eine Ebene mit der Matrix-Eingabeform ( $2 \times 2$ ) angewendet wird, lautet die Ausgabeform der Vektor-Ebene (4).
- Diese Ausgabe kann zu Klassifizierungszwecken an ein ‚normales‘ neuronales Netzwerk übergeben werden.
- In den einzelnen Neuronen wird durch die Übertragungs- und Aktivierungsfunktion eine weitere Verarbeitung durchgeführt.

# Sequenzmodellierung

---



# Natural Language Processing (NLP)

---

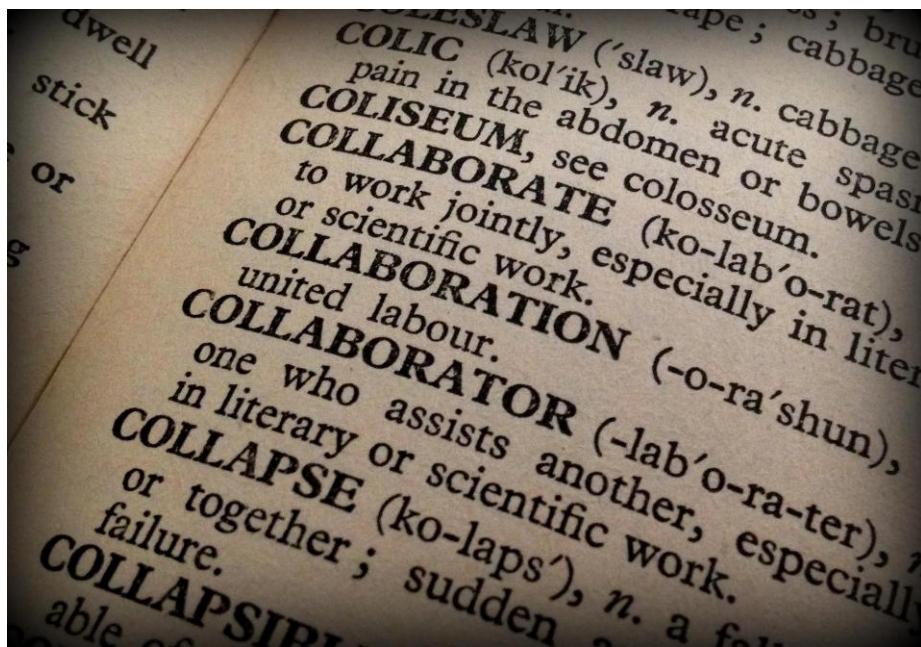


Bild von [Dianne Hope](#) auf [Pixabay](#)

- Natural Language Processing (NLP) ist der Bereich, der sich mit der Verarbeitung und dem Verständnis menschlicher Sprache befasst
- Hierzu gehört u.a.
  - Spracherkennung (Speech-to-Text),
  - Tagging , Bestimmung von Wortarten
  - Begriffsklärung, die Bedeutung eines Wortes
  - Named Entity Recognition (NEM), erkennt Wörter als nützliche Entitäten
  - Stimmungsanalyse (Sentiment Analysis), erkennt subjektive Eigenschaften (Einstellungen, Emotionen, Sarkasmus, ...)
  - Generierung natürlicher Sprache (Text-to-Speech)

Quelle: [What is Natural Language Processing? | IBM](#)

# Natural Language Processing (NLP)

---

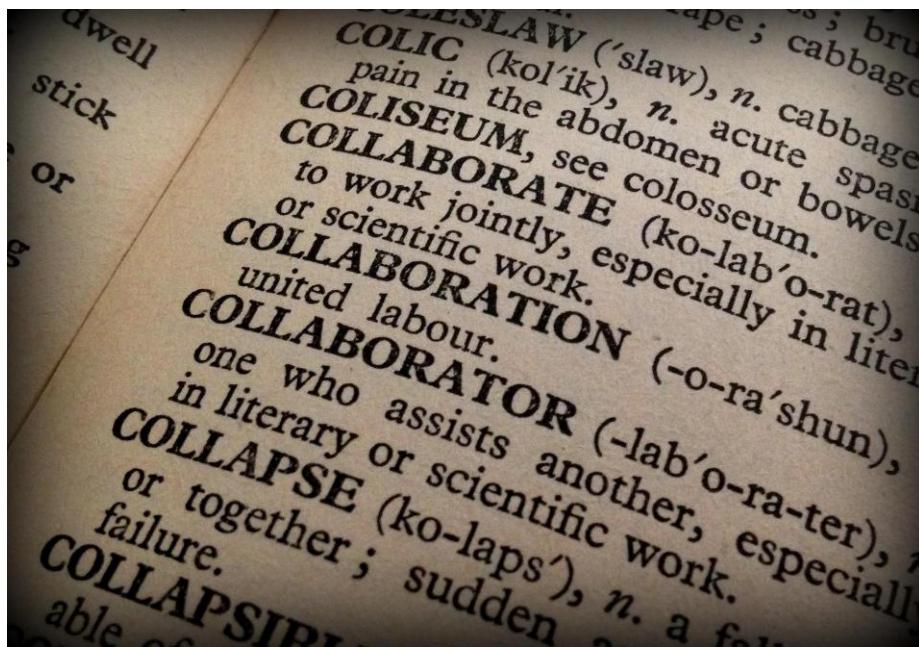


Bild von [Dianne Hope](#) auf [Pixabay](#)

 [KNIME Text Processing](#), [Text Mining](#), [From Words to Wisdom](#)

- Das Preprocessing für NLP unterscheidet sich ein wenig von der normalen Datenbereinigung.
- Die Textnormalisierung wird viel stärker betont als das Entfernen von Ausreißern oder Heelpunkten
  - Normalisierung Schrift (groß/klein),
  - Entfernen von Satzzeichen, Leerzeichen, Wörtern, ...
  - reduzieren Wörter auf ihre Basisform oder Stammform (z.B. Einzahl/Mehrzahl, grammatischer Fall).
  - Vektorisieren des Textes.
- Fortschrittliche NLP-Modelle (LLMs) können ggf. auf eine Textnormalisierung verzichtet.

Deep Dive  
siehe LLM

# Times Series Analysis

---



Foto von [Jake Hills](#) auf [Unsplash](#)

- Zeitreihendaten (zeitgestempelte Daten) sind eine Folge von Datenpunkten, die in zeitlicher Reihenfolge indiziert sind.
- Diese Datenpunkte bestehen typischerweise aus aufeinanderfolgenden Messungen, die über ein Zeitintervall durchgeführt wurden, und werden verwendet, um Änderungen im Laufe der Zeit zu verfolgen.
- Mithilfe von Zeitreihendaten können Muster in den Daten aufgedeckt werden, die dann zur Vorhersage/Prognose verwendet werden können.
- Dies hat seine Grenzen, wenn die gegebene Variable weitgehend von externen Faktoren abhängt, wird ein Modell nicht die besten Ergebnisse liefern.

# Eigenschaften einer Zeitreihe

---

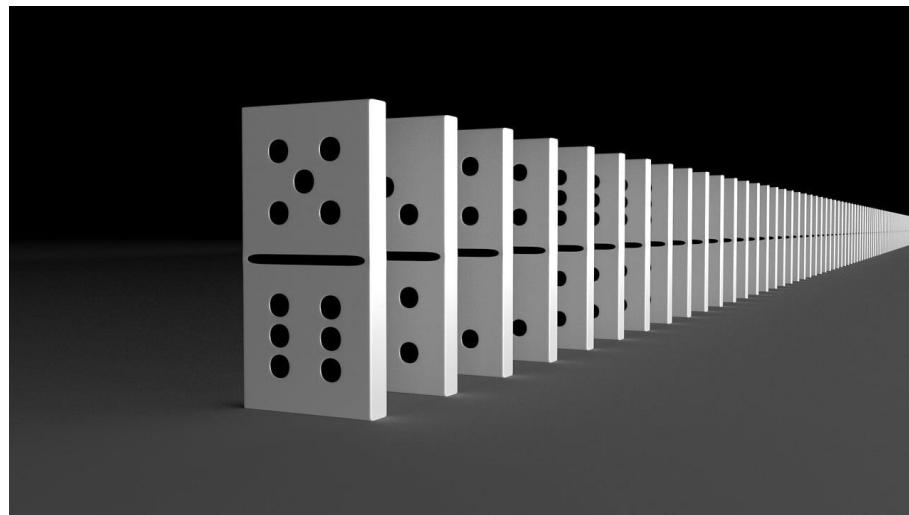


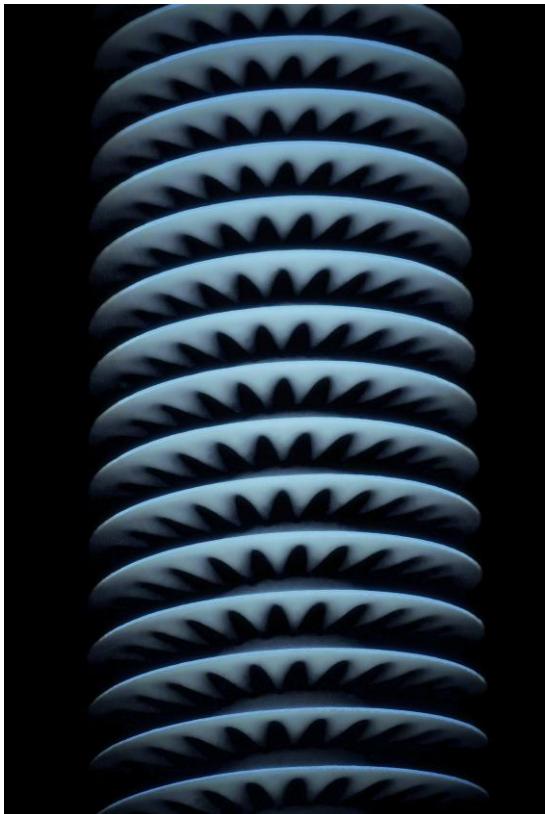
Bild von [Erik Stein](#) auf [Pixabay](#)

Wichtige Eigenschaften:

- **Trend:** Ein Trend in einer Zeitreihe bedeutet, dass die Werte im Laufe der Zeit systematisch steigen oder fallen. Ein positiver Trend deutet auf ein Wachstum oder eine zunehmende Tendenz hin, während ein negativer Trend auf eine Abnahme oder eine abnehmende Tendenz hinweist.
- **Saisonale Effekte:** Saisonale Effekte treten regelmäßig in bestimmten Zeitintervallen auf und können auf saisonale Muster, wie beispielsweise saisonale Veränderungen im Verbrauch oder Produktion, zurückzuführen sein.
- **Zyklische Effekte:** Zyklische Effekte beziehen sich auf wiederkehrende Schwankungen in der Zeitreihe, die jedoch nicht durch saisonale Effekte verursacht werden. Sie können auf wirtschaftliche oder politische Zyklen oder andere langfristige Trends zurückzuführen sein.

