

# Background Workshopleiter: Dr. Thomas Hübsch



## Informatik

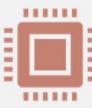
Dipl. Informatik

KIT - Karlsruher Institut für Technologie & Eastern Michigan University



## Promotion

medizinische Bildverarbeitung am Max-Planck-Institut  
für Kognitionswissenschaften & Universität Leipzig



## Berufliche Tätigkeiten

Trainer für Internettechnologien, Softwareentwickler, Projektleiter, Consultant,  
geschäftsf. Gesellschafter eines Multimedia Start-up

Post Doc Max-Planck-Institut für molekulare Genetik, wiss. Berater in der  
Projektförderung

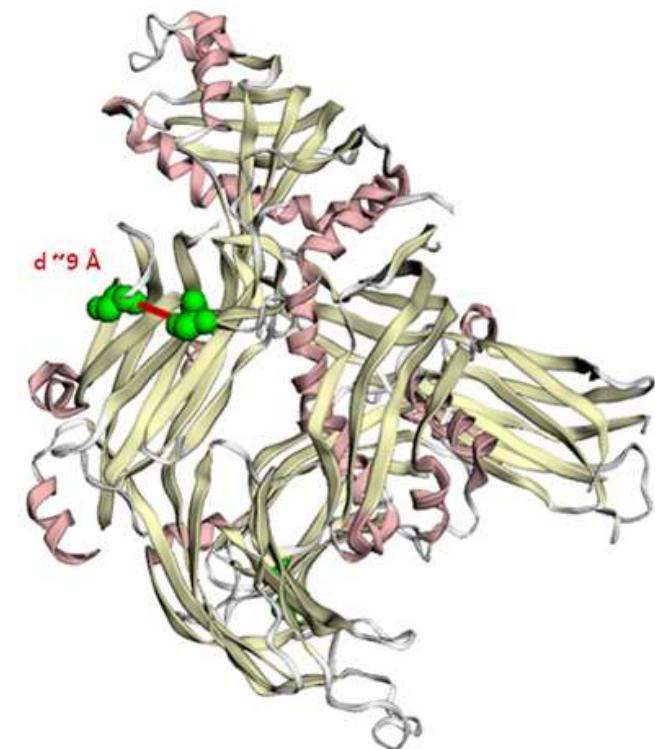
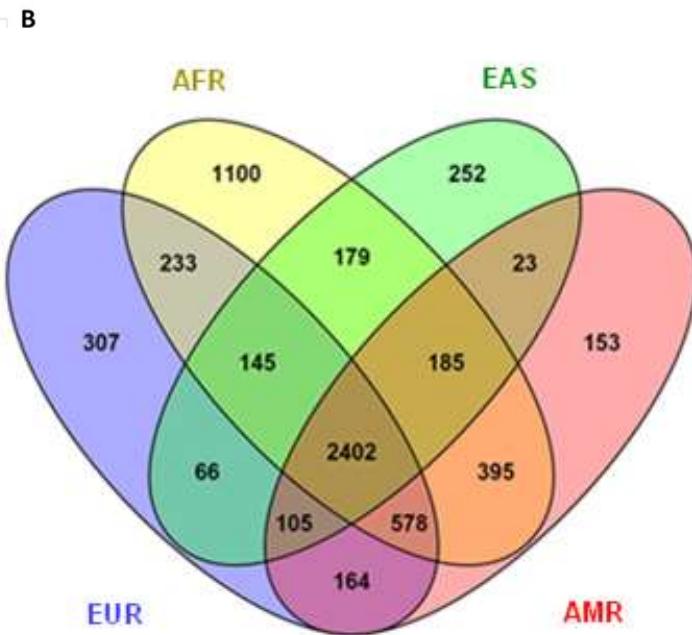
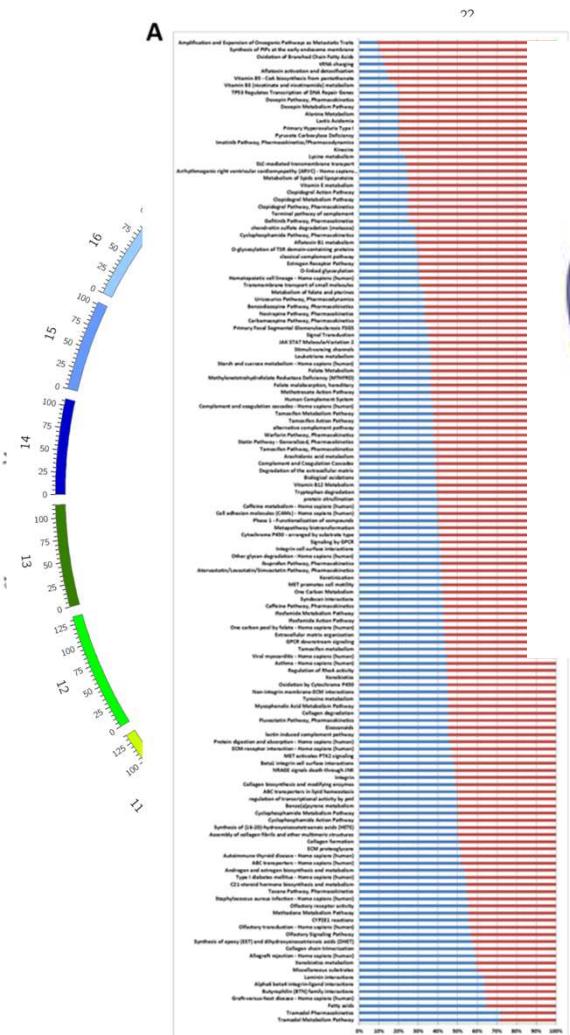


## Data Science Projekte

**E-Commerce Shop Analyse** Buchverlag (2002)

**3D MRT Daten** von Gehirnscans, **DNA Datenanalyse** Genome des Menschen  
Variabilität von Mutationen, **Clustering von unstrukturierten Textdaten**, ...

# Background DataScience



## **INVITE - Technologische Radarboards**

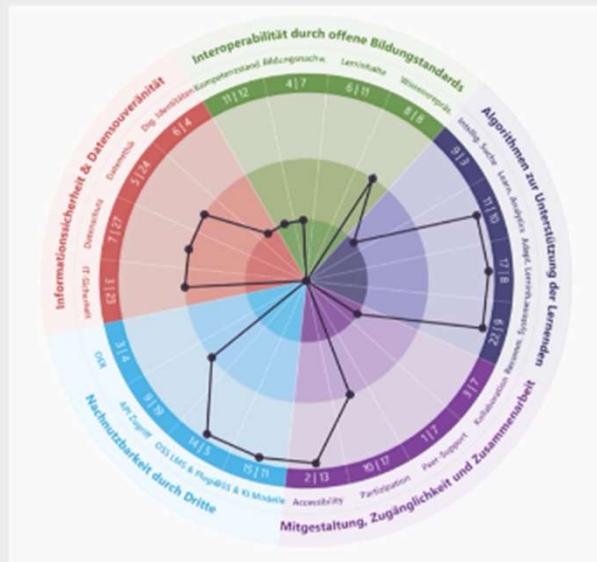
Die Radarboards dienen zur anschaulichen Darstellung und vergleichenden Analyse der vielfältigen Innovationen, die im Rahmen des Innovationswettbewerbs INVITE entwickelt wurden. Sie visualisieren sowohl die fachlich-didaktischen als auch die technologischen Schwerpunkte des Wettbewerbs und sind in Zusammenarbeit des BIBB mit der Digitalbegleitung VDI/VDE-IT entstanden.



## 1 Gesamtübersicht

Schwerpunkte aller INVITE Projekte auf einen Blick.  
Fachlich-Didaktisch & Technologisch.

Zur Ansicht wechseln...



## 2 Einzelprojekte

Schwerpunkte dargestellt pro Einzelprojekt.  
Fachlich-Didaktisch & Technologisch.

[Zur Ansicht wechseln...](#)



### ③ Einzeldimensionen

Ausprägung einzelner Dimension.  
Fachlich-Didaktisch & Technologisch.

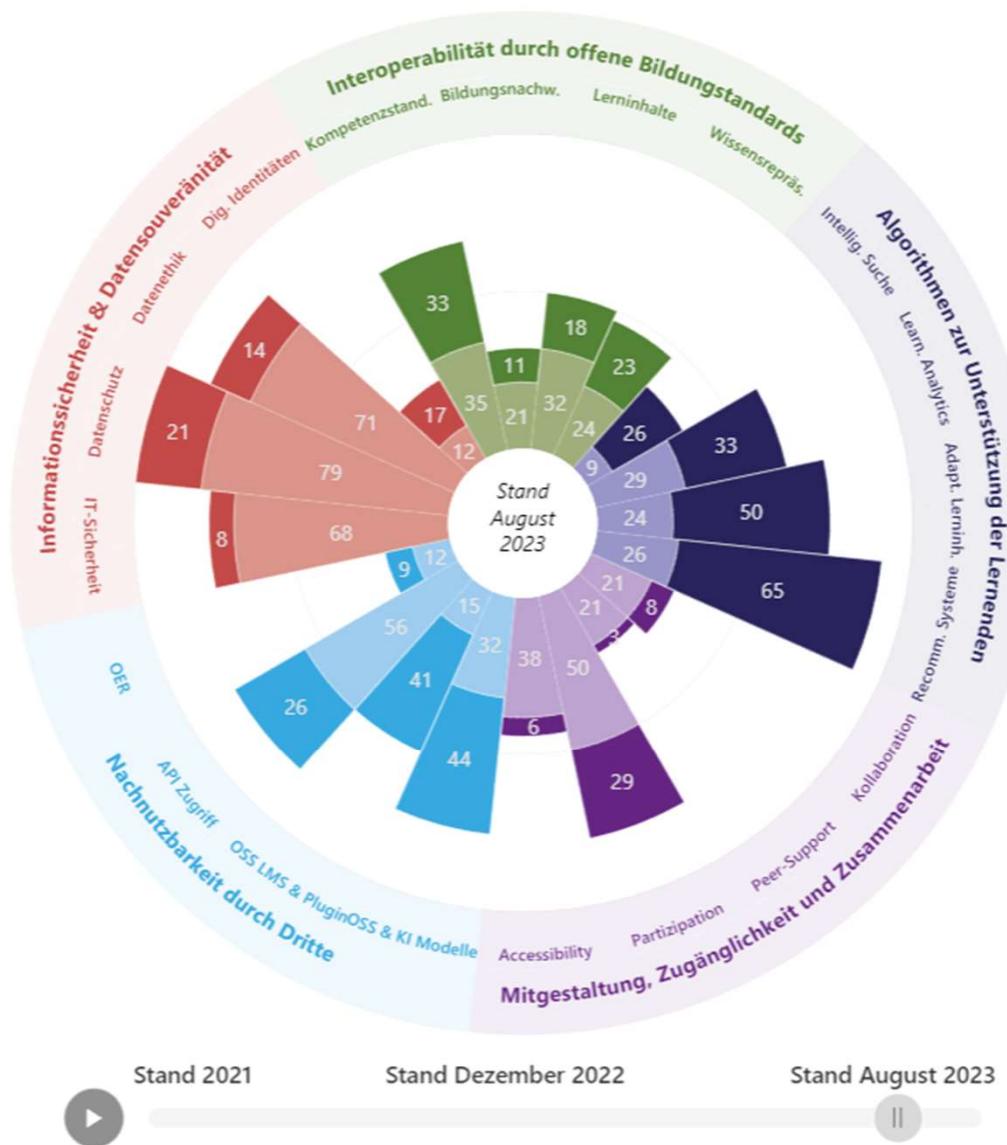
Zur Ansicht wechseln...

Dimensionen ○ Unterdimensionen ●

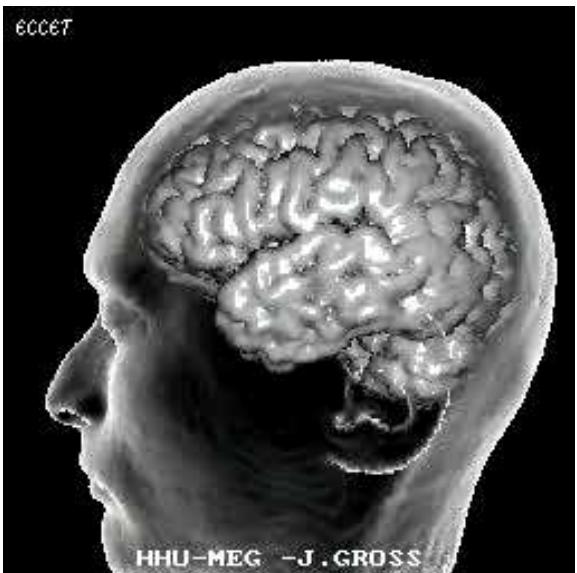
☰ Menü Visualisierung

## Legende

- Dimension ausgeprägt (%)
- Dimension vorhanden (%)
- Dimension in Ansätzen / keine Aussage (%)

 Filter anzeigen

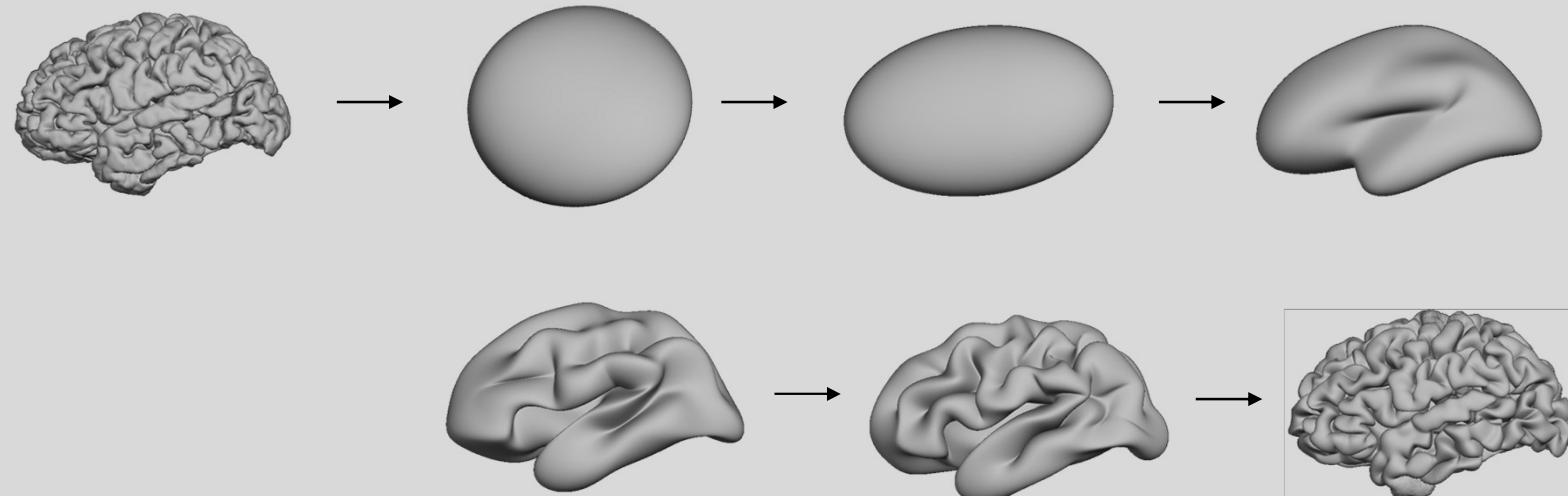
# Background DataScience



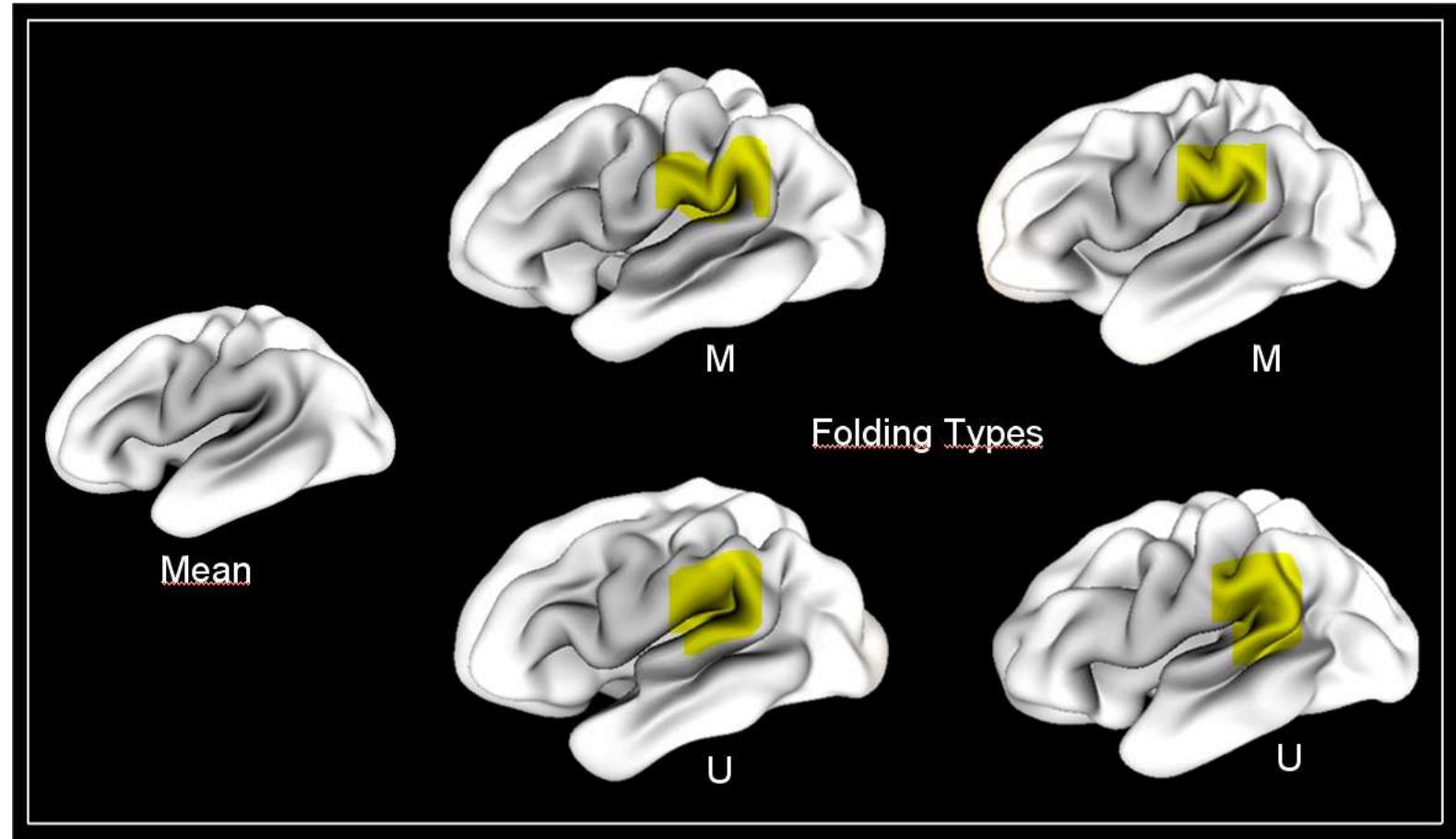
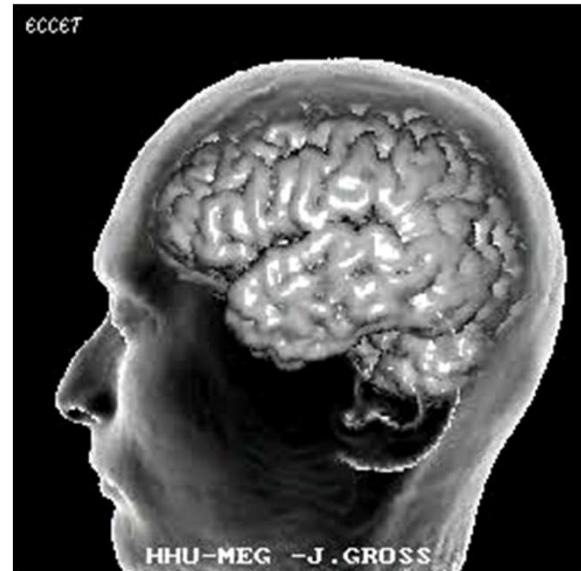
$$f(\theta, \phi) = (X, Y, Z) = \underbrace{\sum_{l=0}^{\infty} \sum_{m=-l}^l C_l^m Y_l^m(\theta, \phi)}_{\text{Cortical surface}} + \underbrace{C_0 + \sum_{m=-1}^1 C_1^m Y_1^m(\theta, \phi) + \sum_{m=-2}^2 C_2^m Y_2^m(\theta, \phi) + \dots}_{\text{Series of spherical harmonic basis functions}}$$

Cortical surface

Series of spherical harmonic basis functions

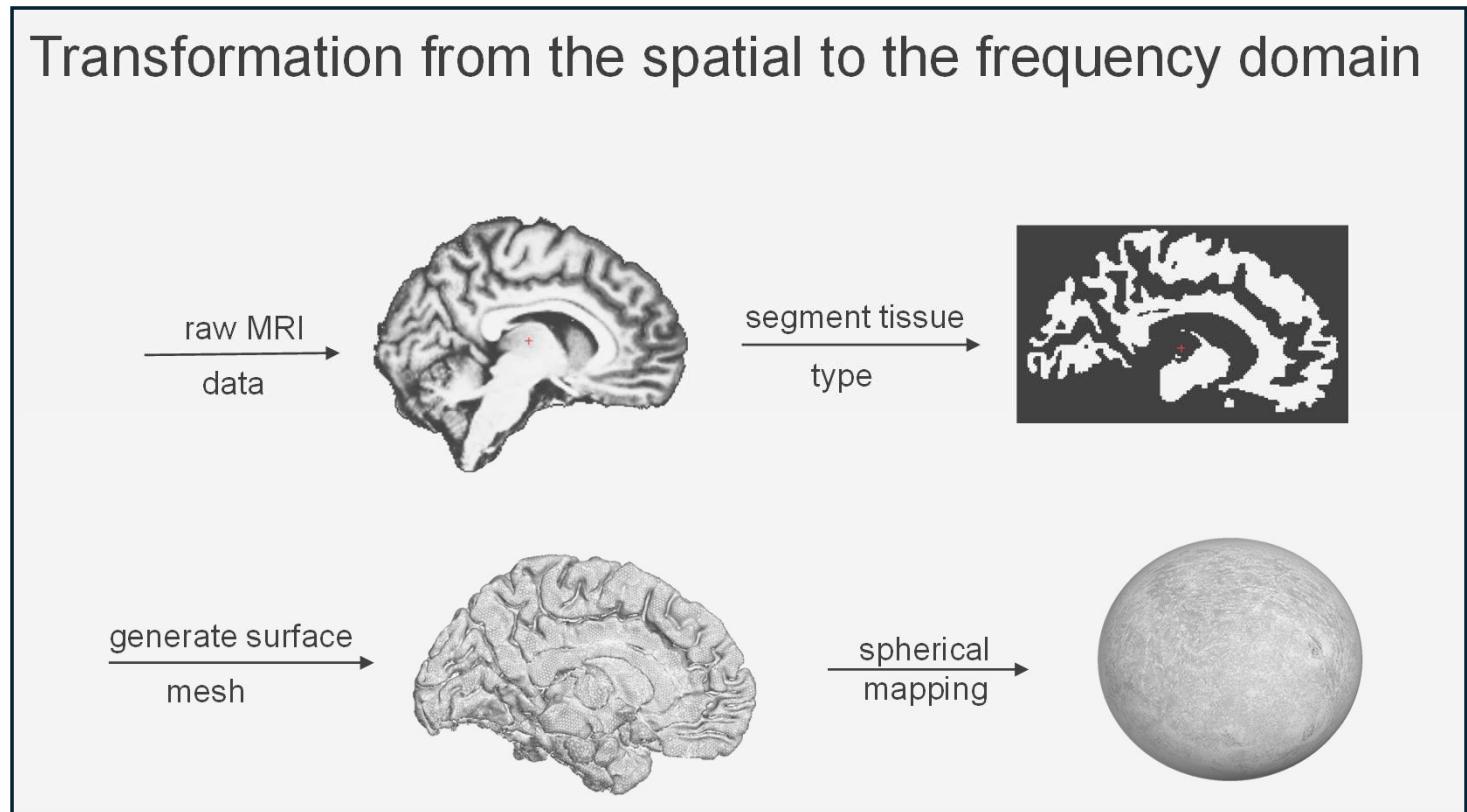
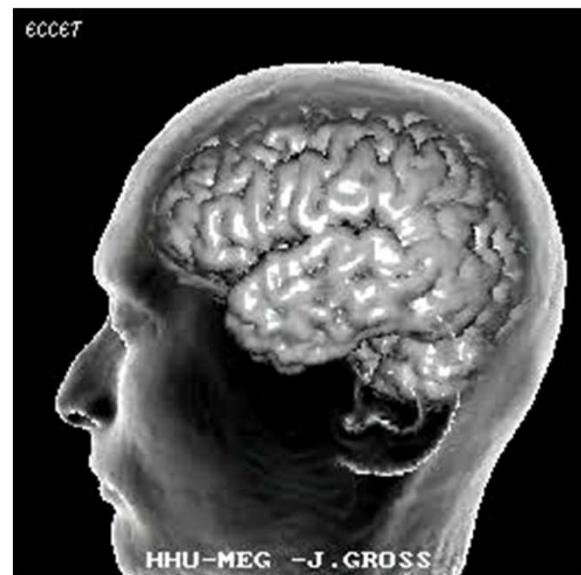


# Background DataScience

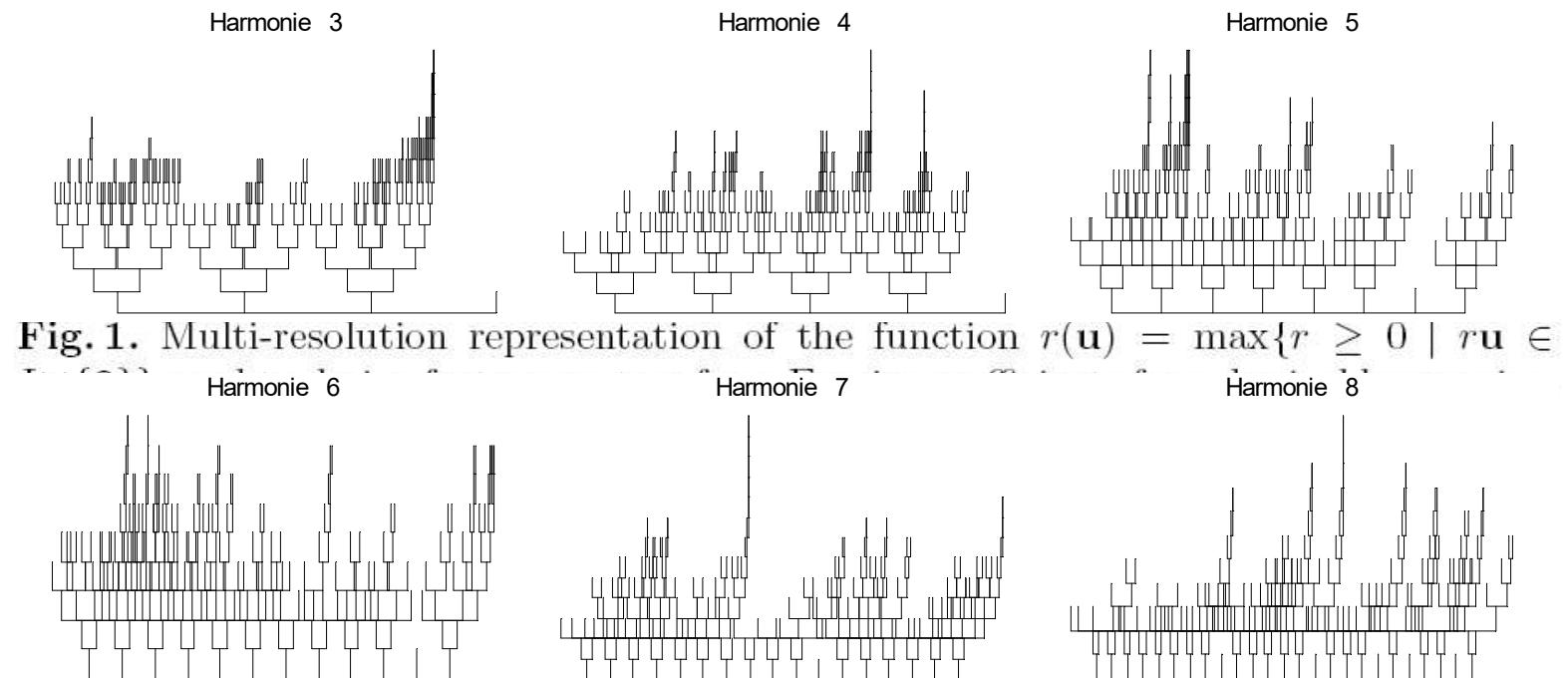
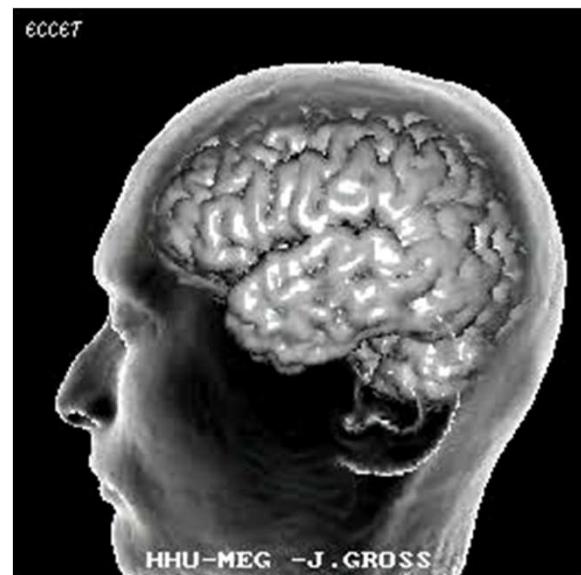


# Background DataScience

Transformation from the spatial to the frequency domain



# Background DataScience



# Organisatorisches

Gemeinsame Notizen und Links mit Padlet



Scanne den Code mit der Padlet-App auf deinem Smartphone oder Tablet.