# Recurrent Neural Network (RNN)

---



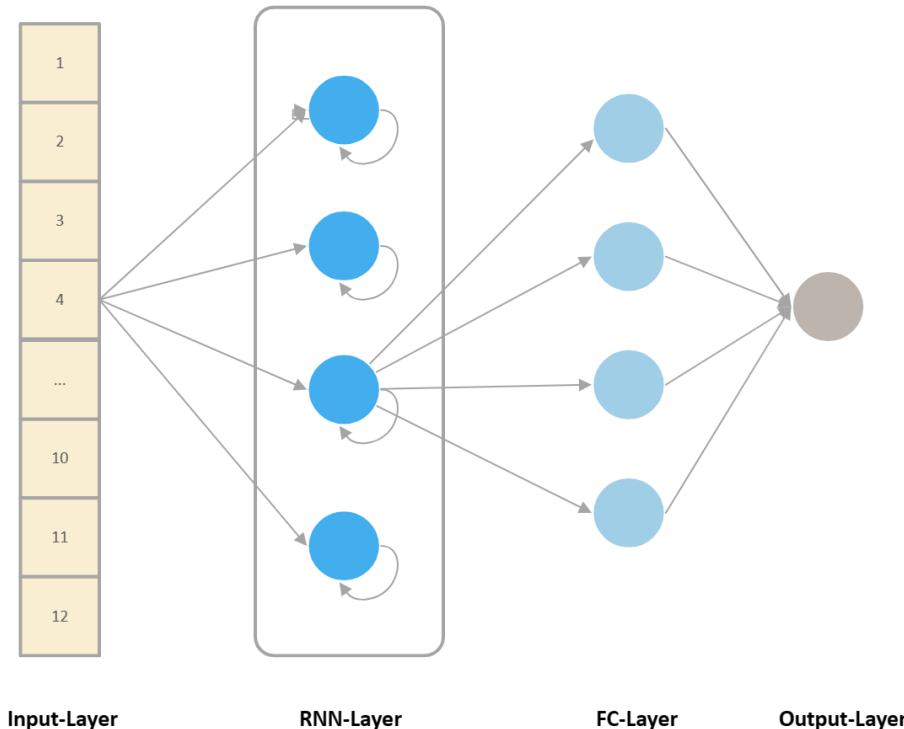
- Ein Recurrent Neural Network (RNN), eine Architektur von Deep-Learning-Modell, das für die Verarbeitung von sequenziellen Daten, wie z. B. Zeitreihen oder Texten, verwendet wird.
- RNN haben eine eingebaute Rückkopplungsschleifen, die es ermöglicht, dass die Informationen mehrmals an denselben Knoten zurückgesendet werden.
- Sie sind daher besonders geeignet Sequenzen zu verarbeiten.
- Recurrent Layer können auf verschiedene Arten implementiert werden, wie z.B. LSTM - Long-Short-Term-Memory.
- Die Wahl der Implementierung hängt von der Art des Problems und der verfügbaren Daten ab.

Foto von [Anne Nygård](#) auf [Unsplash](#)

 [StatQuest RNN](#)  [KNIME Codeless Deep Learning for Sequential Data](#)

# Recurrent Neural Network

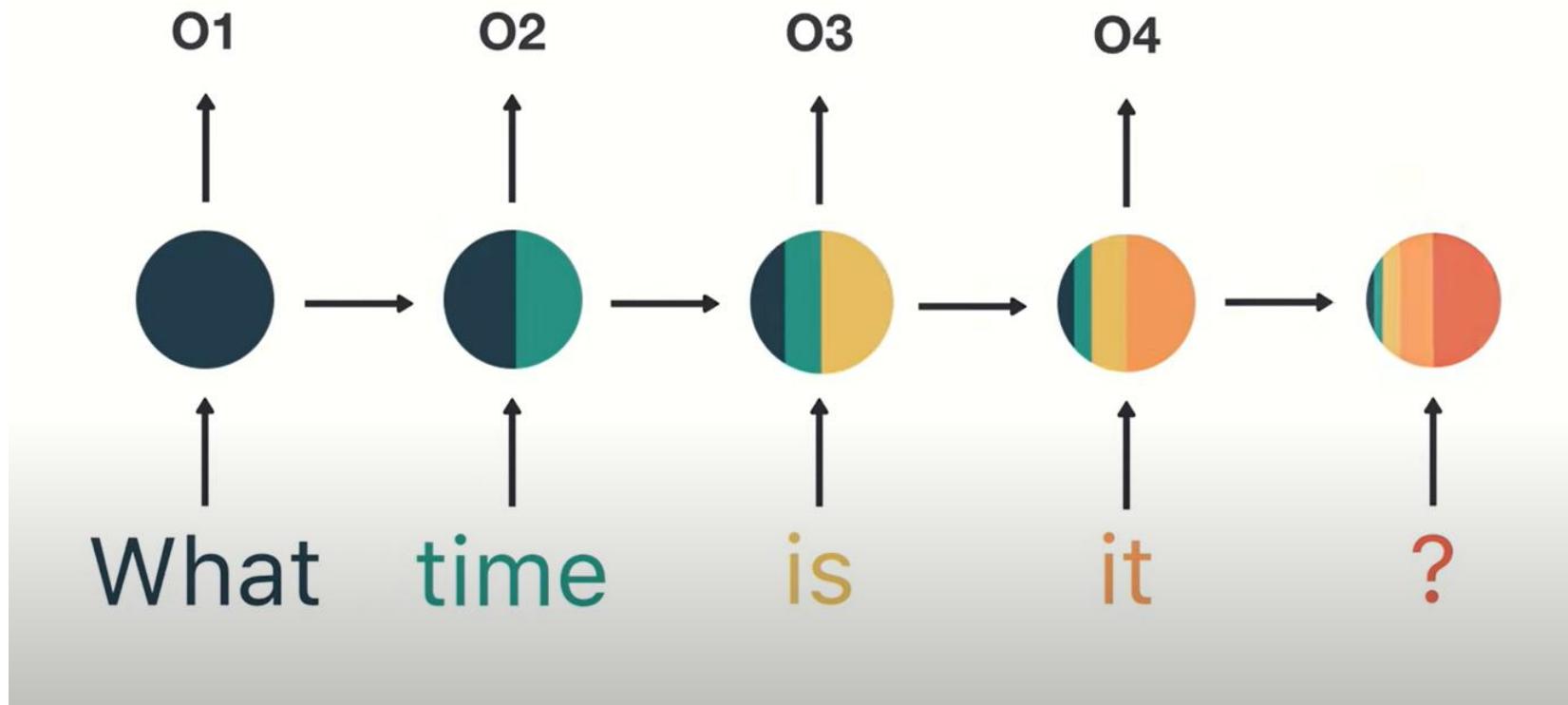
---



- Ein "Recurrent Neural Network" (RNN) ist wie ein normales neuronales Netzwerk, hat aber zusätzlich eine Art "Gedächtnis".
- Während normale neuronale Netzwerke eine Eingabe nehmen und eine Ausgabe produzieren, können RNNs ihre "Erinnerungen" von vorherigen Schritten nutzen, um aktuelle Informationen zu verarbeiten.
- Dies macht sie besonders nützlich für Aufgaben, bei denen die Reihenfolge der Daten wichtig ist, wie z.B. beim Lesen von Text oder beim Hören von Musik.
- Man kann sich ein RNN wie eine Maschine vorstellen, die nicht nur auf das sieht, was gerade vor ihr passiert, sondern auch das berücksichtigt, was sie zuvor gesehen hat.

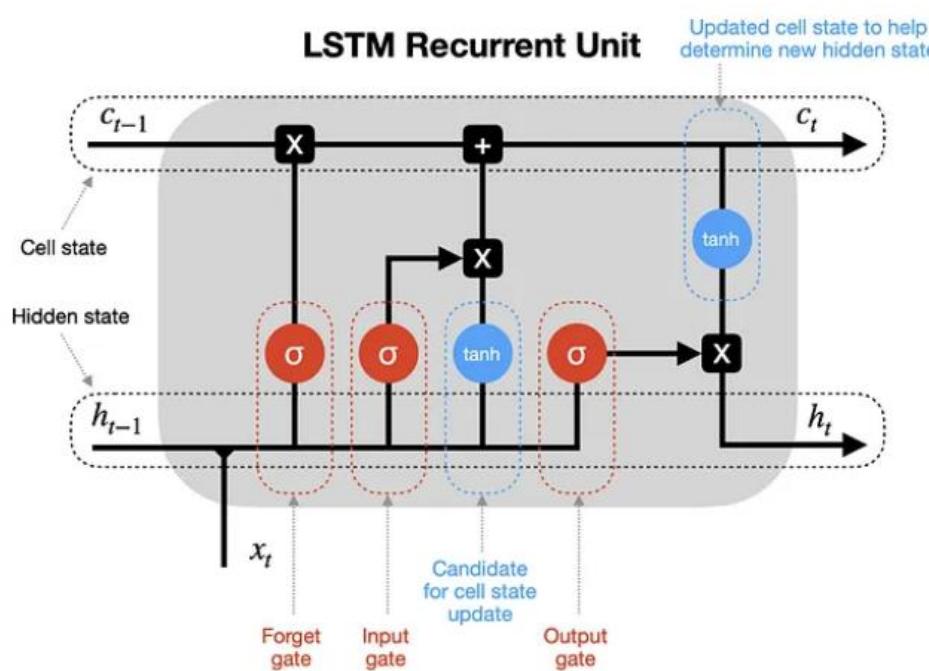
# Recurrent Neural Network

---



Quelle: [Illustrated Guide to Recurrent Neural Networks: Understanding the Intuition](#)

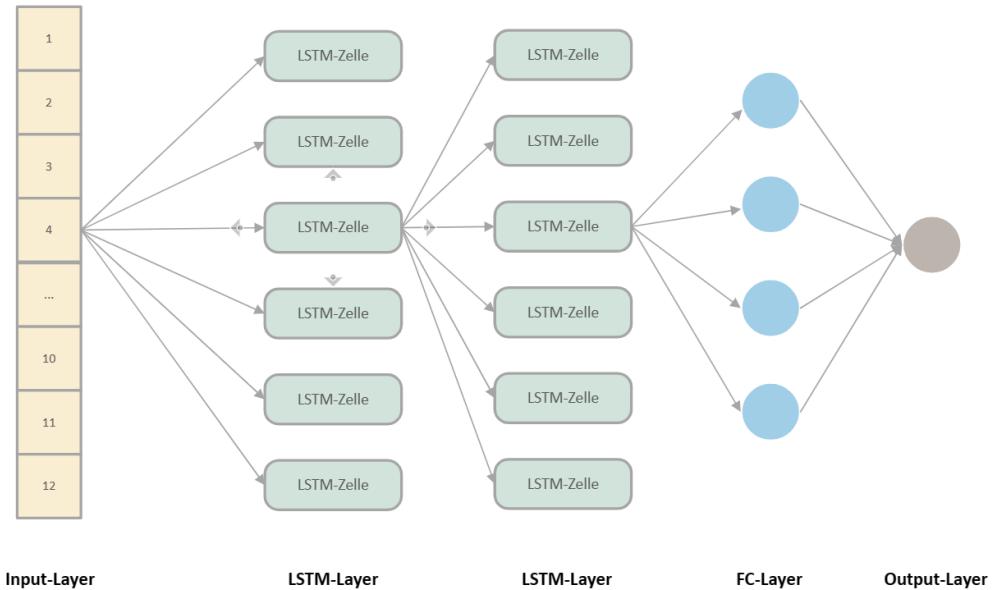
# Long-Short-Term-Memory-Zelle



- Forget Gate:
  - Bestimmt, welche Teile des aktuellen Zellzustands vergessen oder beibehalten werden sollten.
- Input Gate:
  - Entscheidet, welche Informationen aus dem aktuellen Eingabevektor und dem vorherigen Zustand in den Zellzustand aufgenommen werden sollten.
- Output Gate:
  - Entscheidet, welche Teile des aktualisierten Zellzustands als versteckter Zustand für den nächsten Zeitschritt verwendet werden sollten.
- Cell-State:
  - Ein vorläufiger Update-Vektor für den Zellzustand, der neue, potenzielle Werte für den Zellzustand auf Basis des aktuellen Eingabevektors und des vorherigen Zustands vorschlägt.