[t1p.de/ds-pad1](https://t1p.de/ds-pad1)

# Data Science Summer School

## Montag Einführung in Data Science

Was ist Data Science?  
Überblick und  
Schlüsselkonzepte

Einführung in Data Mining:  
Verständnis der Prozesse  
und Werkzeuge

Übungen

## Dienstag Datenanalyse in der Praxis

Praktische Einführung in  
die Datenanalyse:  
Methoden der  
explorativen  
Datenanalyse

Anwendungsfälle aus  
verschiedenen  
Branchen: Wie  
Datenanalyse  
branchenübergreifend  
genutzt wird?

Übungen

## Mittwoch Maschinelles Lernen & Vorhersagemodelle

Grundlagen von  
Machine Learning:  
Algorithmen und deren  
Anwendungen

Neuronale Netze:  
Grundkonzepte und  
Anwendungsbeispiele

Übungen

## Donnerstag Generative KI & Anwendungsfelder

Einführung in generative KI:  
Was ist das und wie  
funktioniert es?

Praktische Anwendungen  
der generativen KI in der  
Datenwissenschaft,  
inklusive Beispiele und  
Fallstudien

Übungen

## Freitag Kommunikation & Storytelling mit Daten

Fortgeschrittene Techniken  
der Datenpräsentation:  
Wie man Daten visuell  
ansprechend und  
informativ darstellt.

Storytelling mit Daten:  
Konzepte und Techniken,  
um Datenanalysen effektiv  
zu kommunizieren

Workshop: Entwickeln Sie  
Ihre eigene Datenstory  
anhand eines Fallbeispiels

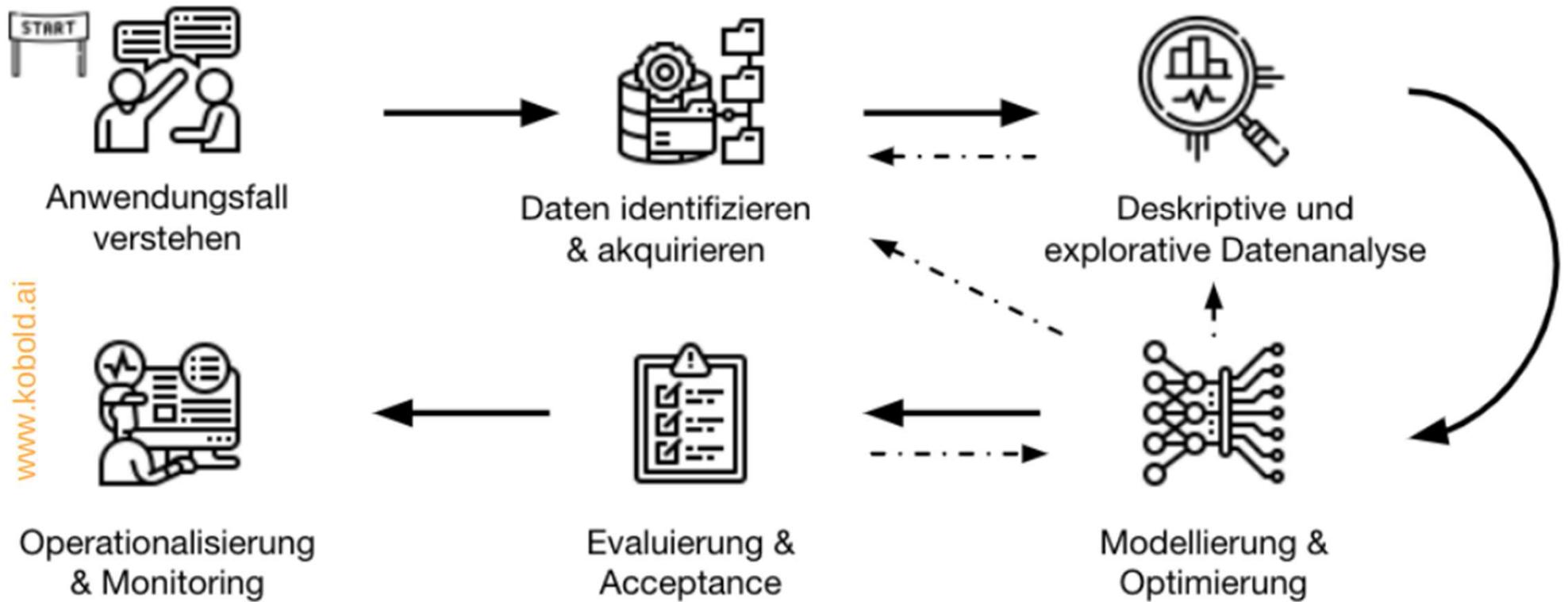
# Was ist Data Science: Überblick und Schlüsselkonzepte

## Zusammenfassung

- Data Science Prozess
- Widerstände überwinden
- Data Science Rollen
- Data Science Projektmanagement
- Tutor Projektmanagement



# Was ist Data Science: Überblick und Schlüsselkonzepte



# Was ist Data Science: Überblick und Schlüsselkonzepte

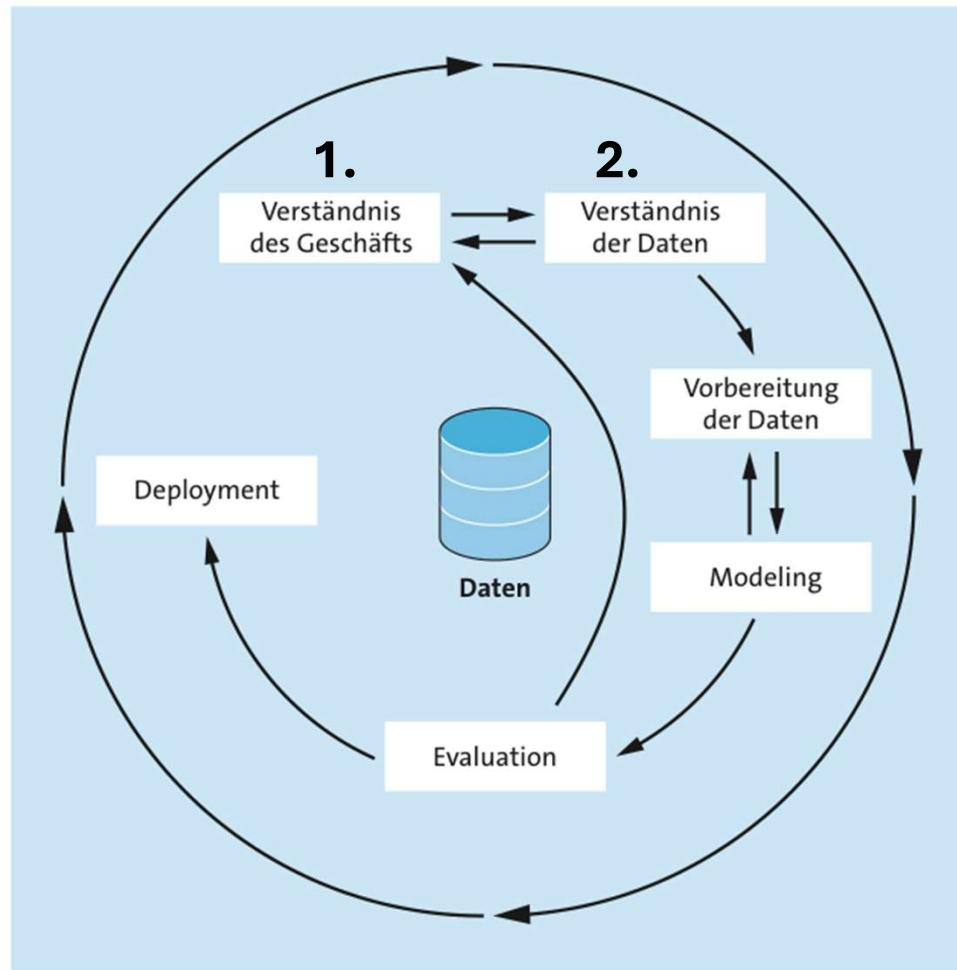
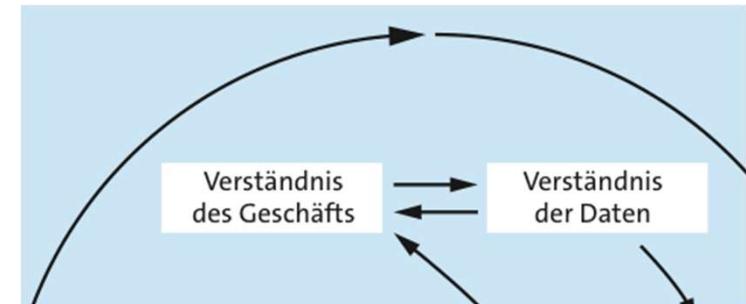


Abbildung 3.1 Das CRISP-DM-Modell (nach dem IBM SPSS Modeler CRISP-DM Guide)

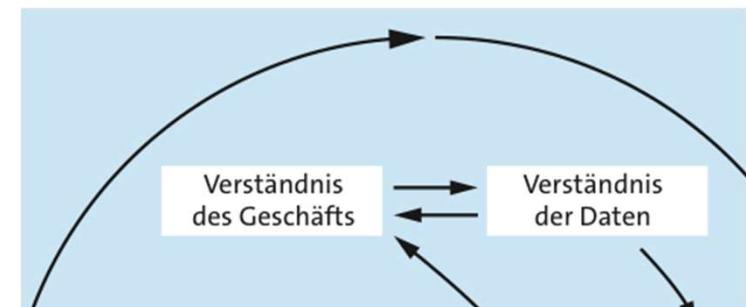
# Was ist Data Science: Überblick und Schlüsselkonzepte

- **Business Understanding:** Welches Problem soll gelöst werden?
- **Senior-Management-Unterstützung** und Einbeziehung der Fachabteilung
- **Data-Science-Projekte scheitern** meistens aus zwei Gründen:
  - Die Anforderungen wurden nicht richtig verstanden, oder das Problem wurde nicht richtig gelöst
  - Es fehlte die Unterstützung des Managements und der Fachabteilung.



# Was ist Data Science: Anforderungen verstehen

- Wichtiger als das Modell ist häufig zu verstehen worum es eigentlich geht
- Wie kann mit dem Data Science Projekt ein Mehrwert generiert werden ?
- Verstehen des Geschäftsmodell in dem Nutzende arbeiten inkl. der Ziele der jeweiligen Fachabteilung