Bild: [LSTM Recurrent Neural Networks – How to Teach a Network to Remember the Past | by Saul Dobilas | Towards Data Science](#)

# Long-Short-Term-Memory



- RNN haben oft „Schwierigkeiten“ langfristige Abhängigkeiten zu erfassen.
- Long-Short-Term-Memory (LSTM) sind spezielle Einheiten, die als Gedächtnis-Zellen bezeichnet werden.
- Ein LSTM ist eine Sammlung von Neuronen mit einer speziellen Architektur, die dazu dient, sequenzielle Daten über längere Zeiträume hinweg effektiv zu verarbeiten.
- Ein LSTM-Block oder eine LSTM-Zelle enthält:
  - Eingabetor (Input Gate)
  - Vergesslichkeitstor (Forget Gate)
  - Ausgabetor (Output Gate)
  - Kandidaten-Zellzustand und den eigentlichen Zellzustand
- Jedes dieser Elemente enthält Neuronen, die Operationen ausführen, um Informationen zu speichern, vergessen oder weitergegeben zu können.

[StatQuest LSTM](#) [KNIME LSTM](#)

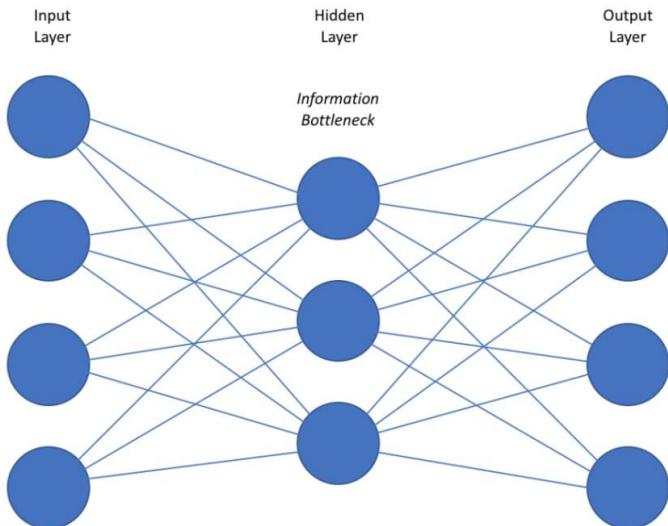
# AutoEncoder

---



# Autoencoder

---



[Building an Autoencoder for Generative Models -  
DEV Community](#)

- Ein Autoencoder ist ein spezieller Typ von neuronalem Netzwerk, das dazu verwendet wird, Daten auf ihre wichtigsten Merkmale abzubilden. Es besteht aus zwei Hauptteilen: dem Encoder und dem Decoder.
- Der **Encoder** nimmt die Eingabedaten und reduziert sie auf ihre wesentlichen Merkmale. Diese Darstellung kann als komprimierte Version der Eingabedaten betrachtet werden.
- Der **Decoder** nimmt die komprimierte Version aus dem Encoder und rekonstruiert damit eine nahezu identische Version der Eingabedaten.
- Autoencoder haben viele Anwendungen, z. B. bei der Bild- und Spracherkennung, der Dimensionsreduktion, die Anomalieerkennung und die Generierung von Daten.
- Ein Beispiel ist die Verwendung von Autoencodern zur Entfernung von Rauschen aus Bildern, indem sie das Rauschen aus der Eingabe entfernen und eine saubere Version der Bilder als Ausgabe rekonstruieren.

# Media Pipe Studio

---



# Media Pipe Studio

The screenshot shows the MediaPipe Studio interface. On the left, a sidebar menu lists categories: VISION, TEXT, and AUDIO. Under VISION, sub-options include Object Detection, Image Classification, Image Segmentation, Interactive Segmentation, Gesture Recognition, Hand Landmark Detection, Image Embedding, Face Stylistation, Face Detection, Face Landmark Detection, and Pose Landmark Detection. Under TEXT, options are Text Classification, Text Embedding, and Language Detection. Under AUDIO, there is Audio Classification. The main area features a title "On-device ML for everyone" and a subtitle "Enjoy a new way to explore and evaluate on-device ML solutions." Below this, there are six demo cards arranged in two rows of three:

- VISION**
  - Object Detection**: Shows a dog and a cat with bounding boxes labeled "Dog" and "Cat".  
Description: Track and label objects in images.  
[See demo](#)
  - Image Classification**: Shows a flamingo.  
Description: Identify content in images.  
[See demo](#)
  - Image Segmentation**: Shows a person's face and a pink silhouette of the same person.  
Description: Locate objects and create image masks with labels.  
[See demo](#)
- TEXT**
  - Interactive Segmentation**: Shows two chairs.  
Description: Segment the object of interest in an image.  
[See demo](#)
  - Gesture Recognition**: Shows a hand with a "Thumbs up 63%" overlay.  
Description: Identify and recognize hand gestures.  
[See demo](#)
  - Hand Landmark Detection**: Shows a hand holding an egg.  
Description: Detect hand landmarks.  
[See demo](#)

On the right side of the main area, there is a small diagram of a person interacting with a hexagonal shape, with a "Point..." button above it and a "Start" button below it.

# Workflow Design

---



# Anwendung von Modellen in sklearn

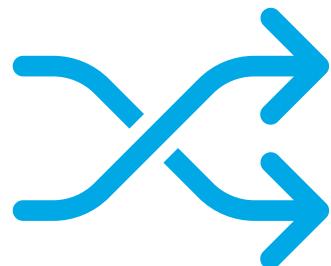
---

Scikit-Learn ermöglicht unterschiedlichen Ansätze zur Definition und Verwendung von Skalierungs- und Codierungsmodellen. Im Kern geht es darum, ob man eine Instanz eines Skalierers oder Codierers (z.B. MinMaxScaler) explizit erstellt und verwendet oder direkt auf den Daten anwendet.

Ansatz	Beispiel	Vorteile	Nachteile
Direkte Anwendung	<code>MinMaxScaler().fit_transform(data)</code>	<ul style="list-style-type: none"><li>▪ Kürzerer, direkter Code</li><li>▪ Geeignet für einmalige Anwendungen</li></ul>	<ul style="list-style-type: none"><li>▪ Keine Wiederverwendung des Modells</li><li>▪ Erschwert Reproduzierbarkeit</li><li>▪ keine inverse Transformation möglich</li></ul>
Explizite Definition	<code>scaler = MinMaxScaler() scaler.fit_transform(data)</code>	<ul style="list-style-type: none"><li>▪ Ermöglicht Wieder-verwendung und Konsistenz</li><li>▪ Gute Reproduzierbarkeit</li><li>▪ Ermöglicht inverse Transformation</li></ul>	<ul style="list-style-type: none"><li>▪ Mehr Codezeilen</li><li>▪ Leicht erhöhte Komplexität</li></ul>

# Daten direkt transformieren?

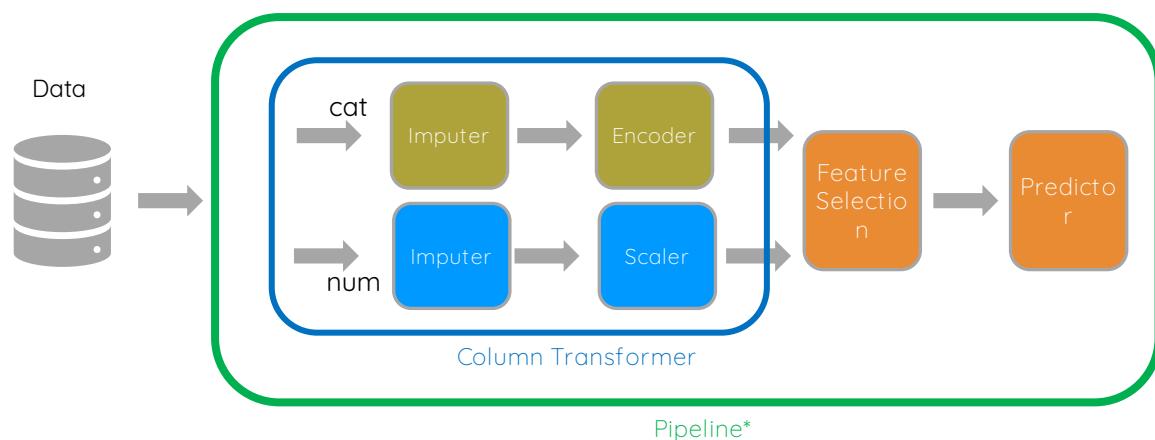
---



- **Datenerhaltung:** Wenn die originalen Daten für spätere Analysen oder Vergleiche benötigt werden, ist es besser, eine separate scaled\_data-Variable zu erstellen. So bleiben die originalen Daten unverändert, und man kann sowohl auf die ursprünglichen als auch auf die skalierten Daten zugreifen.
- **Speicherplatz:** Wenn der Datensatz sehr groß ist und Speicherplatz eine Rolle spielt, könnte die direkte Anwendung der Skalierung auf die Daten effizienter sein, da dies weniger Speicher verbraucht. In diesem Fall sollten Sie sich jedoch bewusst sein, dass die Originaldaten verloren gehen.
- **Klarheit und Wartbarkeit des Codes:** Die Verwendung einer separaten Variable für die skalierten Daten kann den Code klarer und leichter wartbar machen, da deutlich wird, welche Daten bearbeitet wurden und welche nicht.
- **Wiederverwendung und Reproduzierbarkeit:** Wenn Sie planen, denselben Skalierungsvorgang auf mehrere Datensätze (z.B. Trainings- und Testdaten) anzuwenden, ist es sinnvoll, die Skalierung als separaten Schritt zu implementieren. Dies stellt sicher, dass dieselbe Transformation auf alle relevanten Daten angewandt wird.

# Pipeline

---



\* Pipeline kann bei Bedarf durch benutzerdefinierte Transformer ergänzt werden

- Pipeline ist eine Reihe von Aufgaben, die nacheinander ausgeführt werden.
- Die Ausgabe einer Aufgabe ist die Eingabe der nächsten Aufgabe, bis sie am Ende das Endprodukt ausgibt.
- Eine Pipeline kann prinzipiell alle Aufgaben aus allen Phasen des Entwicklungsprozesses beinhalten.
- Der Vorteil von Pipelines ist, dass sie die Datenaufbereitung beschleunigen.
- Außerdem kann die Pipeline problemlos auch für das Test-Set verwendet werden (siehe auch Data Leakage).
- Die macht den Umgang mit noch nicht transformierten Testdaten oder mit neuen Daten noch einfacher.

# Pipeline & ColumnTransformer

---

	Funktionsweise	Einsatzbereich
Pipeline	Verwendet, um eine Reihe von Transformatoren zu einer <b>sequentiellen</b> und dann einen finalen Schätzer.	Erstellen einer maschinelles Lernen Pipeline, die Daten erst transformiert und dann prognostiziert.
ColumnTransformer	Verwendet, um verschiedene Transformatoren auf verschiedenen Teilmengen von Spalten <b>parallel</b> abzuwenden und anschließend die Ausgabe zu verketten.	Bei unterschiedlichen Daten unterschiedliche Transformationen anwenden und diese zusammen in einer Pipeline verwenden.

In der Praxis werden oft beide verwendet, wobei ColumnTransformer innerhalb einer Pipeline genutzt wird, um komplexe Datenverarbeitungs-Workflows zu erstellen.

# Eigenschaften von Pipelines

---

- In einer Scikit-Learn-Pipeline hat man keinen **direkten** Zugriff auf die Zwischenergebnisse der Transformationen, wie beispielsweise die Daten nach der Imputation von Missing Values.
- Eine Pipeline in Scikit-Learn ist so konzipiert, dass sie einen sequenziellen Prozess von Transformationen und Modelltraining **kapselt**, wobei der Fokus auf der Endvorhersage liegt.
- Um auf transformierte Daten in einer Pipeline zuzugreifen, muss man die Transformationen manuell **außerhalb** der Pipeline durchführen bzw. nachbauen.
- Man kann jedoch auf Teilschritte und deren Ergebnisse in einer Pipeline zugreifen.
- Jeder Schritt in einer Pipeline wird als ein Tupel aus einem Namen und einem Transformer oder Modell definiert.
- Beispielsweise kann man mit `pipeline.named_steps['schrittnname']` auf einen bestimmten Schritt zugreifen. Dies ermöglicht es, die Eigenschaften des Transformators oder Modells zu überprüfen, wie beispielsweise die gelernten Koeffizienten in einem linearen Modell.

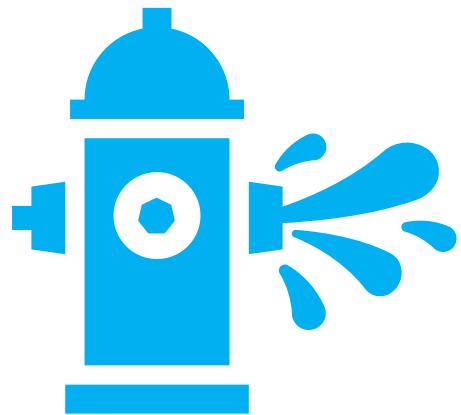
# Pipelines in Keras

---

- In Keras lassen sich **keine** Pipelines definieren.
- Es können jedoch verschiedene Vorverarbeitungsschritte direkt in das Modell integriert werden. Dazu gehören:
  - Normalisierung: Anpassung der Skalierung der Eingabedaten, z.B. durch die Normalization Schicht.
  - Codierung kategorialer Daten: Umwandlung von kategorialen Daten in numerische Formate, beispielsweise durch CategoryEncoding.
  - Text-Vorverarbeitung: Umwandlung von Textdaten in eine geeignete Form für das Modell, z.B. mit TextVectorization.
  - Bild-Vorverarbeitung: Anpassungen an Bildern, wie das Reskalieren oder Anwenden von Daten-Augmentierung.
- Diese Schritte ermöglichen es, die Datenvorverarbeitung als Teil des Modells zu definieren, was den Workflow vereinfacht und die Portabilität des Modells verbessert.

# Data Leakage

---



- Vorhersagemodellierung sollen genaue Vorhersagen auf neuen Daten treffen, die während des Trainings nicht sichtbar sind.
- Daher wird bei der Modellerstellung eine Trennung zwischen Trainings- und Testdaten vorgenommen.
- Datenlecks können auftreten, wenn Informationen von außerhalb des Trainingsdatensatzes verwendet werden, um das Modell zu erstellen.
- Diese zusätzlichen Informationen können zu übermäßig optimistischen, wenn nicht sogar völlig ungültige Vorhersagemodelle.
- Neben einer Trennung von Trainings- und Testdaten sollten auch Codierung/Skalierung getrennt erfolgen.

# XAI - eXplainable Artificial Intelligence

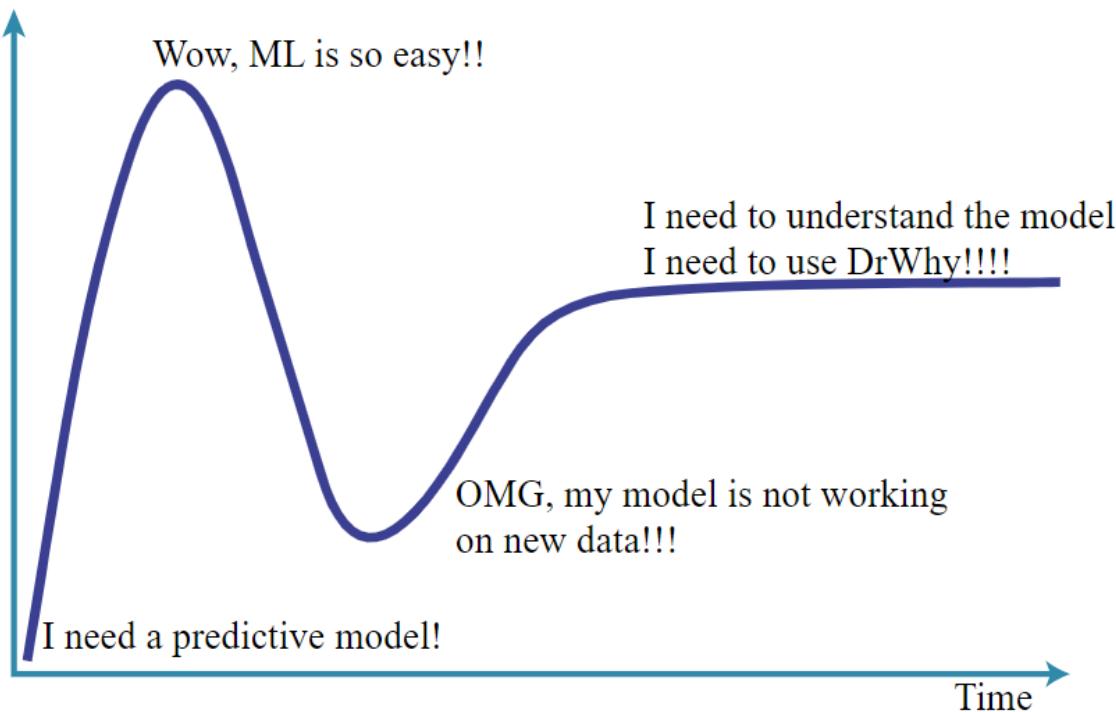
---



# Why XAI?

---

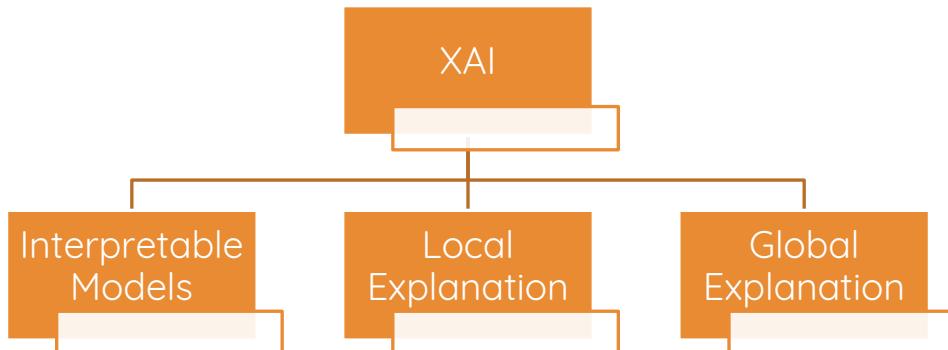
## Hype Cycle for Predictive Models



Quelle: [DrWhy/README.md at master · ModelOriented/DrWhy \(github.com\)](#)

# XAI - Intro

---



- Explainable AI oder XAI ist ein Ansatz, der darauf abzielt, dass die Funktionsweise und Entscheidungen von ML-Modellen für Menschen verständlich und nachvollziehbar ist.
- Ansätze für XAI:
  - Interpretable Models: Verwendung von ML-Modellen, die von Grund auf so konzipiert sind, dass sie erklärbar sind. Hierbei zählen z.B. Entscheidungsbäume oder regelbasierte Systeme eingesetzt werden.
  - Local Explanation: Hierbei wird eine individuelle Prognose erklärt, indem die wichtigsten Merkmale und ihre Ausprägungen gezeigt werden, die zu einem bestimmten Ergebnis führen. Methoden: SHAP-Values oder LIME.

# XAI - Intro

---



... Fortsetzung

- Global Explanation: Hierbei wird, die Prognosefähigkeit eines ML-Modells ganzheitlich erklärt, indem die Wichtigkeit von Merkmalen & Merkmalsgruppen gezeigt wird. Methoden: Variable Importance, Partial Dependence Profile & Accumulated Local Dependence Profile.
- Die Umsetzung von XAI-Methoden kann dazu beitragen, das Vertrauen in KI-Systeme zu erhöhen, indem sie Transparenz und Nachvollziehbarkeit in den Entscheidungsprozess bringen.
- Dadurch kann die Akzeptanz von KI-Systemen in verschiedenen Bereichen wie z.B. der Medizin, der Rechtswissenschaften oder der Finanzindustrie gesteigert werden.

# Break-Down & Break-Down Plus

---



Foto von [Milivoj Kuhar](#) auf [Unsplash](#)

## Break-Down

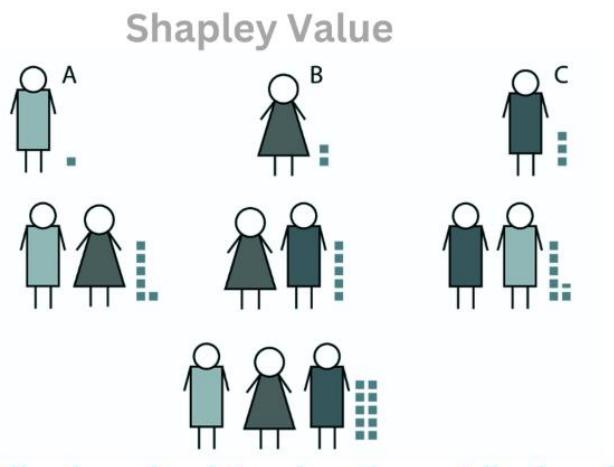
- Die Methode berechnet den Beitrag jedes Merkmals zur Vorhersage, indem sie die Vorhersage für jeden einzelnen Datenpunkt schrittweise aufteilt.
- Die Idee dahinter ist, dass die Vorhersage einer Beobachtung in zwei Teile zerlegt werden kann: einen Grundanteil, der durch den Durchschnittswert aller Vorhersagen erklärt werden kann, und einen Beitragsteil, der durch den Einfluss der einzelnen Merkmale erklärt wird.
- Bei der Break-Down-Methode wird der Beitrag eines Merkmals als die Differenz zwischen der Vorhersage für die gesamte Beobachtung und der Vorhersage für die Beobachtung ohne dieses Merkmal berechnet.

## Break Down Plus

- Break Down erweitert um Beziehungen zwischen Merkmalspaaren bzw. Merkmalskombinationen, Beispiel: Fahrpreis und Passagierklasse.