# Das Zeitalter der Daten & KI = Hype?

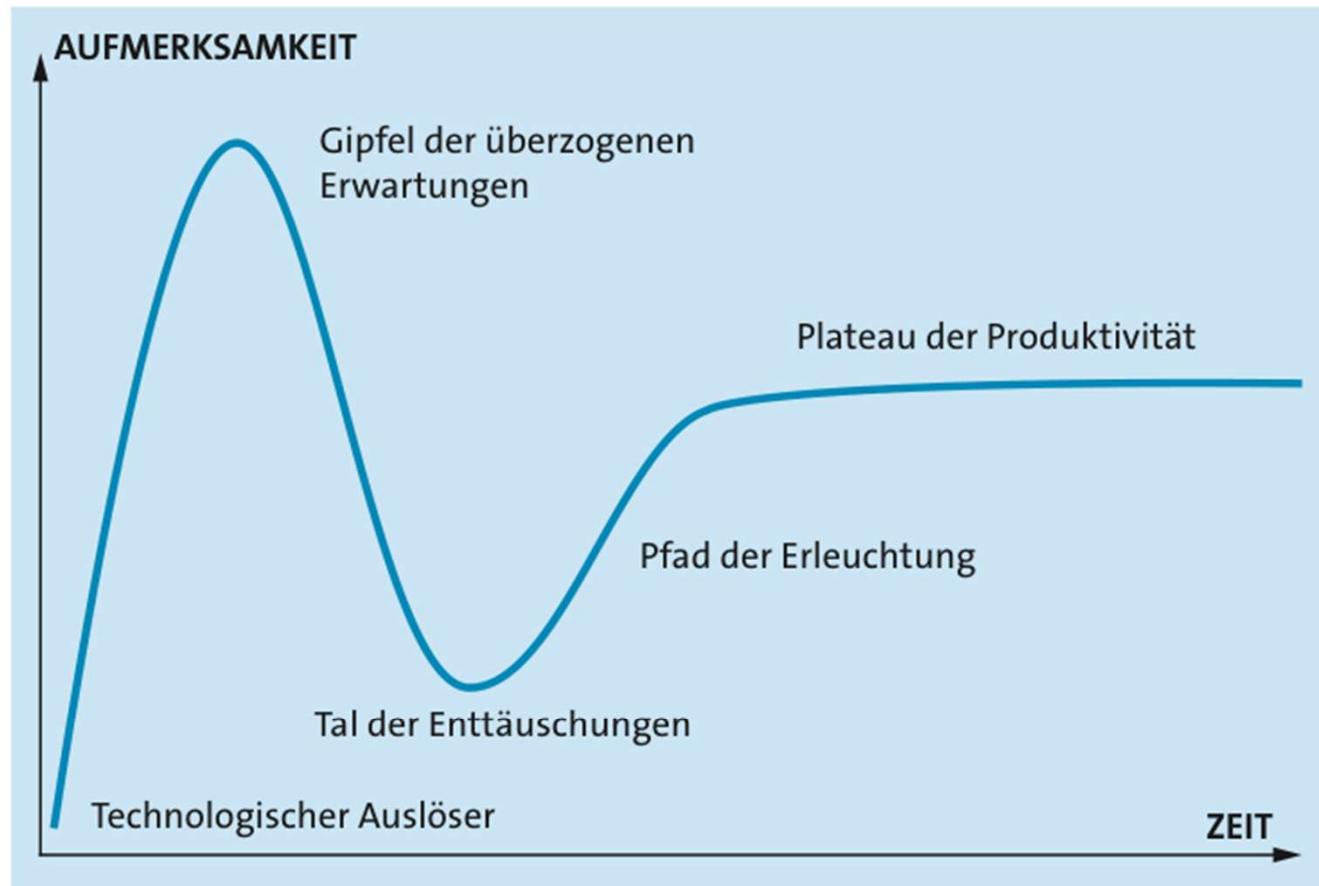


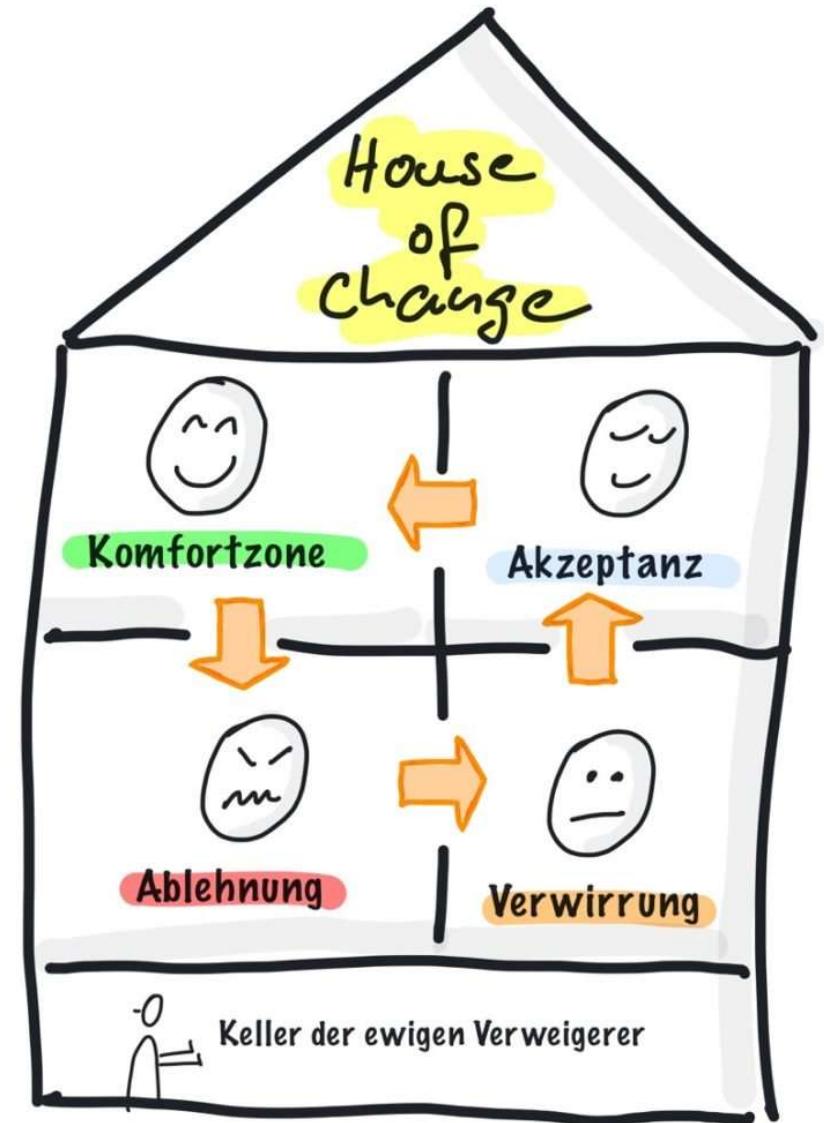
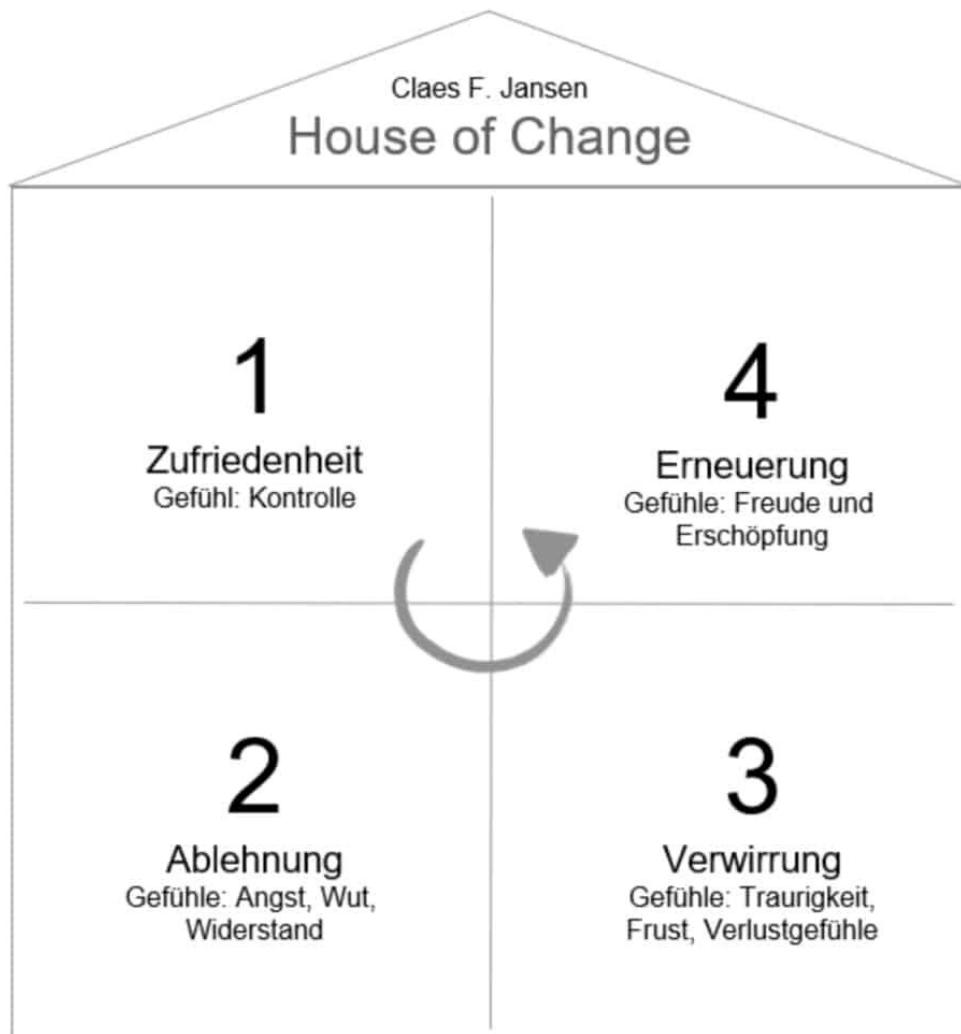
Abbildung 1.1 Der Hype-Zyklus nach Gartner, Quelle: Gartner (siehe [www.gartner.com/en/documents/3887767](http://www.gartner.com/en/documents/3887767))

# Data Science Projekt: Widerstände überwinden



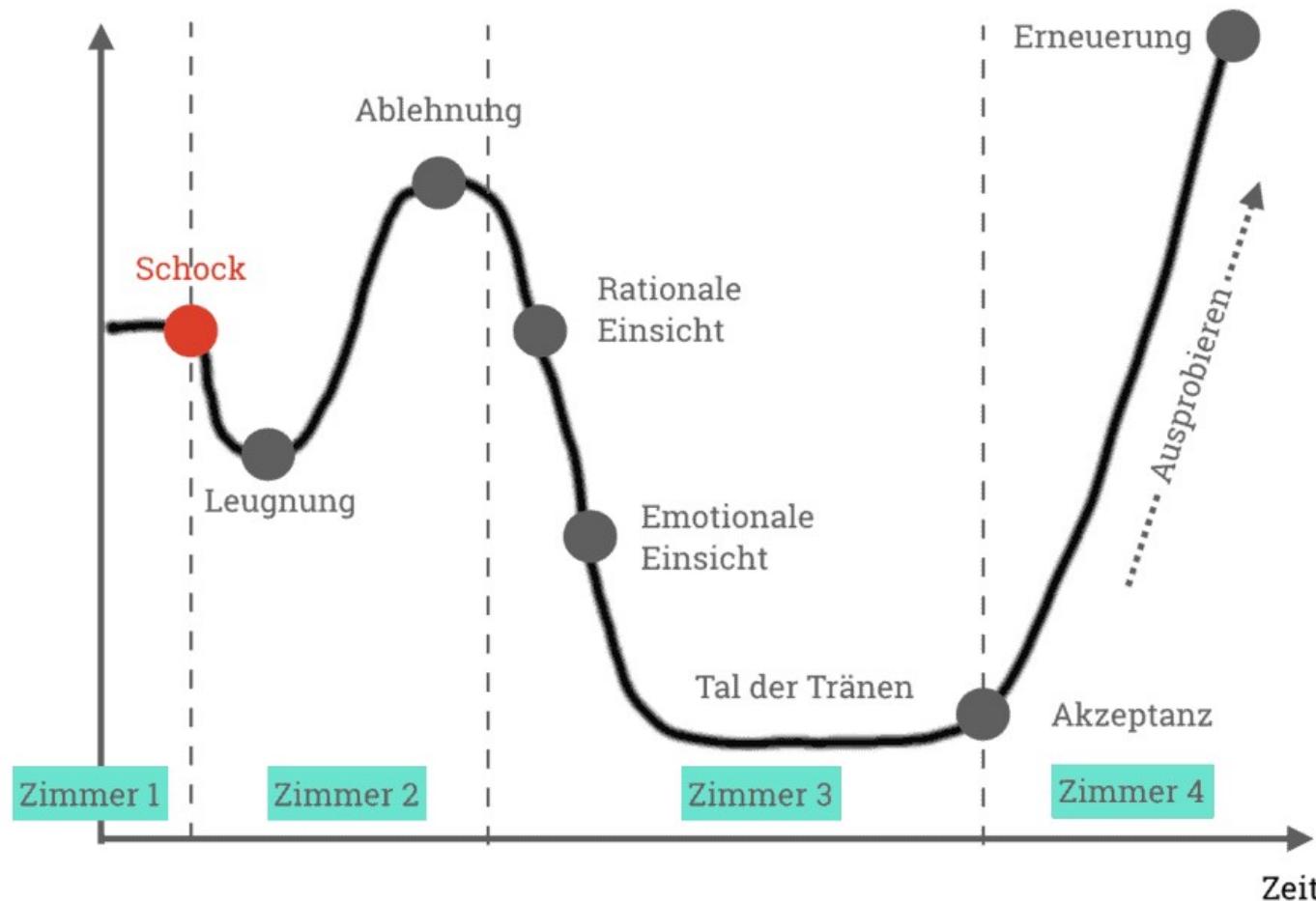
Abbildung 3.3 Titelseite der ZEIT Nr. 29 aus dem Jahr 2014

# Data Science Projekt: Widerstände überwinden



# Data Science Projekt: Widerstände überwinden

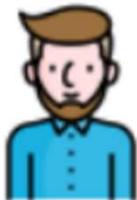
Wahrgenommene  
Steuerungskompetenz  
= Kontrolle



# DATA SCIENCE ROLLEN

---

## DATA SCIENTIST



- Abstimmung mit Domänenexperten
- Entwicklung von Use Cases
- Explorative Datenanalyse
- Machine Learning

## DATA ENGINEER



- Aufbau und Pflege von Daten-Infrastruktur und Cloud-Lösungen
- Machine Learning Model Deployment

## DATA ANALYST



- Identifikation und deskriptive Analyse von strukturierten Daten
- Visualisierung und Dashboarding

## BUSINESS STAKEHOLDER



- Definition und Priorisierung von Anwendungsfällen
- Evaluierung des Erfolgs von Data Science Projekten

[www.kobold.ai](http://www.kobold.ai)

<https://www.kobold.ai/data-scientist/>

<https://www.kobold.ai/data-engineer/>

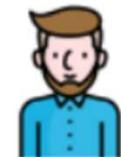
# Data Science: Rollen

## Anforderungen an den idealen Data Scientist:

- ein fortgeschrittenes Wissen in Statistik und wie Daten effektiv visualisiert werden
- Beherrschen von mehreren Programmiersprachen, die für Data Science wichtig sind
- Business Knowledge
- sehr gute Kommunikations- und Präsentationsfähigkeiten

➤ **Es ist schwer in allen dieser Bereiche gut zu sein, aber Teammitglieder können sich gegenseitig ergänzen**

**DATA  
SCIENTIST**



- Abstimmung mit Domänenexperten
- Entwicklung von Use Cases
- Explorative Datenanalyse
- Machine Learning

# Data Science: Rolle - Data Engineer

- Berufsfeld des Data Engineers ist relativ jung.
- Ähnlich wie Chemiker früher auch Labortechniker waren, sind **Data Engineers die Labortechniker in der Data Science.**
- Aufgaben eines Data Engineers:
  - **Bereitstellung von Data Pipelines.**
  - **Bereitstellung von Daten** aus einem Data Warehouse für Modelle.
  - **Vereinheitlichung von Datenstrukturen.**
  - **Überwachung erstellter Anwendungen.**
  - **Behebung von Ausfällen** und Wiederinbetriebnahme der Anwendungen.

**DATA  
ENGINEER**



- Aufbau und Pflege von Daten-Infrastruktur und Cloud-Lösungen
- Machine Learning Model Deployment

# Data Science Projekt: Rolle - Business Intelligence & Data Analyst

- Wird oft mit einem Data Scientist verwechselt.
- Verfügt über **erweiterte Fähigkeiten** für den **Umgang mit einem Data Warehouse**.
- Zieht **Erkenntnisse aus Daten**, um Verbesserungsmöglichkeiten in einer Firma aufzuzeigen, jedoch meist ohne Data-Science-Modelle.
- Grenzen zum Data Scientist sind fließend.
- **Präsentiert häufiger Erkenntnisse**.
- **Nutzt BI-Tools** wie **Tableau** zur Visualisierung.
- Programmierung ist weniger umfangreich als bei Data Scientists.
- Prognosemodelle werden eher von Data Scientists erstellt.

**DATA  
ANALYST**



- Identifikation und deskriptive Analyse von strukturierten Daten
- Visualisierung und Dashboarding

# Data Science Projekt: Rolle – Business Stakeholder & Fachexperte

- Data Scientists müssen sich in die Zusammenhänge einer Aufgabe einarbeiten, der Fachexperte (Subject Matter Expert) stellt dem Data Scientist notwendige Informationen zur Verfügung.
- Idealerweise **arbeitet der Data Scientist eng mit den relevanten Mitarbeitern** aus der Fachabteilung zusammen.
- **Frühzeitige Einbeziehung** des Fachexperten **für Prototypen und Feedback** hilfreich.
- **Direkte Zusammenarbeit ist nicht immer gern gesehen**, da Kollegen in den Fachabteilungen oft ausgelastet sind.
- **Traditionelle Erwartung**: Anforderungen werden übergeben, das Entwicklungsteam liefert das Endergebnis.
- Moderne Softwareentwicklung erfordert **Commitment zur ausreichenden Einbeziehung von Fachexperten** von Beginn an.

## BUSINESS STAKEHOLDER



- Definition und Priorisierung von Anwendungsfällen
- Evaluierung des Erfolgs von Data Science Projekten

# Data Science Projekt: Rolle – Citizen Data Scientist

- Citizen Data Scientist ist eine neue, noch nicht etablierte Rolle, die **mehr als einfache Datenanalyse durchführt, aber kein vollwertiger Data Scientist** ist.
- Citizen Data Scientists sind nützlich für Aufgaben wie
  - **Datenbeschaffung,**
  - **Datenreinigung,**
  - **explorative Datenanalysen und**
  - **Erstellen von Dashboards und Reports.**
- übernehmen die **Vorarbeit bei datenbezogenen Projekten** und ziehen Data Scientists nur bei komplexeren Problemen hinzu.
- **Citizen Data Scientists sind oft Fachexperten**, wodurch kein Informationsverlust bei der Übertragung geschäftlicher Anforderungen in das Data-Science-Vokabular entsteht.
- **ermöglicht Unternehmen, effizienter zu arbeiten**, indem nicht für jedes datenbezogene Projekt ein Data Scientist benötigt wird.

# Data Science Projektmanagement

Ein Projekt ist ein zeitlich begrenztes Vorhaben mit dem Ziel, ein einmaliges Produkt, eine einmalige Dienstleistung oder ein einmaliges Ergebnis zu schaffen.