# SHAP-Values

---



**The Shapley value determines the contribution of each player participating in a collaborative work.**



Quelle: [Shapley Value](https://wallstreetmojo.com/shapley-value/) (wallstreetmojo.com)

- Der Shapley-Wert ist ein in der Spieltheorie verwendetes Lösungskonzept, bei dem sowohl Gewinne als auch Kosten gerecht auf mehrere in Koalitionen zusammenarbeitende Akteure verteilt werden.
- Shapley-Werte können verwendet werden, um zu bestimmen, wie wichtig einzelne Merkmale für die Vorhersage von Ergebnissen durch ein ML-Modell sind.
- Die Idee hinter den Shapley-Werten ist, dass jedes Merkmal einen gewissen Beitrag zum Endergebnis hat, der auf seine einzigartige Fähigkeit zurückzuführen ist, das Ergebnis zu beeinflussen.
- Die Shapley-Werte berücksichtigen diese Beiträge und berechnen sie, indem sie alle möglichen Kombinationen von Merkmalen berücksichtigen.
- Die Verwendung von Shapley-Werten kann dazu beitragen, das Verständnis darüber zu verbessern, welche Merkmale die Vorhersage von Ergebnissen beeinflussen.
- Dies kann wiederum dazu beitragen, das Vertrauen in das KI-Modell zu stärken und die Transparenz und Erklärbarkeit von Entscheidungen zu verbessern.

# SHAP-Values - Beispiel

A-> Alone will pay 50 to go his home  
B->Alone will pay 60 to go his home  
C->Alone will pay 80 to go his home  
If A and B car pool they will pay 70.  
If A and C car pool they will pay 100.  
If B and C car pool they will pay 90.  
Lets say A,B and C go in pooled car and pay 120 to driver.

Combination	A	B	C
A B C	50	20	50
B A C	10	60	50
A C B	50	20	50
C A B	20	20	80
B C A	30	60	30
C B A	30	10	80

1

So the average value for A, B and C

$$A = 31.67$$

$$B = 31.67$$

$$C = 56.67$$

2

Beispiel: Shapley Value: Explaining AI. Machine learning is gradually becoming... | by Anantech.ai | Geek Culture | Medium

Nehmen wir an, A, B und C wollen nach einer Party Fahrgemeinschaften bilden. Folgende Annahme:

- A-> Allein zahlt 50, um nach Hause zu fahren
- B-> Allein zahlt 60, um nach Hause zu fahren
- C-> Allein zahlt 80, um nach Hause zu fahren
- Wenn A und B zusammen fahren, zahlen sie 70.
- Wenn A und C zusammen fahren, zahlen sie 100.
- Wenn B und C zusammen fahren, zahlen sie 90.
- Wenn A, B und C zusammen fahren zahlen 120.

Was ist eine faire Verteilung der Fahrkosten unter Berücksichtigung der Reihenfolge des Einsteigens.

1. Berechnung der Anteile für jede Kombination
2. Mittelwertbindung der Einzelergebnisse

Beispiele Berechnung der Werte zu 1.:

ABC: A: 50 + B: 20 (AB=70-50) + C: 50 (Rest: 120-70) 50

BCA: B: 60 + C: 30 (BC=90-60) + A:30 (Rest: 120-90)

# Übersicht XAI - Techniken

XAI-Technik	Beschreibung	Bibliotheken
LIME	Erklärbarkeit auf der Ebene einzelner Vorhersagen	Lime, Skater, Innvest
SHAP	Berechnet die Beiträge jedes Merkmals zur Vorhersage des Modells	SHAP, Skater, Innvest, Dalex
Break Down	Eine Methode zur Aufschlüsselung des Vorhersagebeitrags jeder Feature-Variable in einem bestimmten Modell	Dalex
Permutation Feature Importance	Ermittelt die Wichtigkeit der Merkmale durch Permutation	ELI5, Skater, Innvest
Anchors	Entdeckt Regeln, die die Vorhersage des Modells erklären	Alibi-Explain
Counterfactuals	Findet alternative Eingaben, um das Modellverhalten zu erklären	Alibi-Explain, What-If
Prototypes	Generiert Prototypen, die typische Beispiele für Vorhersagen darstellen	Alibi-Explain
Partial Dependence Plots	Stellt die Abhängigkeit der Vorhersage von einem Merkmal grafisch dar	Skater, Innvest, Dalex
Feature Importance	Ermittelt die wichtigsten Merkmale für die Vorhersage des Modells	Skater, Innvest, InterpretML, Dalex
Local Surrogate Models	Erstellt lokale, verständliche Modelle, um das Modellverhalten zu erklären	Skater, Innvest, InterpretML
SHAPley Interaction Index	Ermittelt die Interaktionen zwischen Merkmalen	SHAP

# Anhang

---



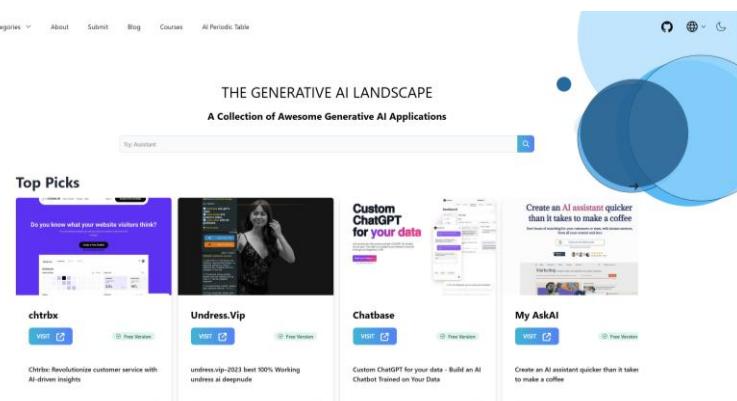
# KI-Tools

---



# KI-Tools

The screenshot shows the homepage of KI-Tools. At the top, a banner reads "THERE'S AN AI FOR THAT". Below it, a section titled "Our favorites" lists tools like SEO Writing AI, Voicify, and Happy Copy. A "Just launched" section features Player2tax, JobSearch.Coch, and Tenorshare. On the left, there's a sidebar for "Agen4" and "Sphinc Mind". The footer includes links for Home, Categories, About, Submit, Blog, Courses, and AI Periodic Table.



The screenshot of Futurepedia shows its homepage. The header includes the Futurepedia logo, navigation links for Favorites, Deals, Advertise, Resources, and Subscribe, and statistics for 150K tools and 350K news items. The main search bar contains the query "I want to create a Twitter thread". Below the search bar are filters for categories like SEO, Summaries, Chatbots, Music, Funny, and more. A "Trending AI tools" section lists VideoToBlog, BG Remover, FollowFox, Amazon Sage Maker, Posit, and Gledia. The footer includes links for Home, Categories, About, Submit, Blog, Courses, and AI Periodic Table.

This block displays a grid of 12 articles from Futurepedia. The articles include:

- 10 Best Midjourney Anthropomorphic Prompts (January 13, 2024)
- 20 Best ChatGPT Prompts for Book Writing (January 9, 2024)
- 10 Best Midjourney Prompts for Wall Art (January 28, 2024)
- 70+ ChatGPT Prompts for Email Marketing (January 13, 2024)
- 120+ Best ChatGPT Prompts for Social Media (January 6, 2024)
- 40+ Best Coloring Book Pages Prompts for Midjourney (December 23, 2023)
- 10 Best ChatGPT Prompts for Fiverr to Create Gigs in Seconds (December 31, 2023)
- 40 Best ChatGPT Prompts for Farming (Agriculture Prompting) (November 21, 2023)
- How to Write Better ChatGPT Prompts (5 Step Guide) (December 31, 2023)

Each article includes a thumbnail image, the title, the author (MUKUND KAPOOR or SAMUEL BRAINARD), the publication date, and a brief description.

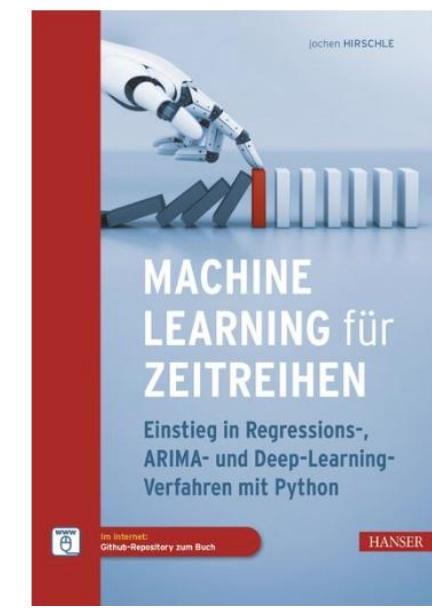
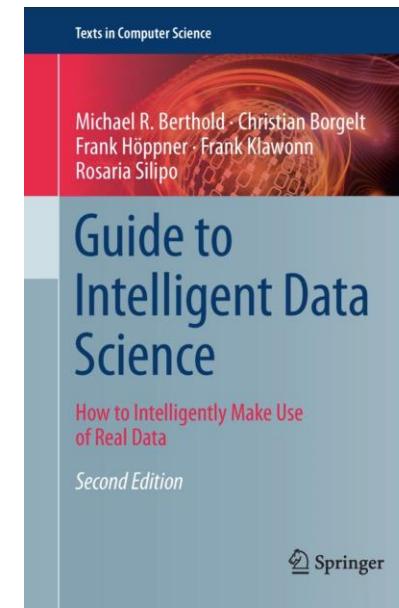
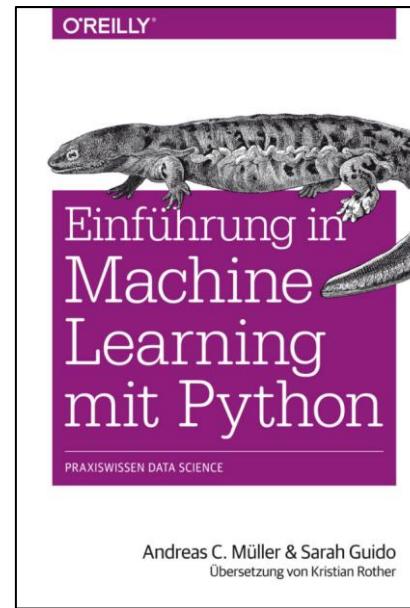
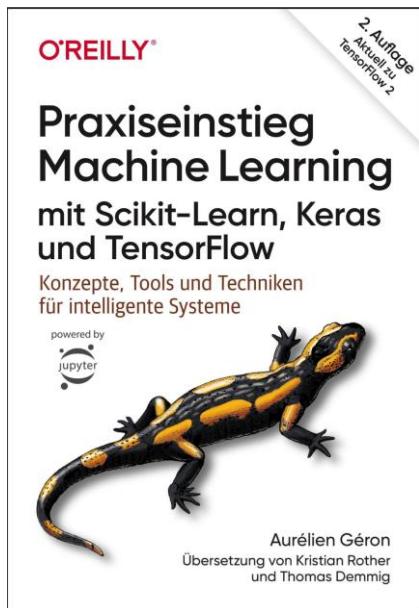
# Links, Videos, ebooks, Bücher

---



# Machine Learning - Bücher

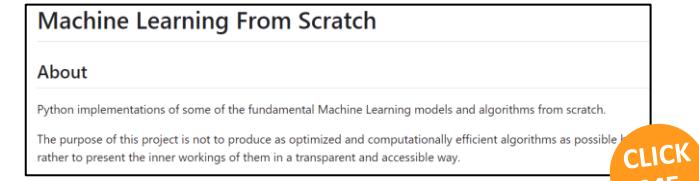
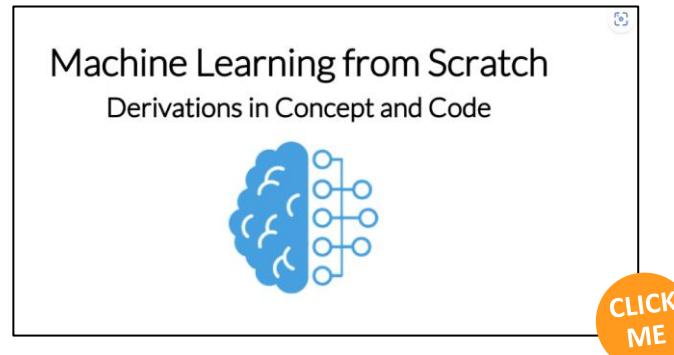
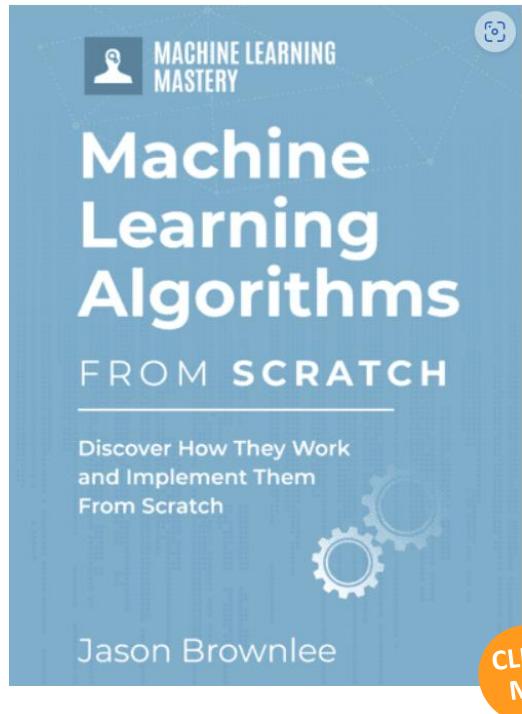
---



Beispiele  
Jupyter  
Notebooks

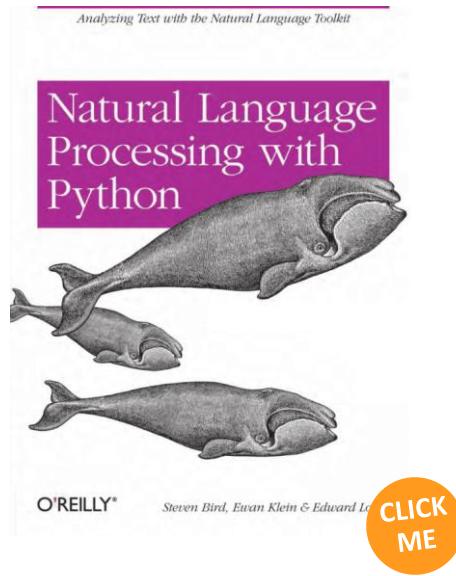
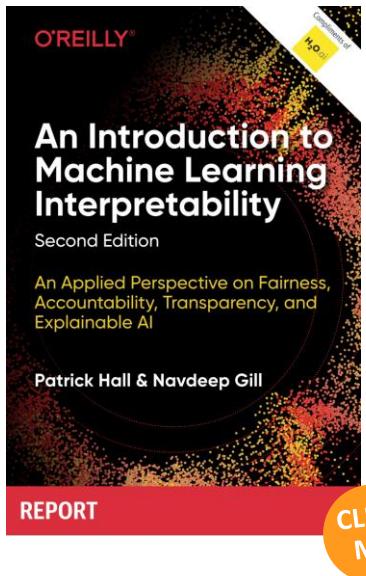
# ML - Python from Scratch, No Library

---



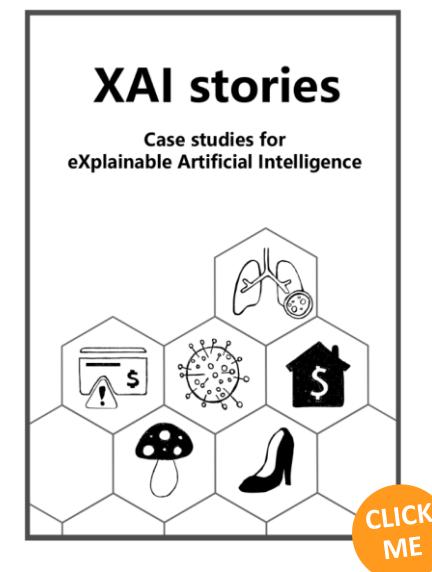
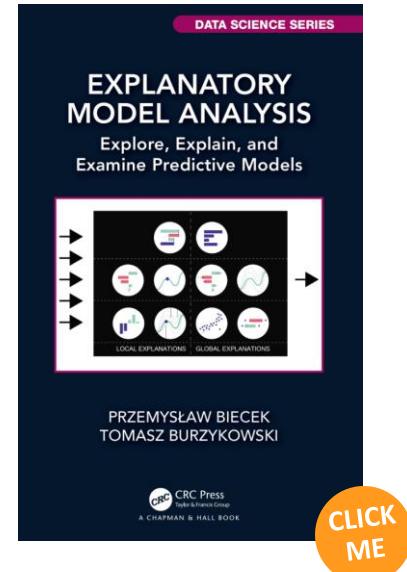
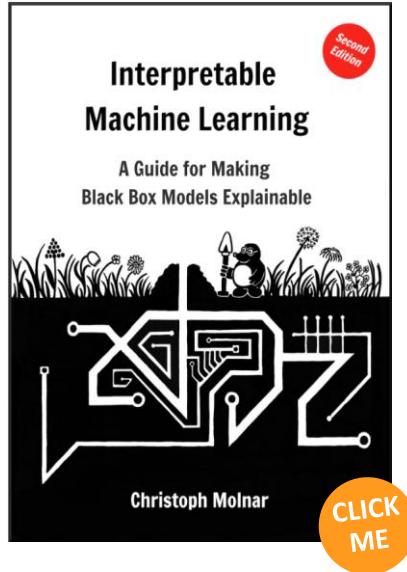
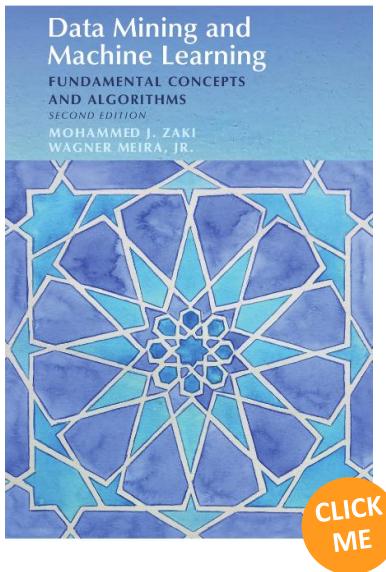
# Machine Learning - ebooks, Websites

---



# Machine Learning - ebooks, Websites

---



Beispiele  
Jupyter  
Notebooks

# ML - Verstehen und ausprobieren

---

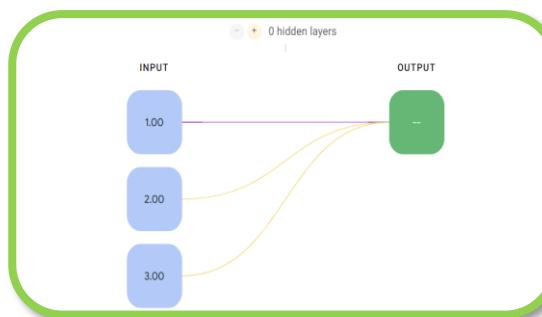


**AI by Hand**

with Prof. Tom Yeh

By Tom Yeh · Over 7,000 subscribers

**AI-by-Hand-exercises-Tom-Yeh**



**IMAGINARY**  
open mathematics

Events   Programme   Galerien   Hands-on   Filme   Texte   Ausstellungen



# Module & Pakete

---



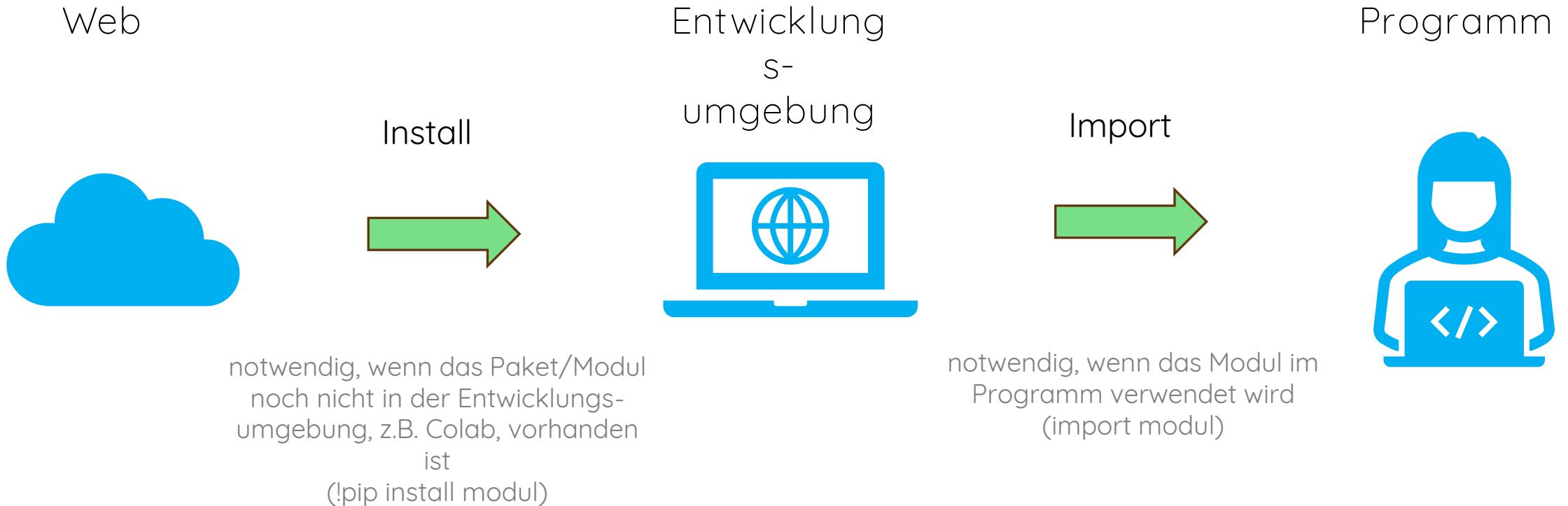
# Modul

---

- In Python ist ein Modul eine Datei, die Funktionen, Klassen und Variablen enthält.
- Module werden verwendet, um den Code in wiederverwendbare Teile aufzuteilen, was dazu beitragen kann, den Code effizienter zu gestalten und ihn einfacher zu warten.
- Ein Modul kann über das Import-Statement in einem anderen Python-Skript oder Modul geladen werden, um auf dessen Funktionen, Klassen und Variablen zuzugreifen.
- Das Import-Statement wird in der Regel am Anfang des Skripts oder Moduls verwendet.
- Python verfügt über eine Vielzahl von Standardmodulen, die für eine Vielzahl von Aufgaben verwendet werden können.
- Es gibt auch eine große Anzahl von Drittanbietermodulen, die von der Python-Community entwickelt wurden und für spezielle Aufgaben oder Anwendungen nützlich sein können.