Sprichwort aus dem Projektmanagement:

**If you fail to plan, you plan to fail.**

# Data Science Aufgaben Projektmanager

- **Verantwortlich für die Erreichung der Zielvorgaben des Projekts**
- Nicht der direkte Vorgesetzte der Teammitglieder, aber verantwortlich für die Teamproduktion
- **Aufgaben:**
  - **Erstellung und Überwachung von Zeit- und Budgetplänen**
  - **Umfangsmanagement**
  - **Kommunikation mit dem Sponsor, Stakeholdern und Teammitgliedern**
  - **Koordination und Ausgleich widersprüchlicher Stakeholder-Ziele**
- Muss kein Data Scientist sein, aber ausreichendes Wissen und Verständnis haben
- **Sicherstellung ausreichender Ressourcen** für das Projekt
- Festlegen und **Überprüfen der Einhaltung von Qualitätskriterien**
- Wesentlicher Faktor für den Erfolg eines Data-Science-Projekts

# Projektmanagement & Data Science Prozess

- |                                     |   |
|-------------------------------------|---|
| <b>1. Initiierung</b>               | <b>1. Business Understanding &amp; Data Understanding</b> |
| <b>2. Planung</b>                   | <b>2. Data Preparation</b>                                |
| <b>3. Ausführung</b>                | <b>3. Modeling</b>  |
| <b>4. Überwachung und Steuerung</b> | <b>4. Evaluation</b>                                      |
| <b>5. Abschluss</b>                 | <b>5. Deployment</b>                                      |

# Projektmanagement & Data Science Prozess: KI Tutor-Diskussion

## t1p.de/ds-pm1

Free Account 10 Messages free (evtl. mit 2 Accounts arbeiten)

Paid Account (23€ / Monat / monatlich kündbar) 40-80 Messages per Day inkludiert

Weitere Alternative Bing Chat (6 Messages pro Chat free)

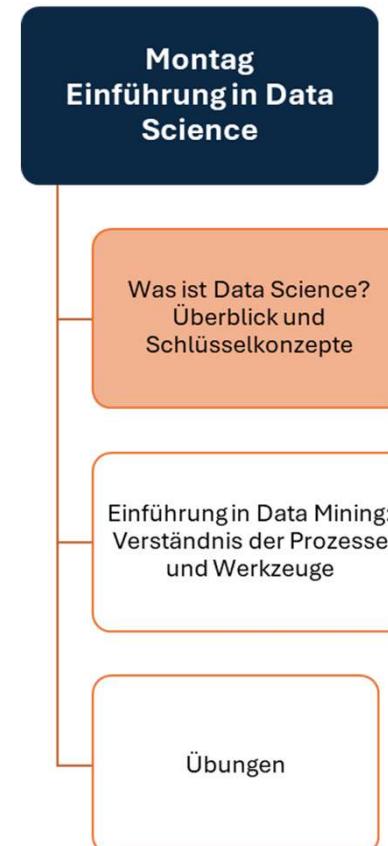
Emojis	Win + Punkt
Rückgängig	Strg + Z
Wiederholen	Strg + Umschalttaste + Z
Ausschneiden	Strg + X
Kopieren	Strg + C
Einfügen	Strg + V
Als unformatierten Text einfügen	Strg + Umschalttaste + V
Alles auswählen	Strg + A
Rechtschreibprüfung	>
Schreibrichtung	>
Im Lesemodus öffnen <span style="background-color: #e0f2e0;">Neu</span>	
MarkDownload - Markdown Web Clipper	>
Untersuchen	

Wer will von seinen Ergebnissen berichten ?

# Was ist Data Science: Überblick und Schlüsselkonzepte

## Zusammenfassung

- Data Science Prozess
- Widerstände überwinden
- Data Science Rollen
- Data Science Projektmanagement
- Tutor Projektmanagement



# Einführung in Data Mining: Verständnis der Prozesse und Werkzeuge

## Zusammenfassung

- Data Scientist Tools
- Von Rohdaten zu Erkenntnissen
- Data Mining Methoden
- Data Mining Tools

Montag  
Einführung in Data Science

Was ist Data Science?  
Überblick und Schlüsselkonzepte

Einführung in Data Mining:  
Verständnis der Prozesse und Werkzeuge

Übungen

# DATA SCIENTIST

## TOOLS

### Data Acquisition & Engineering



Informatica



SQL



Talend



Pentaho

### Analytics & Machine Learning



python



R



RapidMiner

### Visualisierung & Deployment



PowerBI



Tableau



Java



Docker

### Cloud Technologie



Microsoft Azure



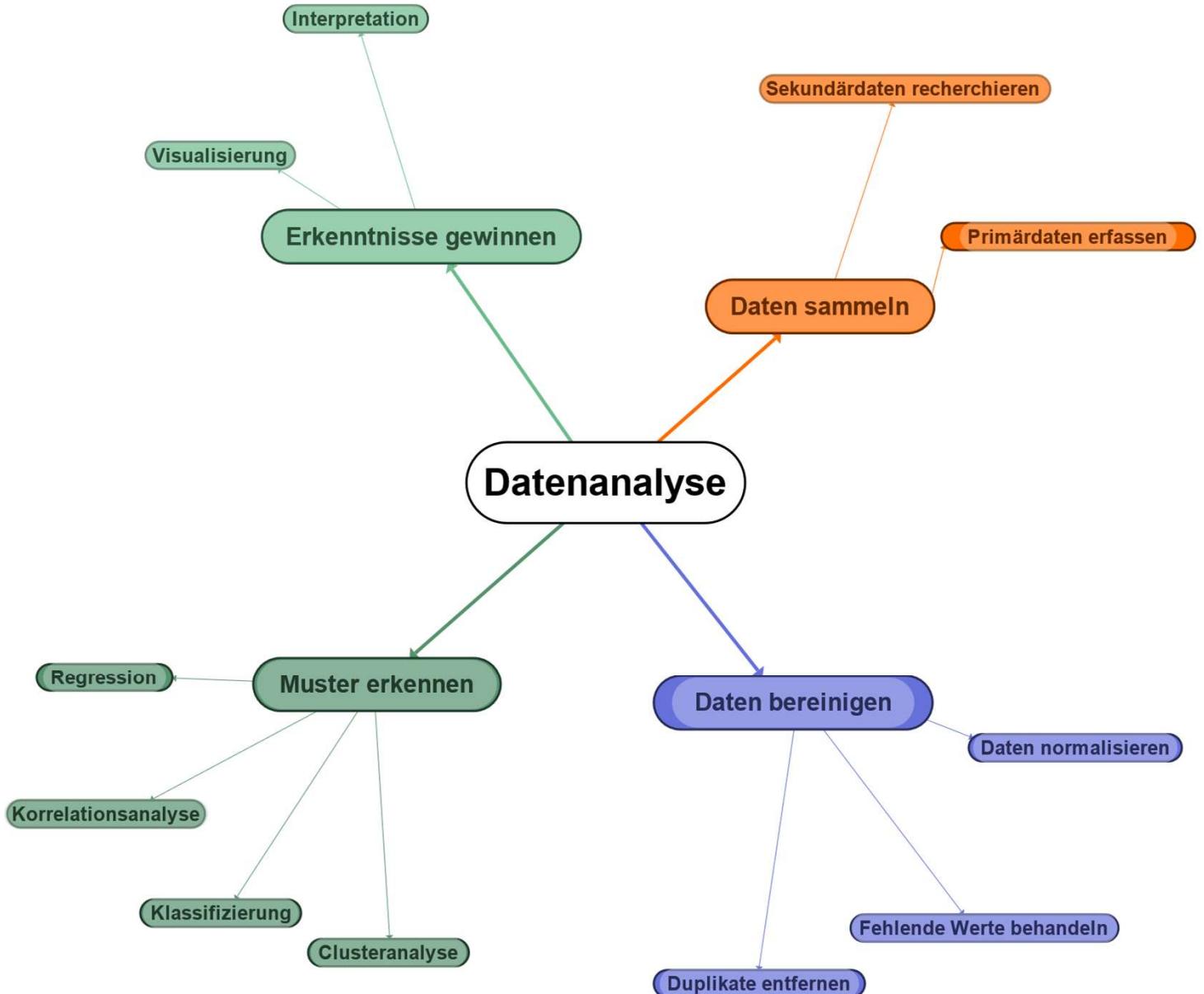
Google Cloud



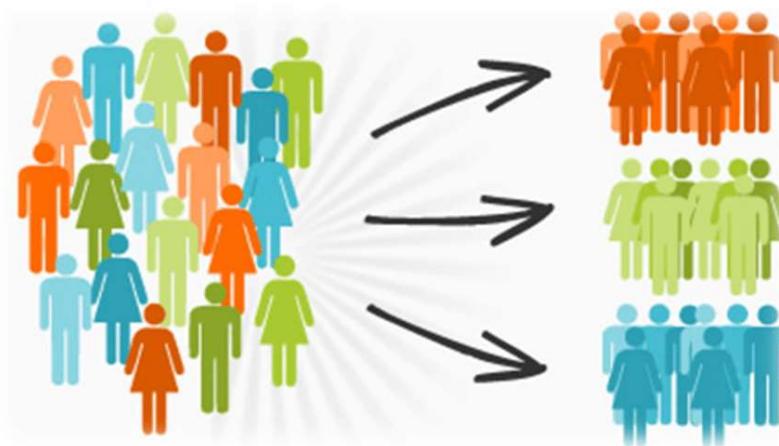
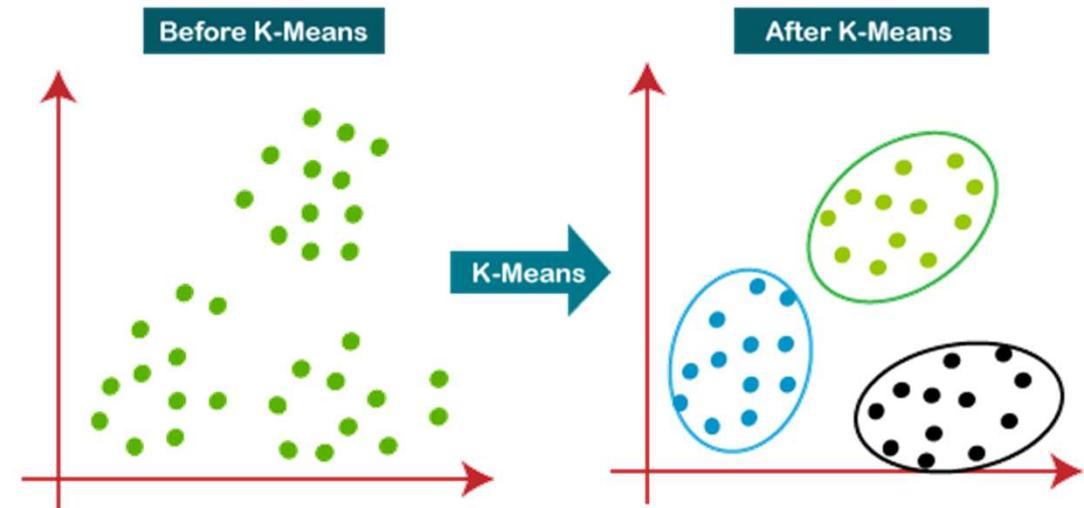
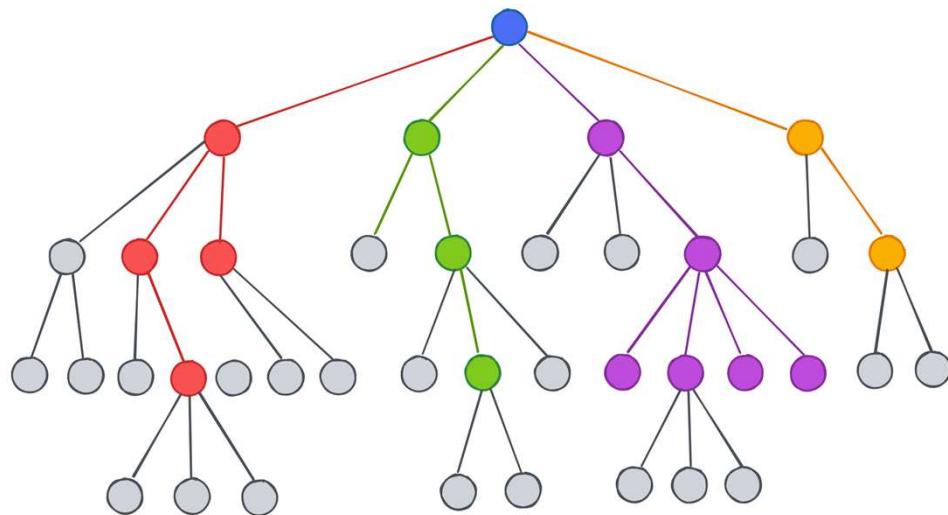
Amazon Web Services

[www.kobold.ai](http://www.kobold.ai)

## Data Mining: Von Rohdaten zu Erkenntnissen



# Data Mining: Methoden und Techniken



# DATA MINING

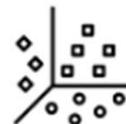
## METHODEN

### STATISTIK



Häufigkeiten, Relationen und Interaktionen erkennen.

### CLUSTERING



Identifizieren von ähnlichen Einträgen und deren Relation.

### ASSOZIATIONEN



Quantifizieren von zusammen auftretenden Ereignissen.

### OUTLIERS



Detektion von ungewöhnlichen Datensätzen.

### KLASSIFIKATION



Die Vorhersage von Gruppenzugehörigkeiten.

### REGRESSION



Die Vorhersage von numerischen Werten.

[www.kobold.ai](http://www.kobold.ai)



# Data Mining: Werkzeuge & Tools

# Data Mining Tools

## Analyse



python



R



KNIME

## GUI-basiert



RapidMiner



SPSS

## Visualisierung



PowerBI



Tableau



Google  
Data Studio

## Anderes



Excel

# Einführung in Data Mining: Verständnis der Prozesse und Werkzeuge

## Zusammenfassung

- Data Scientist Tools
- Von Rohdaten zu Erkenntnissen
- Data Mining Methoden
- Data Mining Tools

Montag  
Einführung in Data Science

Was ist Data Science?  
Überblick und Schlüsselkonzepte

Einführung in Data Mining:  
Verständnis der Prozesse und Werkzeuge

Übungen

# Übung

## Einführung in KNIME Interface

### Zusammenfassung

- Interface
- Nodes and Workflows
- KNIME Community Hub

Montag  
Einführung in Data Science

Was ist Data Science?  
Überblick und Schlüsselkonzepte

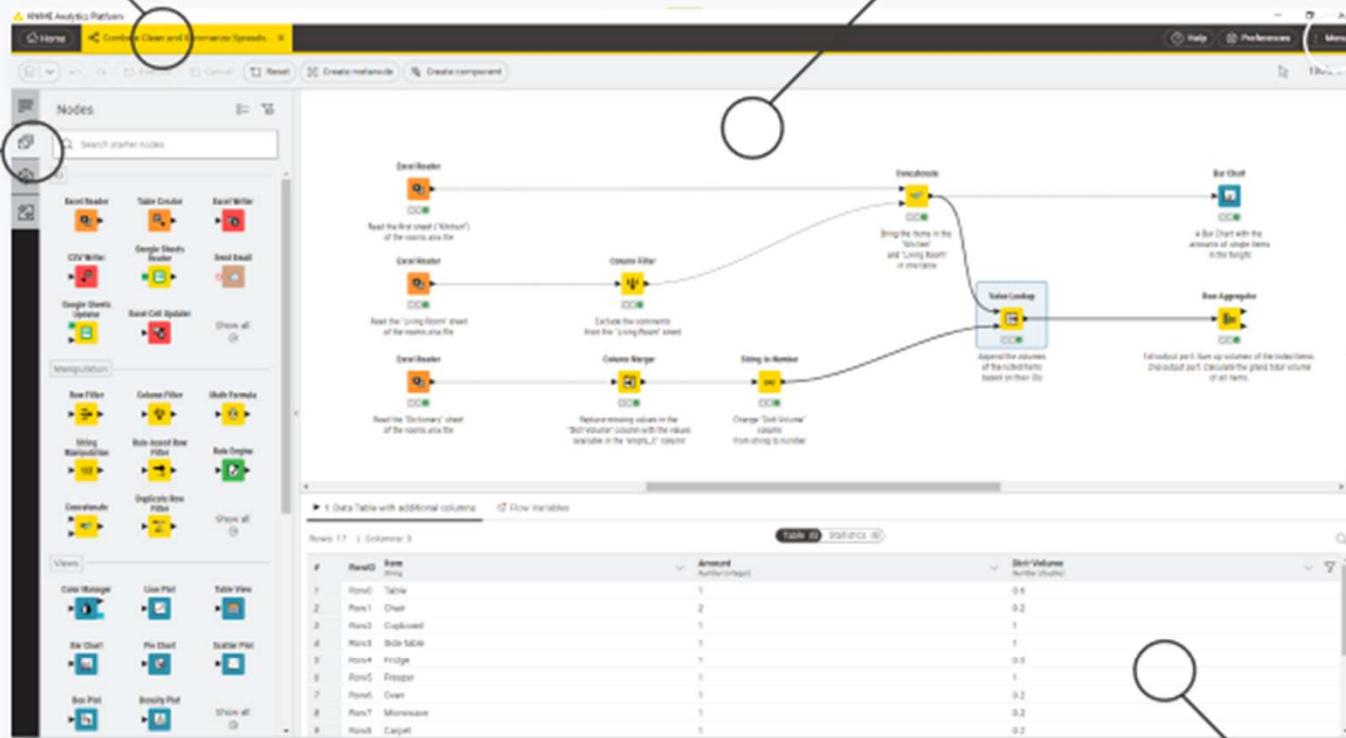
Einführung in Data Mining:  
Verständnis der Prozesse und Werkzeuge

Übungen

# Praktische Einführung in KNIME

## Application tabs

Entry page tab and all opened workflows tabs.



## Workflow Editor

Canvas for editing the currently active workflow.

## Help, Preferences, Menu

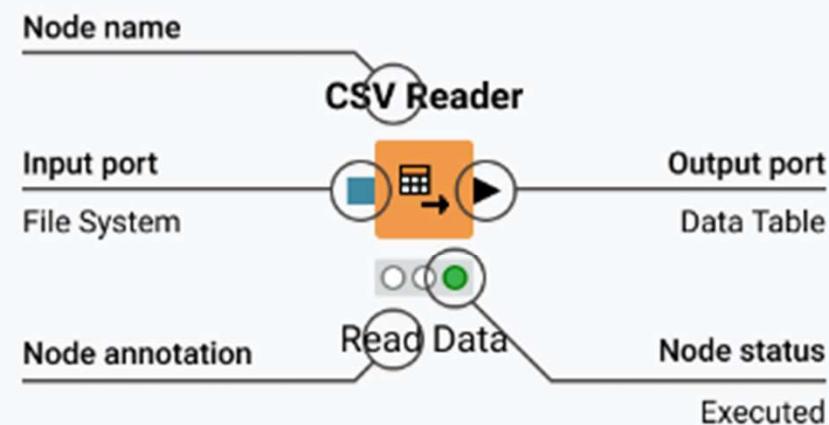
Access to more material, install additional extensions and change settings for the node repository.

## Node Monitor

Shows the output of the current selected node and also the flow variable values.

# Praktische Einführung in KNIME

## Nodes and Workflows



# Praktische Einführung in KNIME

## What is the node status?

A node can be in four different states. The node status is shown by a traffic light below each node.



### Not configured

The node is waiting for configuration or incoming data.



### Configured

The node has been configured correctly, and can be executed.



### Executed

The node has been successfully executed. Results may be viewed and used in downstream nodes.



### Error

The node has encountered an error during execution.

KNIME Analytics Platform

KNIME Getting Started Furniture 1 X

Home Execute Cancel Reset Create metanode Create component Help Preferences Menu 154% ▾

**Excel Reader**

This node reads Excel files (xlsx, xlsxm, xlsb, and xls format). It can read a single or multiple files at the same time, however reading only one sheet per file. The supported Excel types that can be read in are string, number, boolean, date, and time but not pictures, diagrams, etc.

The data is read in and converted to the KNIME types string, integer, long, double, boolean, local date, local time, and local date&time. Upon execution, the node will scan the input file to determine number and types of the columns and output a table with the auto-guessed structure and KNIME types.

Formulas can also be read in and reevaluated if desired. The performance of this node is limited (due to the underlying library of the Apache POI project).

Reading large files takes a long time and uses a lot of memory (especially files in xlsx format when formula reevaluation is enabled).

This dialog of this node shows a preview and the file content. While the preview shows the table that will be read in having the settings applied and is updated once a setting is changed, the file content shows the content of the file as it is displayed in Excel. This allows finding row numbers and column names easier to specify different settings.

This node can access a variety of different file systems. More information about file handling in KNIME can be found in the official File Handling Guide.

**Ports Options Views**

**Output ports**

- Type: File Table
- File being read with number and types of columns guessed automatically.

**File System Connection (Dynamic Import)**

The file system connection.

- Type: File System

**Extension**

KNIME Excel Support

provided by KNIME AG, Zurich, Switzerland

```

graph LR
    ER1[Excel Reader] --> CF1[Column Filter]
    ER1 --> C1[Concatenate]
    CF1 --> C1
    C1 --> BC[Bar Chart]
    C1 --> VLU[Value Lookup]
    C1 --> RA[Row Aggregator]
    VLU --> RA
    RA --> BC
    ER2[Excel Reader] --> CM[Column Merger]
    CM --> STN[String to Number]
    STN --> RA
  
```

**Data Preview**

1: File Table

#	RowID	Item	Amount
1	Row0	Table	1
2	Row1	Chair	2
3	Row2	Cupboard	1
4	Row3	Side table	1
5	Row4	Fridge	1
6	Row5	Freezer	1
7	Row6	Oven	1
8	Row7	Microwave	1
9	Row8	Carpet	1

KNIME Analytics Platform

KNIME Getting Started Furniture 1 X

Home Execute Cancel Reset Create metanode Create component Help Preferences Menu

154% ▾

Nodes

Search starter nodes

IO

- Excel Reader Excel Writer Microsoft Authenticator
- Google Authenticator Google Sheets Reader Google Sheets Writer
- CSV Reader CSV Writer Show all

Manipulation

- Row Filter Column Filter Concatenate
- Value Lookup Row Aggregator Table Splitter
- String Cleaner Table Cropper Show all

Views

- Bar Chart Line Plot Pie Chart
- Stacked Area Chart Scatter Plot Statistics View
- Heatmap Histogram Show all

Other Data Types

- String to Date&Time Extract Date&Time Fields Date&Time-based Row Filter
- Date&Time to String Date&Time Difference Date&Time Shift
- Modify Time Moving Aggregator Show all

Reporting

Bar Chart

```

graph LR
    ER1[Excel Reader] --> CF1[Column Filter]
    ER2[Excel Reader] --> CF1
    CF1 --> VL1[Value Lookup]
    VL1 --> RA1[Row Aggregator]
    RA1 --> BC[Bar Chart]
    
```

Concatenate

Add comment Excel Reader

Column Filter

Value Lookup Row Aggregator

Excel Reader Column Merger String to Number

String to Number

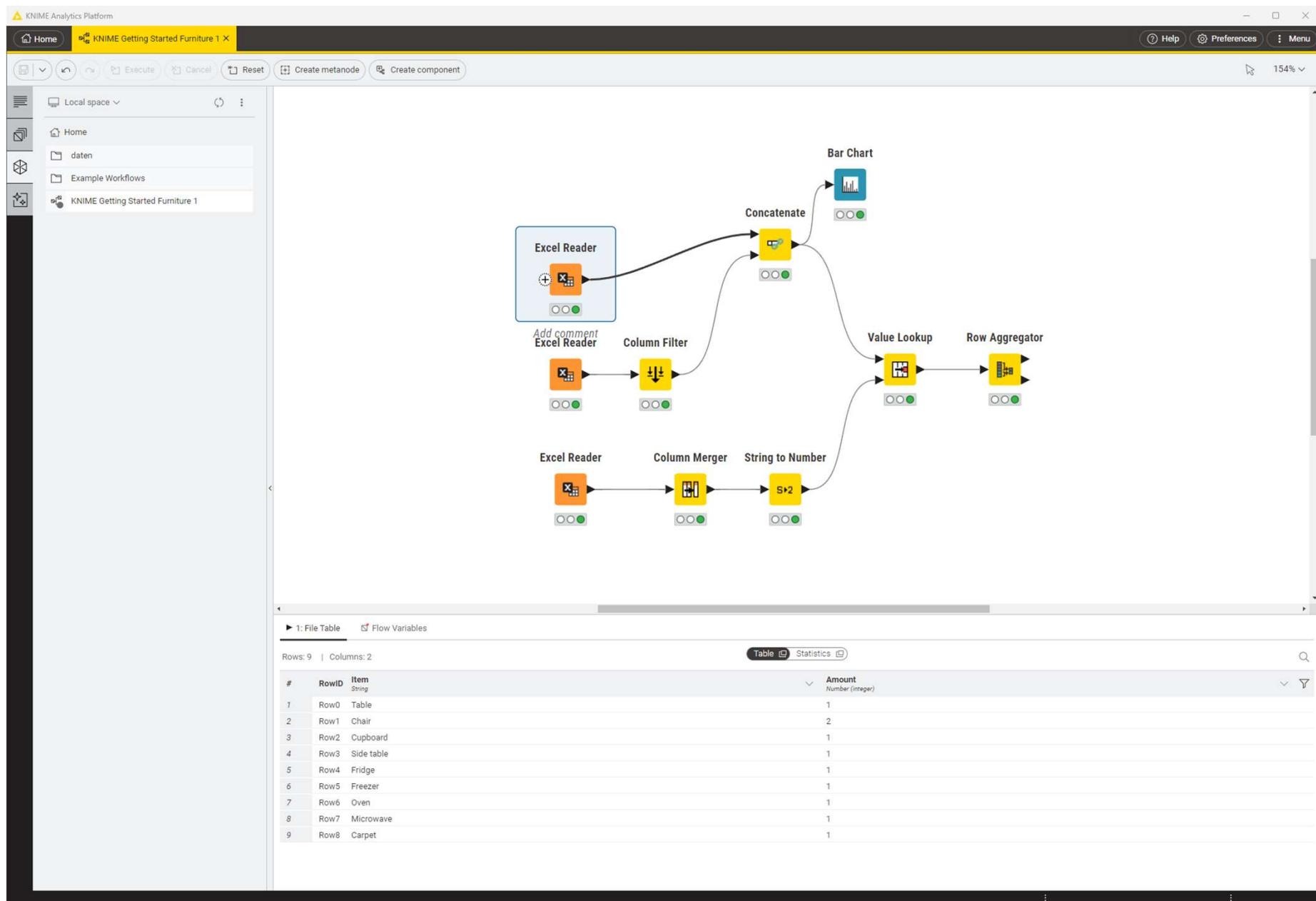
File Table

Flow Variables

Table Statistics

Rows: 9 | Columns: 2

#	RowID	Item	Amount
1	Row0	Table	1
2	Row1	Chair	2
3	Row2	Cupboard	1
4	Row3	Side table	1
5	Row4	Fridge	1
6	Row5	Freezer	1
7	Row6	Oven	1
8	Row7	Microwave	1
9	Row8	Carpet	1



KNIME Analytics Platform

KNIME Getting Started Furniture 1

Home Execute Cancel Reset Create metanode Create component Help Preferences Menu

154% ▾

**KNIME AI Assistant**

Hi! I am K-AI, your KNIME Q&A Assistant. What would you like to know?

was macht der Excel Reader?

Der "Excel Reader" Knoten in KNIME ermöglicht das Lesen von Daten aus Excel-Dateien (.xls und .xlsx) und das Einlesen dieser Daten in KNIME-Tabellen zur weiteren Verarbeitung.

**Nodes**

Excel Reader

wie funktioniert der Row Aggregator

Der "Row Aggregator" Knoten in KNIME ermöglicht das Aggregieren von Spalten mithilfe einer Aggregationsfunktion. Die Zellen können optional nach einer Kategorie-Spalte gruppiert werden. Einige Aggregationstypen unterstützen die Verwendung eines Gewichts aus einer Gewichtsspalte. Wenn Sie eine komplexere Funktionalität benötigen, können Sie den "GroupBy" Knoten verwenden.

**Nodes**

Row Aggregator GroupBy DB GroupBy

DB Pivot

```

graph LR
    ER1[Excel Reader] --> CF1[Column Filter]
    ER2[Excel Reader] --> CM[Column Merger]
    CF1 --> Concat[Concatenate]
    CM --> S2N[String to Number]
    S2N --> VL[Value Lookup]
    VL --> RA[Row Aggregator]
    Concat --> RA
    RA --> BC[Bar Chart]
  
```

1: File Table Flow Variables

Table Statistics

Rows: 9 | Columns: 2

#	RowID	Item	Amount
1	Row0	Table	1
2	Row1	Chair	2
3	Row2	Cupboard	1
4	Row3	Side table	1
5	Row4	Fridge	1
6	Row5	Freezer	1
7	Row6	Oven	1
8	Row7	Microwave	1
9	Row8	Carpet	1

ChatGPT - AutoExpert (Chat) | Getting Started Guide | KNIME | KNIME Community Hub | knime/Beginners Space – KNIM | +

hub.knime.com/?pk\_vid=aae8cb3c0669311b17182902295c59b3

Open for Innovation **KNIME** Hub Pricing About : Sign in

# Welcome to KNIME Community Hub

Access 22953 workflows, 2092 components and 244 extensions provided by KNIME and our community.