# Install & Import

---



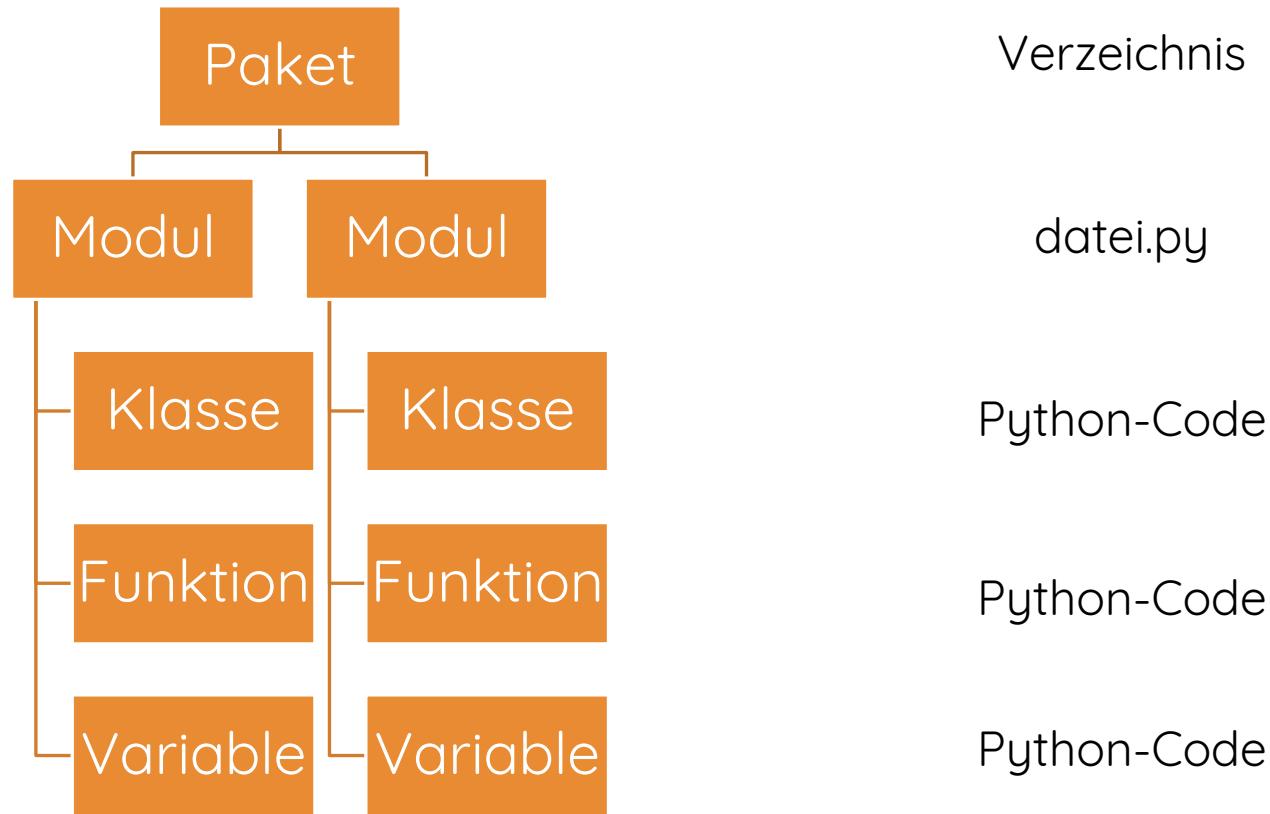
# Python Package Index (PyPI)

---

- Bei der Verwendung von pip install zum Installieren von Python-Bibliotheken werden die Pakete standardmäßig von Python Package Index (PyPI) heruntergeladen. PyPI ist eine Software-Repository für Python, die eine umfangreiche Sammlung von Open-Source-Paketen für die Python-Programmiersprache bereitstellt.
- PyPI agiert als zentrale Anlaufstelle für Python-Entwickler, um ihre Software zu teilen, damit andere sie leicht finden und installieren können. Wenn pip install <paketname> ausgeführt wird, sucht pip nach dem Paketnamen in PyPI, lädt das Paket und seine Abhängigkeiten herunter und installiert sie dann in der Python-Umgebung.
- Man kann auch Pakete von anderen Quellen als PyPI installieren, indem man die URL der Paketdatei oder eines Git-Repositories angibt. Zum Beispiel:
  - Um ein Paket direkt von einer URL zu installieren: pip install <Paket-URL> verwenden.
  - Um ein Paket von einem Git-Repository zu installieren: pip install git+<Repository-URL>.
  - Es ist auch möglich, pip so zu konfigurieren, dass es Pakete von einem anderen Index als PyPI sucht, indem man die Option --index-url verwendet.
- Zur Frage der Qualität von Bibliotheken sie u.a. [How to Evaluate the Quality of Python Packages – Real Python](#)

# Pakete & Module

---



# Keras

---



# Keras Model Building

---



Bild von [Stefan Schweohofer](#) auf [Pixabay](#)

- Das **sequentielle Modell**, es ist sehr einfach ist ein Stapel von Schichten (Layern) mit einer einzigen Eingabe und einer einzigen Ausgabe beschränkt ist (wie der Name schon sagt).
- **Funktionale API**, ermöglicht komplexere Architekturen, einschließlich Multi-Input- oder Multi-Output-Modelle, gemeinsam genutzte oder nicht gemeinsam genutzte Schichten und Modelle mit Restverbindungen.
- **Modellunterklassen**, bei der man alles von Grund auf selbst implementiert. Zu verwenden, wenn man komplexe, sofort einsatzbereite Forschungsanwendungsfälle hat.
- Darüber hinaus können auch **vortrainierte Modelle** von Keras für Ihre eigenen Modelle verwenden werden. Dies ist z.B. nützlich, wenn nur begrenzte Daten verfügen sind, da vortrainierte Modelle als **starke Merkmalsextrahierer** dienen können, sodass man ein gutes Modell mit begrenzten Daten trainieren kann.

# Callback

---



Bild von [Pexels](#) auf [Pixabay](#)

- Keras Callback ermöglicht, während des Trainingsprozesses eines Modells spezifische Aktionen auszuführen oder auf bestimmte Ereignisse zu reagieren.
- Callbacks bieten eine flexible Möglichkeit, in den Trainingsprozess einzugreifen, ihn zu überwachen und zu steuern, ohne den eigentlichen Trainingscode zu verändern.
- Einige der häufigsten Keras Callbacks sind:
  - ModelCheckpoint: Speichert das Modell oder Modellgewichte in regelmäßigen Abständen, so dass das Modell zu einem späteren Zeitpunkt geladen und verwendet oder weiter trainiert werden kann.
  - EarlyStopping: Beendet das Training frühzeitig, wenn eine überwachte Metrik sich für eine festgelegte Anzahl von Epochen nicht verbessert, was hilft, Overfitting zu vermeiden und Rechenzeit zu sparen.

# Callback

---



Bild von [Pexels](#) auf [Pixabay](#)

- ReduceLROnPlateau: Verringert die Lernrate, wenn eine Metrik sich nicht mehr verbessert, was dabei helfen kann, aus lokalen Minima herauszukommen und die Modellleistung zu verbessern.
- TensorBoard: Ermöglicht die Visualisierung von Trainings- und Validierungsmetriken in TensorBoard, einem Tool für die Visualisierung von Lernprozessen in TensorFlow.

# Compile Loss & Metric

---

- Beim Kompilieren eines Modells in Keras sind "Loss" und "Metrics" Parameter die unterschiedliche Zwecke erfüllen.
- Loss (Verlustfunktion):
  - Die Verlustfunktion, oft einfach als "Loss" bezeichnet, ist eine Funktion, die quantifiziert, wie gut oder schlecht die Vorhersagen des Modells im Vergleich zu den tatsächlichen Daten sind. Die Loss-Funktion ist zentral für das Training, da der Optimierungsprozess (durch den Optimizer wie z.B. Adam oder SGD) darauf abzielt, diesen Verlust während des Trainings zu minimieren. Kurz gesagt, die Verlustfunktion ist das Kriterium, das das Modell verwendet, um sich selbst zu verbessern.
  - Sie beeinflusst direkt, wie die Gewichte des Modells angepasst werden, um die Modellleistung zu verbessern.
- Metrics (Metriken)
  - Metriken sind Funktionen, die verwendet werden, um die Leistung des Modells während des Trainings und der Validierung zu bewerten. Im Gegensatz zur Verlustfunktion sind Metriken nicht direkt in den Trainingsprozess eingebunden (d.h., sie beeinflussen nicht die Gewichtsanpassung des Modells). Stattdessen bieten sie Einblick in die Effektivität des Modells aus verschiedenen Perspektiven, wie Genauigkeit (Accuracy), Präzision, Recall, F1-Score usw.
  - Metriken sind optional und können mehrere ausgewählt werden. Sie sind nützlich für die Beurteilung und das Monitoring des Modells und helfen, die Leistung des Modells in einer für Menschen verständlichen Form zu verstehen.