Search

...or browse curated collections of workflows and resources

**Starter collection**

Discover beginner-friendly examples, familiarize yourself with essential nodes, and integrate...

knime

**KNIME for Generative AI**

Explore this collection of ready to use workflows to get started with using Large Language Models (LLMs)....

knime

**First Steps with KNIME Reporting**

You can use the Reporting extension to automatically generate and distribute static reports for regular...

knime

Show more collections

ChatGPT - AutoExpert (Chat) × Getting Started Guide | KNIME × knime/Beginners Space – KNIME × +

knime.com/getting-started-guide

# Getting Started Guide

for KNIME Analytics Platform

## Table of Contents

- [What is KNIME Analytics Platform?](#)
- [Getting set up with KNIME Analytics Platform](#)
- [Build your first workflow](#)
- [Nodes and workflows](#)
- [More examples](#)
- [Extensions and integrations](#)
- [KNIME Forum](#)

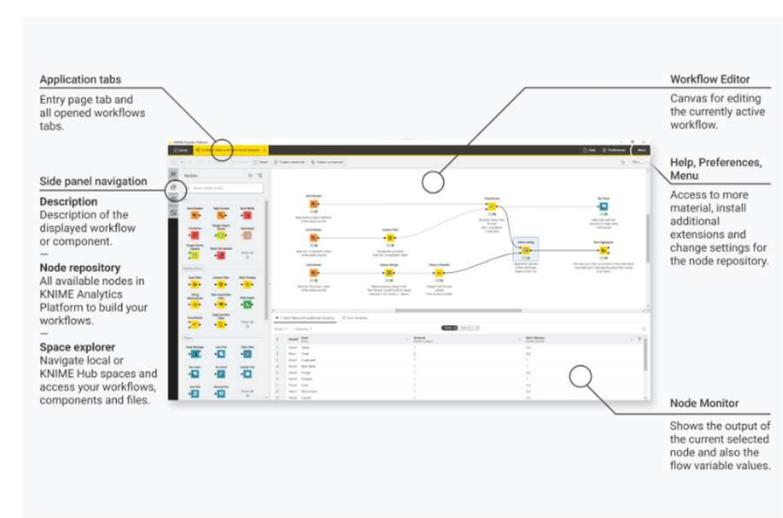
## What is KNIME Analytics Platform?

KNIME Analytics Platform is an open source software with an intuitive, visual interface that lets you build analyses of any complexity level - from automating spreadsheets to ETL to machine learning.

A more detailed explanation of the different views in the KNIME Workbench is provided in the [KNIME Analytics Platform User Guide](#).

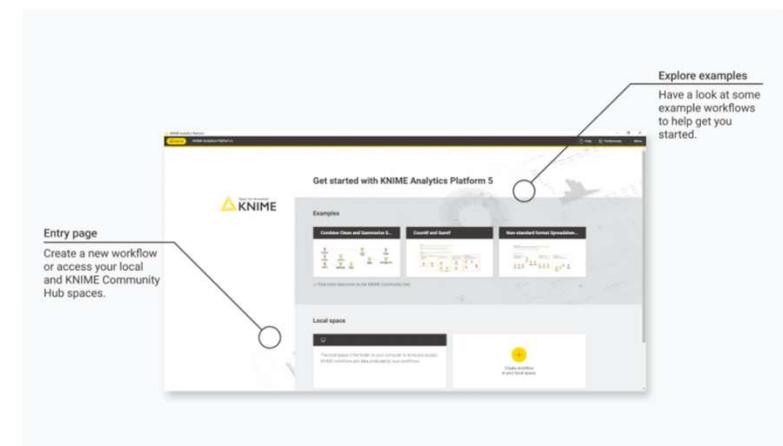
[Return to top](#)

## Getting Set Up with KNIME Analytics Platform



If you haven't downloaded KNIME Analytics Platform yet, you can do so [here](#).

Once you have installed KNIME Analytics Platform, you can start analyzing your data right away. The entry page is the first thing you will see. Here you can access three example workflows to get started, or, if you're following along with this guide, create your first workflow from scratch.



# Übung

## Einführung in KNIME Interface

### Zusammenfassung

- Interface
- Nodes and Workflows
- KNIME Community Hub

Montag  
Einführung in Data Science

Was ist Data Science?  
Überblick und Schlüsselkonzepte

Einführung in Data Mining:  
Verständnis der Prozesse und Werkzeuge

Übungen

# Übung

## Einführung in KNIME Workflows

### Zusammenfassung

- Getting Started Guide
- erster eigener Workflow
  - Reader Node
  - Column Filter, Merger
  - Bar Chart

Montag  
Einführung in Data Science



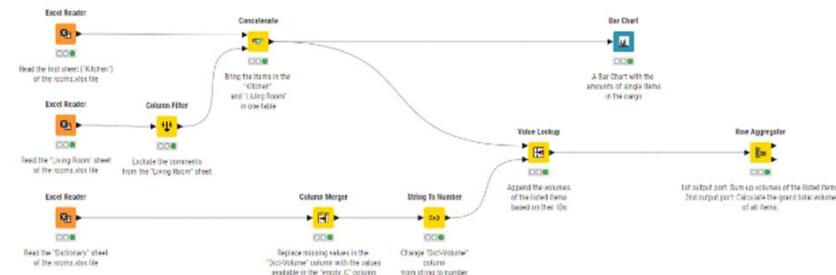
# Übung 1: Erster eigener Workflow

<https://www.knime.com/getting-started-guide>

## Build your First Workflow

You can follow along with this guide by either downloading the workflow and reading the guide to better understand it, or, attempting to build the workflow on your own. Either way, you can access the workflow at any time [on KNIME Community Hub](#).

Now let's say that you have some data that you want to process, analyze and visualize. With the following example workflow, you will read, combine, clean, and summarize data from multiple Excel sheets. Then you will calculate the total volume of a cargo that carries furniture from one house to another.



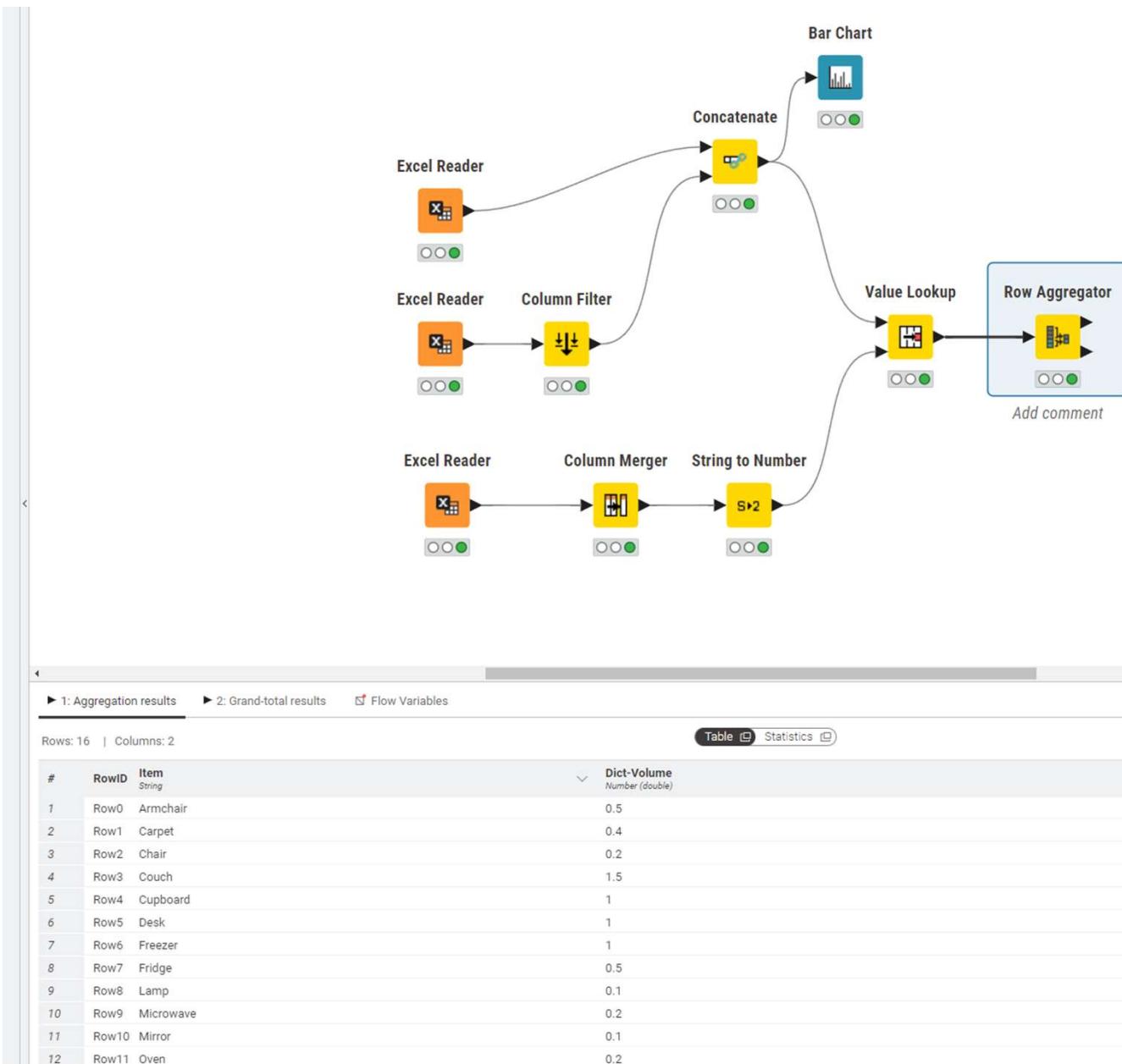
### Step 1/6 – Download data and create new workflow

To get started, first [download the xls file](#) that contains the data that you are going to use in the workflow. Open your KNIME Analytics Platform and create a new, empty workflow by clicking the yellow "plus" button on the entry page.

### Step 2/6 - Drag and drop XLSX file into workbench editor

From the download folder, drag and drop the xlsx file into the workflow editor. An Excel Reader node will appear on the canvas. The node is already configured with the default settings. To open the configuration dialog of the node and inspect the settings, click the configuration cog button in the node action bar.

## Lösung



# Übung

## Einführung in KNIME Workflows

### Zusammenfassung

- Getting Started Guide
- erster eigener Workflow
  - Reader Node
  - Column Filter, Merger
  - Bar Chart

Montag  
Einführung in Data Science



# Übung

## KNIME Data Exploration

### Zusammenfassung

- Datensatz Schwertlilie Data  
Exploration

Dienstag  
**Datenanalyse in der  
Praxis**

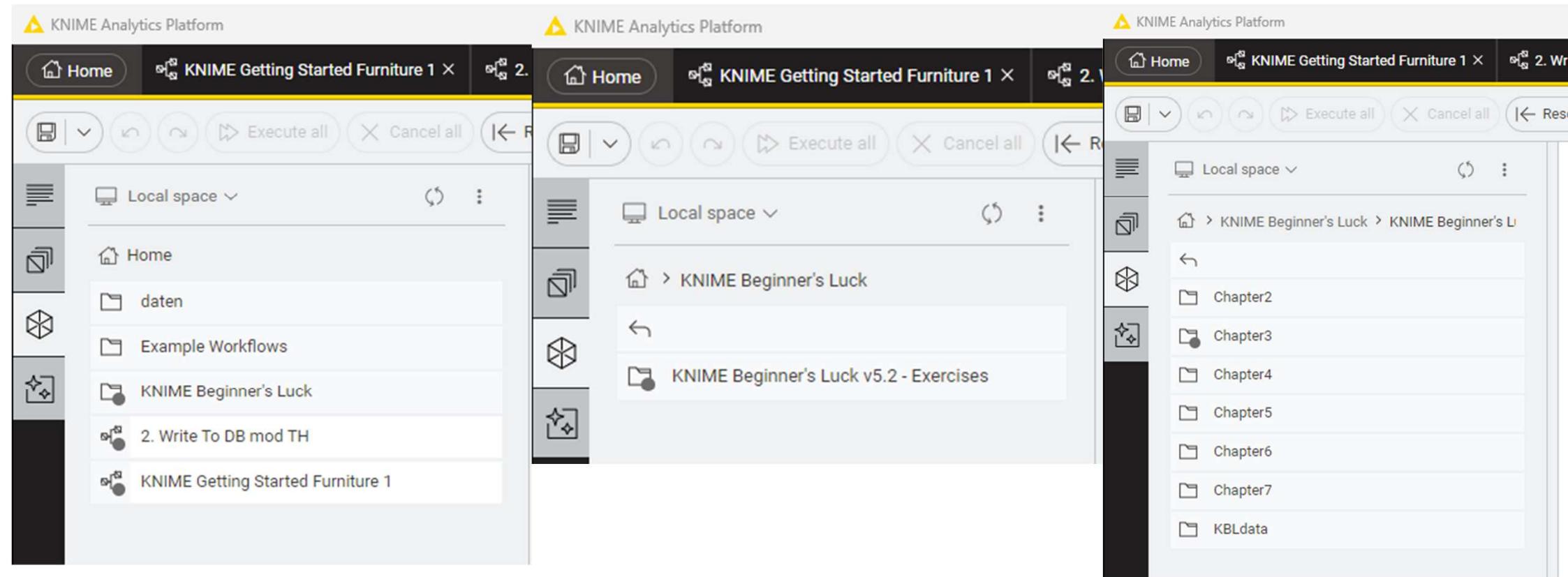
Praktische Einführung in  
die Datenanalyse:  
Methoden der  
explorativen  
Datenanalyse

Anwendungsfälle aus  
verschiedenen  
Branchen: Wie  
Datenanalyse  
branchenübergreifend  
genutzt wird?

Übungen

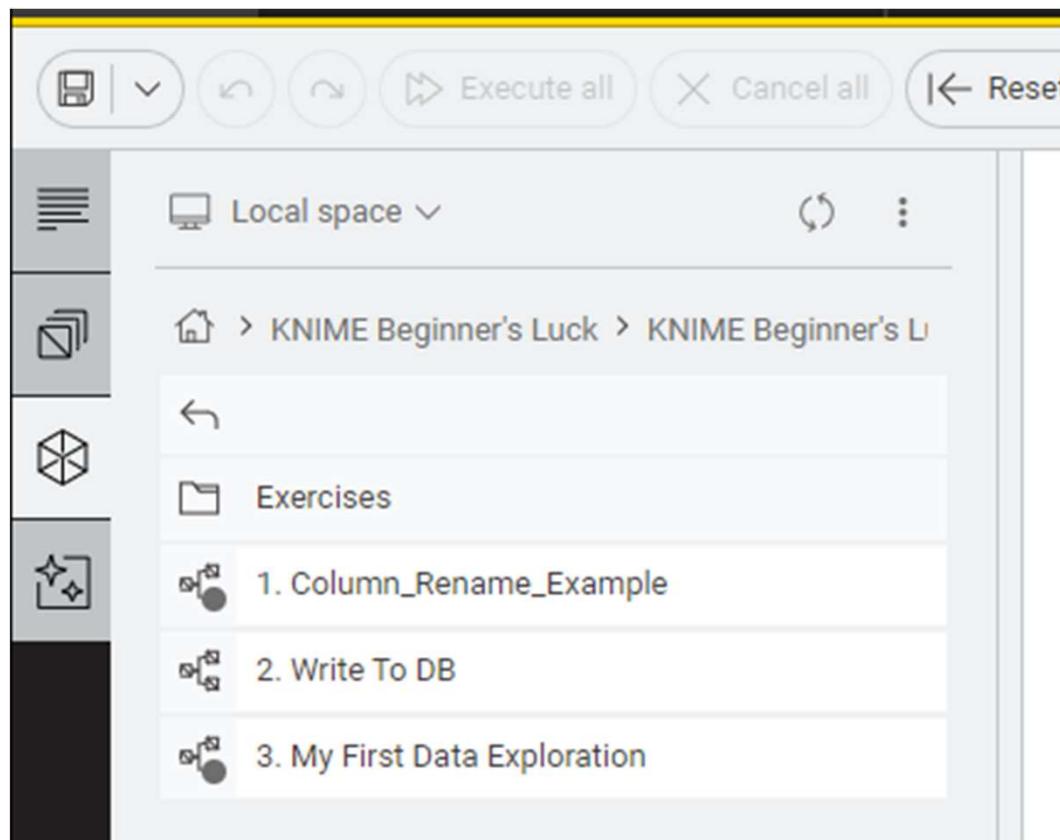
# Übung 2: Data Exploration

- Buch KNIME Beginner's Luck Kapitel 3 Seite 62-63 und 99-114,
- Workflow unter Chapter 3
  - vorher KNIME Beginner's Luck.knar als Workflow in KNIME importieren