# Statistik – ein paar Grundbegriffe

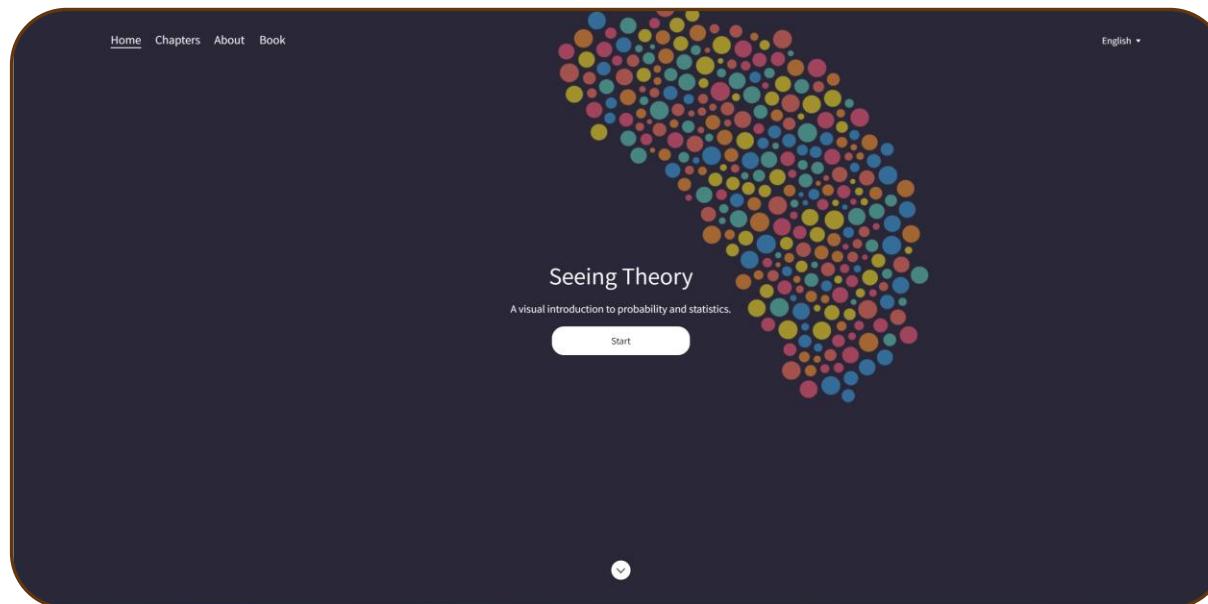
---



# Seeing Theory

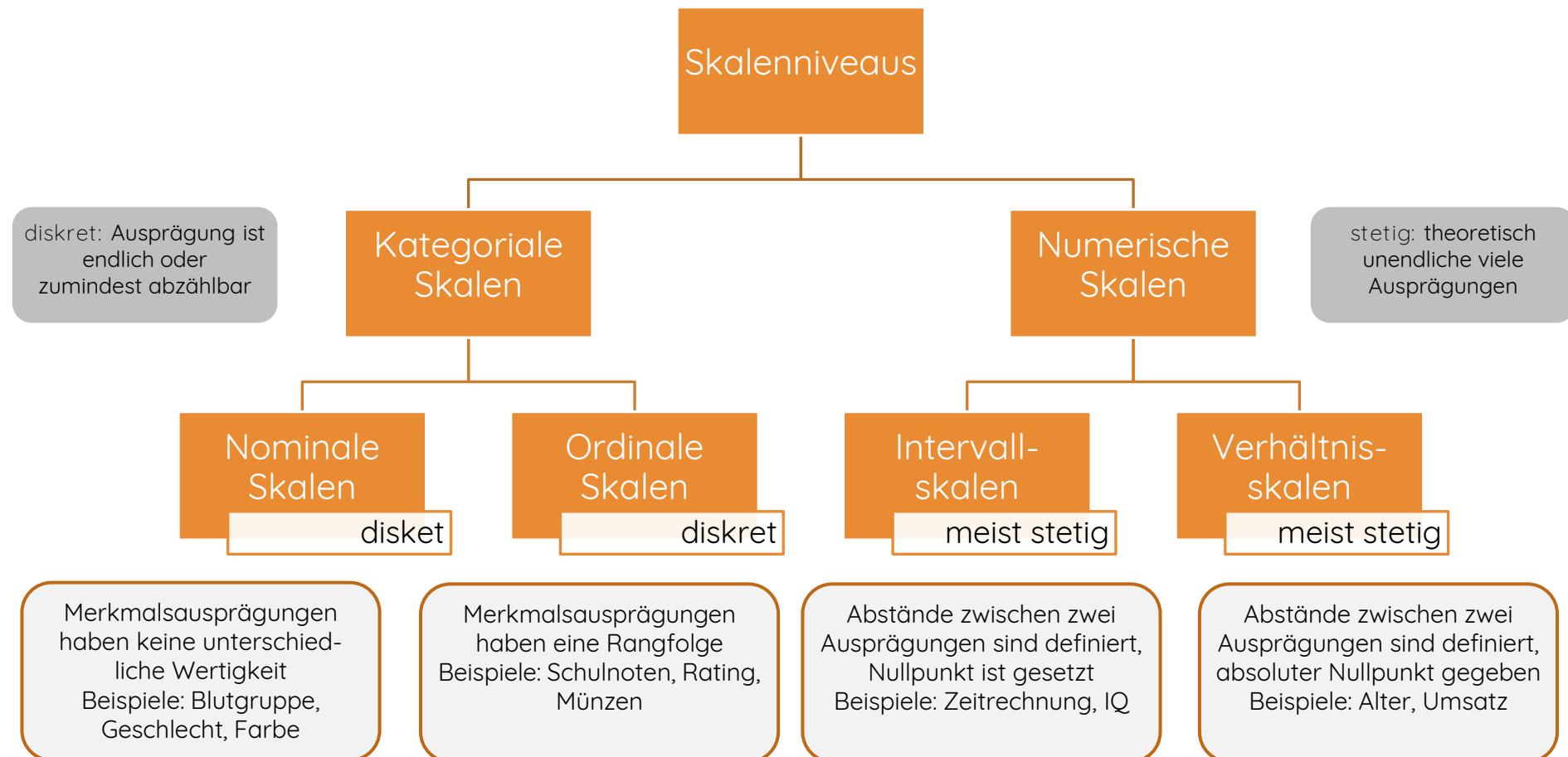
---

Seeing Theory ist ein von Daniel Kunin mit Unterstützung des Royce Fellowship Program der Brown University entworfenes und erstelltes Projekt. Ziel des Projekts ist es, Statistiken durch interaktive Visualisierungen einem breiteren Spektrum von Studierenden zugänglicher zu machen.



# Skalenniveaus

---



# Lageparameter

---



Maßgröße	Beschreibung	Wertebereich/ Bedeutung
Modus (Mode)	Messwert/Ausprägung einer Stichprobe, der am häufigsten vorkommt	Lageparameter
Mittelwert (Mean)	Summe aller Messwerte/Ausprägungen geteilt durch die Anzahl der Messwerte/ Ausprägungen (arithmetisches Mittel).	Lageparameter
Median (Median)	Der Median teilt Messwerte/Ausprägungen in zwei gleich große Hälften. Er ist also der Messwert, der genau in der Mitte liegt, wenn alle Messwerte in eine aufsteigende Rangreihe gebracht wurden.	Lageparameter
Min, Max	Kleinster, größter Wert der Messwerte	Lageparameter

# Steuerungsparameter

---

Maßgröße	Beschreibung	Wertebereich/ Bedeutung
Varianz (Variance) 	Maß für die Streuung von einer endlichen Anzahl von Messwerten/ Ausprägungen um den Mittelwert	Streuungs-parameter
Standard-abweichung (Standard Deviation) 	Maß für die Schwankungsbreite oder Streuung einer Reihe von Werten. Quadratwurzel der Varianz	Streuungs-parameter

# Schiefe & Kurtosis

---

Maßgröße	Beschreibung	Wertebereich/Bedeutung
Schiefe (Skewness)	Kennzahl, die die Art und Stärke der Asymmetrie einer Verteilung angibt (englisch skewness bzw. skew)	>0: rechtsschief, linkssteil <0: linksschief, rechtssteil 0: symetrisch
Kurtosis (Kurtosis)	Maßzahl für die Steilheit bzw. „Spitzigkeit“ einer Verteilung	>0: steilgipflig <0: flachgipflig

# Korrelation & Kausalität



Maßgröße	Beschreibung	Wertebereich/Bedeutung
Korrelation (Correlation)	Maß dafür, wie stark zwei Variablen zusammenhängen. Hängen zwei Variablen miteinander zusammen, dann lassen sich Aussagen darüber treffen, wie sich die Werte der einen Variable verhalten, wenn die Werte der anderen Variable ansteigen oder abfallen. Offen: Welche Variable beeinflusst die andere Variable.	1: positiver Zusammenhang -1: negativer Zusammenhang 0: kein Zusammenhang
Kausalität	Bedeutet, dass zwischen Variablen ein klarer Ursache-Wirkungs-Zusammenhang besteht. Klarheit: Welche Variable (bzw. Einflussfaktor) beeinflusst die andere Variable.	

# Hypothesen

---

Maßgröße	Beschreibung
Hypothese	Annahme, die geprüft wird.
Nullhypothese ( $H_0$ )	Die Nullhypothese besagt, dass kein Effekt/Zusammenhang besteht. Diese These soll verworfen werden, so dass die Alternativhypothese als Möglichkeit übrig bleibt.
Alternativhypothese ( $H_1$ )	Annahme oder Vermutung, die zur Erklärung bestimmter Effekte/Zusammenhänge dient und die der Nullhypothese entgegensteht. (auch: Gegenhypothese, Arbeitshypothese)

# Signifikanz

---

Maßgröße	Beschreibung
Signifikanz	Signifikanz macht eine Aussage darüber, ob ein Zusammenhang zwischen Variablen besteht, der über den Zufall hinausreicht.
Signifikanzniveau	<p>Das Signifikanzniveau (auch Irrtumswahrscheinlichkeit) gibt an, wie wahrscheinlich es ist, dass die Nullhypothese irrtümlich verworfen werden könnte (Fehler 1. Art).</p> <p>Als kritischer Schwellwert wird oft 5% (<math>\alpha = 0.05</math>) gewählt. Entsprechend beträgt dann die Wahrscheinlichkeit, eine richtige Nullhypothese aufgrund des statistischen Tests nicht abzulehnen, <math>1-\alpha=0.95</math>, also mindestens 95%.</p> <p>Hält man diese Folgen eher für gravierend, so wird man hier eher ein niedriges Niveau als ein Schwellwert wählen, beispielsweise 1% statt 5 %.</p>

# Sicherheit, Konfidenz, Cohen's Kappa

Maßgröße	Beschreibung
Sicherheitswahrscheinlichkeit	Die Sicherheitswahrscheinlichkeit ( $1-\alpha$ ) ist die Gegenwahrscheinlichkeit der Irrtumswahrscheinlichkeit $\alpha$ und kann selbst festgelegt werden. In den meisten Fällen wird die Sicherheitswahrscheinlichkeit (Konfidenzniveau) auf den Wert $\alpha = 0.95$ festgesetzt.
Konfidenzintervall	Ein Konfidenzintervall (auch Vertrauensintervall, Erwartungsbereich) ist ein Intervall, das die Präzision der Schätzung angeben soll. Das Konfidenzintervall gibt den Bereich an, der mit einer gewissen Wahrscheinlichkeit den Parameter einer Verteilung einer Zufallsvariablen einschließt.
Cohen's kappa	Statistisches Maß für die Urteilerübereinstimmung (real vs predict): <ul style="list-style-type: none"><li>• <math>k &lt; 0,0</math>: schlechte Übereinstimmung</li><li>• <math>k \leq 0,2</math>: etwas Übereinstimmung</li><li>• <math>k \leq 0,4</math>: ausreichende Übereinstimmung</li><li>• <math>k \leq 0,8</math>: beachtliche Übereinstimmung</li><li>• <math>k &gt; 0,8</math>: (fast) vollständige Übereinstimmung</li></ul>



# F-Score & P-Value

Maßgröße	Beschreibung
F-Score	Der F-Score misst die Abhängigkeit zwischen einem Merkmal/Feature und der Zielvariable. Ein hoher F-Score deutet darauf hin, dass das Merkmal/Feature besser zur Erklärung der abhängigen Variable/Zielvariablen beiträgt. Der F-Score kann einen beliebigen positiven Wert annehmen und ist nicht auf einen bestimmten Wertebereich beschränkt.
p-Wert (p-value)	Der p-Wert gibt an, wie signifikant diese Abhängigkeit ist. <ul style="list-style-type: none"><li>• <math>p\text{-Wert} &gt; 5\% = \text{keine signifikante Abhängigkeit}</math></li><li>• <math>p\text{-Wert} \leq 5\% = \text{signifikante Abhängigkeit}</math></li><li>• <math>p\text{-Wert} \leq 1\% (2,3 \text{ Standardabweichungen}) = \text{sehr signifikante Abhängigkeit}</math></li><li>• <math>p\text{-Wert} \leq 0,1\% (3,1 \text{ Standardabweichungen}) = \text{hoch signifikante Abhängigkeit}</math></li></ul>

# Hypothesentest vs. Modell

---

## Hypothesentest

- Überprüft ob und wie eine unabhängige Variable erklärende Wirkung auf die abhängige Variable hat.
- Beschreibt präzise eine Stichprobe und muss für neue Stichproben wiederholt werden.
- Güte ist das Signifikanzniveau! (90%, 95%, 99%, etc.)

## Modell

- Erläutert die erklärende Wirkung von unabhängigen Variablen auf die abhängige Variable.
- Ziel ist die allgemeine Beschreibung des Effekts. Präzision ist weniger relevant.
- Wichtiger als die Präzision ist die Übertragbarkeit auf neue Daten.
- Je höher die Präzision im Training, desto schlechter die Übertragbarkeit!

### INFO BOX

Signifikanz: Aussage, ob ein Zusammenhang zwischen Variablen besteht, der über den Zufall hinausreicht.