# Übung 2: Data Exploration

- Buch KNIME Beginner's Luck Kapitel 3 Seite 62-63 und 99-114,
- Workflow unter Chapter 3, My First Data Exploration
  - vorher KNIME Beginner's Luck.knar als Workflow in KNIME importieren



# Data Exploration

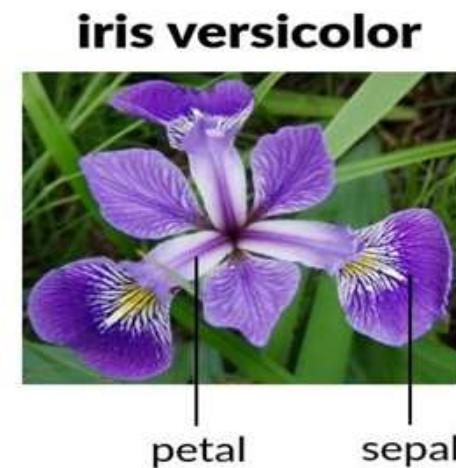
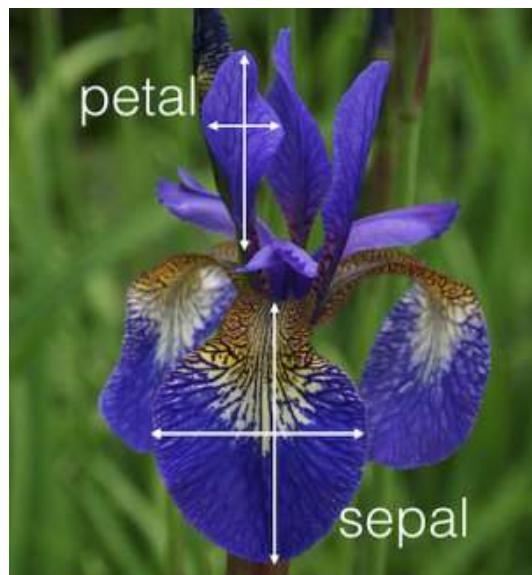
## Iris Flower Data Set (Schwertlilien)

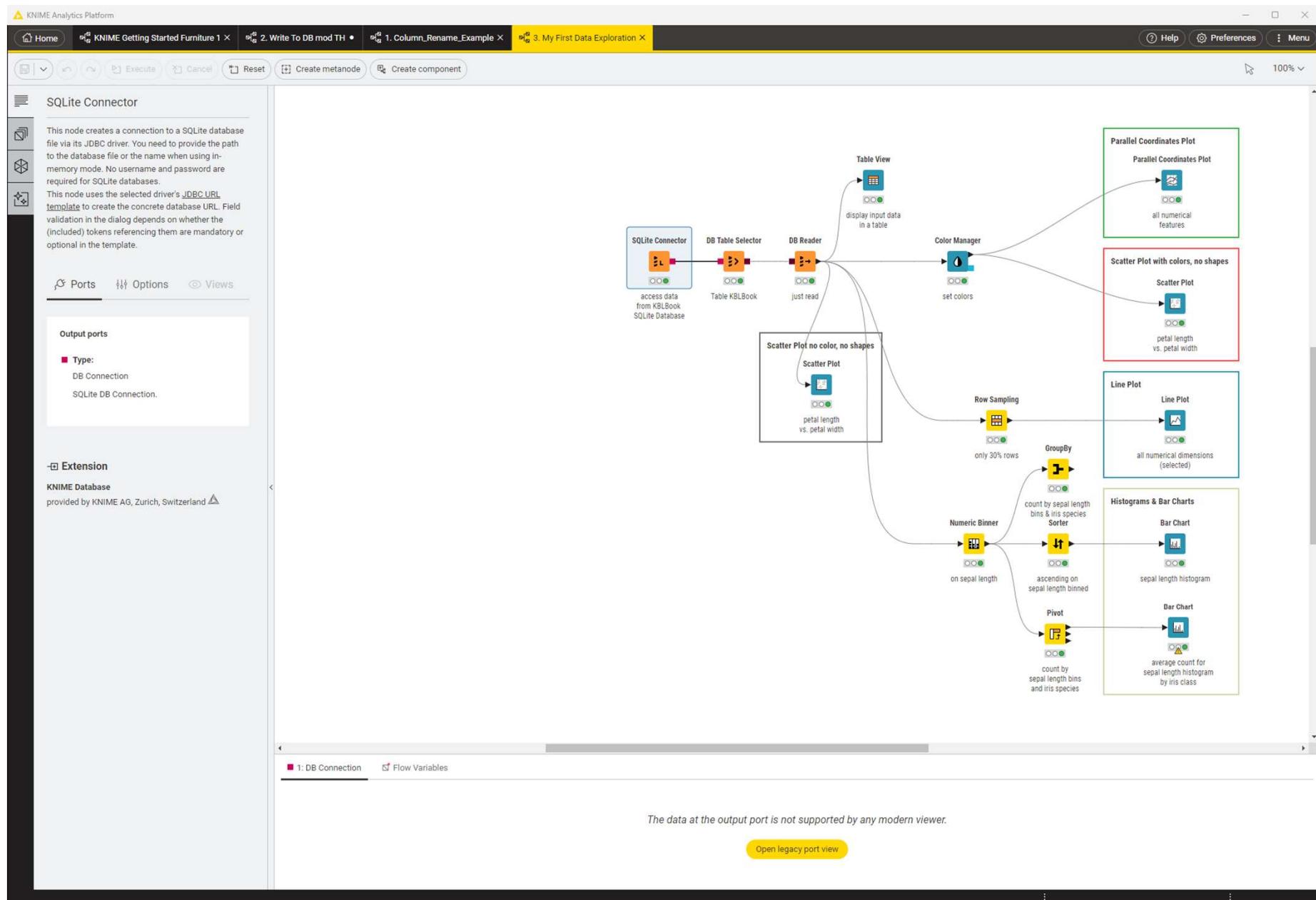
Das Iris Flower Data Set ist ein bekannter Datensatz in der Welt der maschinellen Lern- und Statistikforschung. Es wurde erstmals von dem britischen Biologen und Statistiker Ronald A. Fisher im Jahr 1936 eingeführt. Fisher sammelte die Daten durch Vermessung von Blütenmerkmalen von drei verschiedenen Arten der Iris-Blume: Setosa, Versicolor und Virginica. Dieser Datensatz diente ursprünglich dazu, die Anwendbarkeit statistischer Methoden zur Klassifizierung und Identifikation von Pflanzenarten zu demonstrieren.

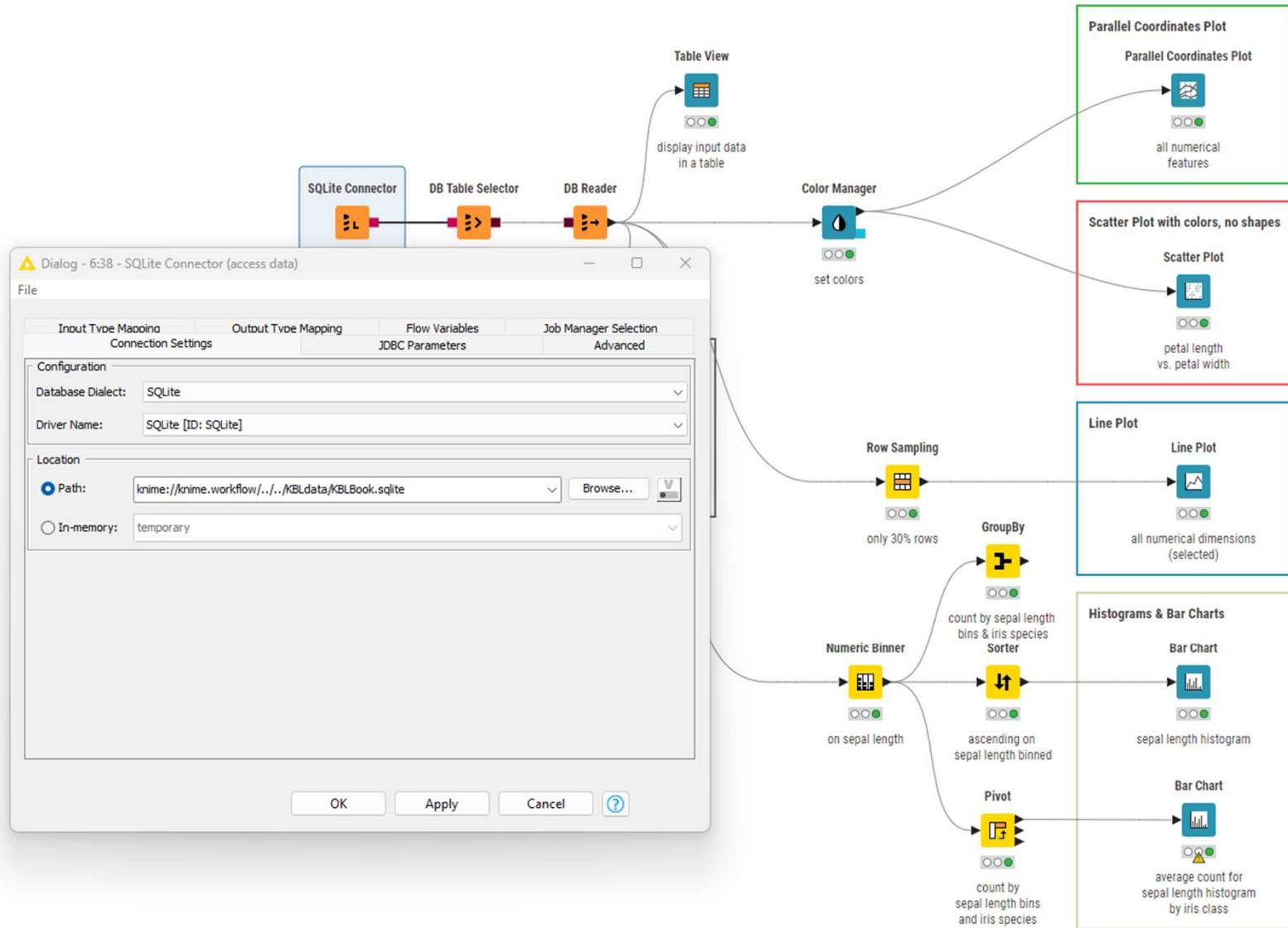
Aufgrund seiner klaren Struktur und der gut definierten Klassifikationsaufgabe wird das Iris Flower Data Set oft als Einstiegspunkt für Anfänger in maschinellem Lernen verwendet. Forscher und Entwickler nutzen ihn als Benchmark, um die Leistung verschiedener Klassifikationsalgorithmen zu vergleichen. Der Datensatz hat somit einen bedeutenden Beitrag zur Entwicklung und Evaluation von Klassifikationsalgorithmen geleistet und ist nach wie vor ein beliebtes Instrument in der maschinellen Lerngemeinschaft.

## Dataset Übersicht

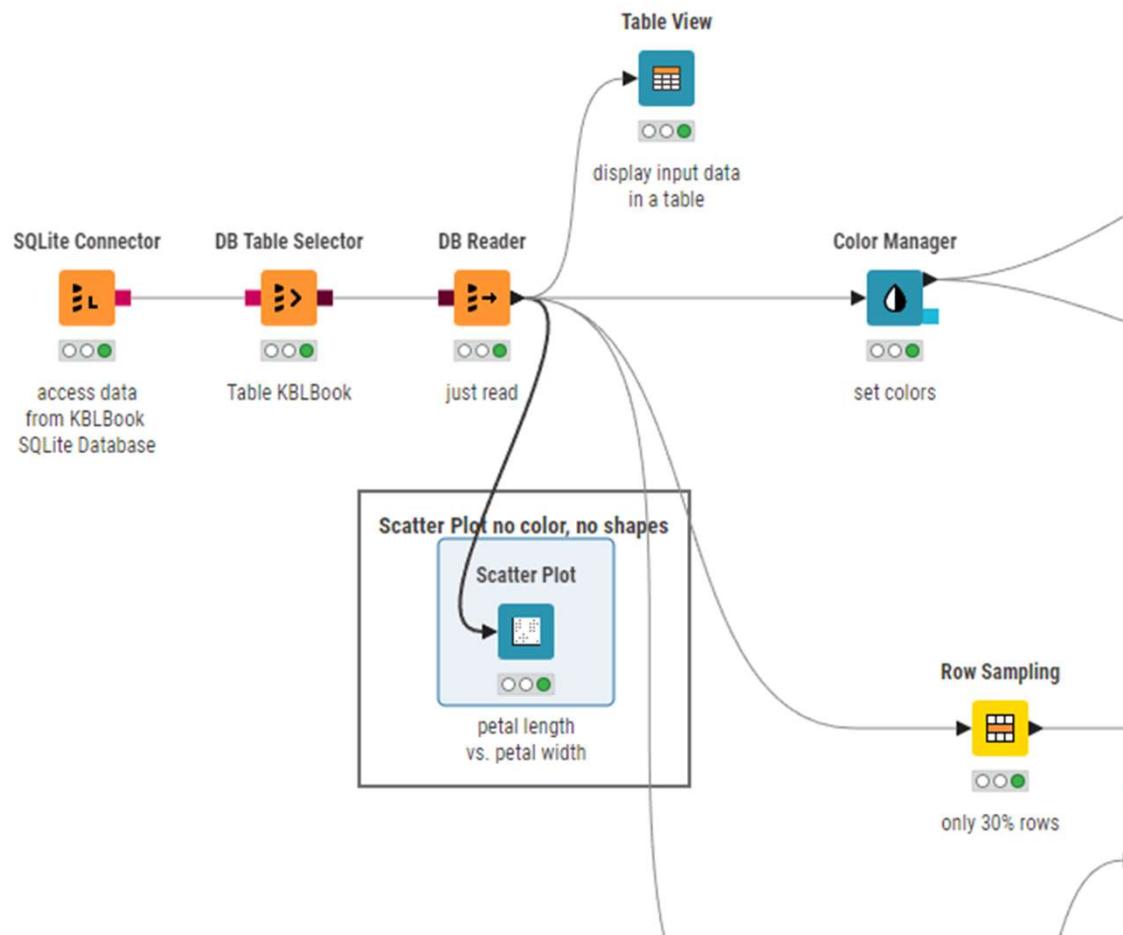
Das Iris Flower Data Set besteht aus 150 Datenpunkten, jeweils 50 für jede der drei Iris-Arten. Jeder Datenpunkt enthält vier Merkmale: **Sepal Length (Kelchlänge)**, **Sepal Width (Kelchbreite)**, **Petal Length (Blütenblattlänge)** und **Petal Width (Blütenblattbreite)**, alle gemessen in Zentimetern. Diese Merkmale dienen als Eingabeveriablen für maschinelle Lernalgorithmen. Das Hauptziel besteht darin, anhand dieser Merkmale die Iris-Art korrekt zu klassifizieren.

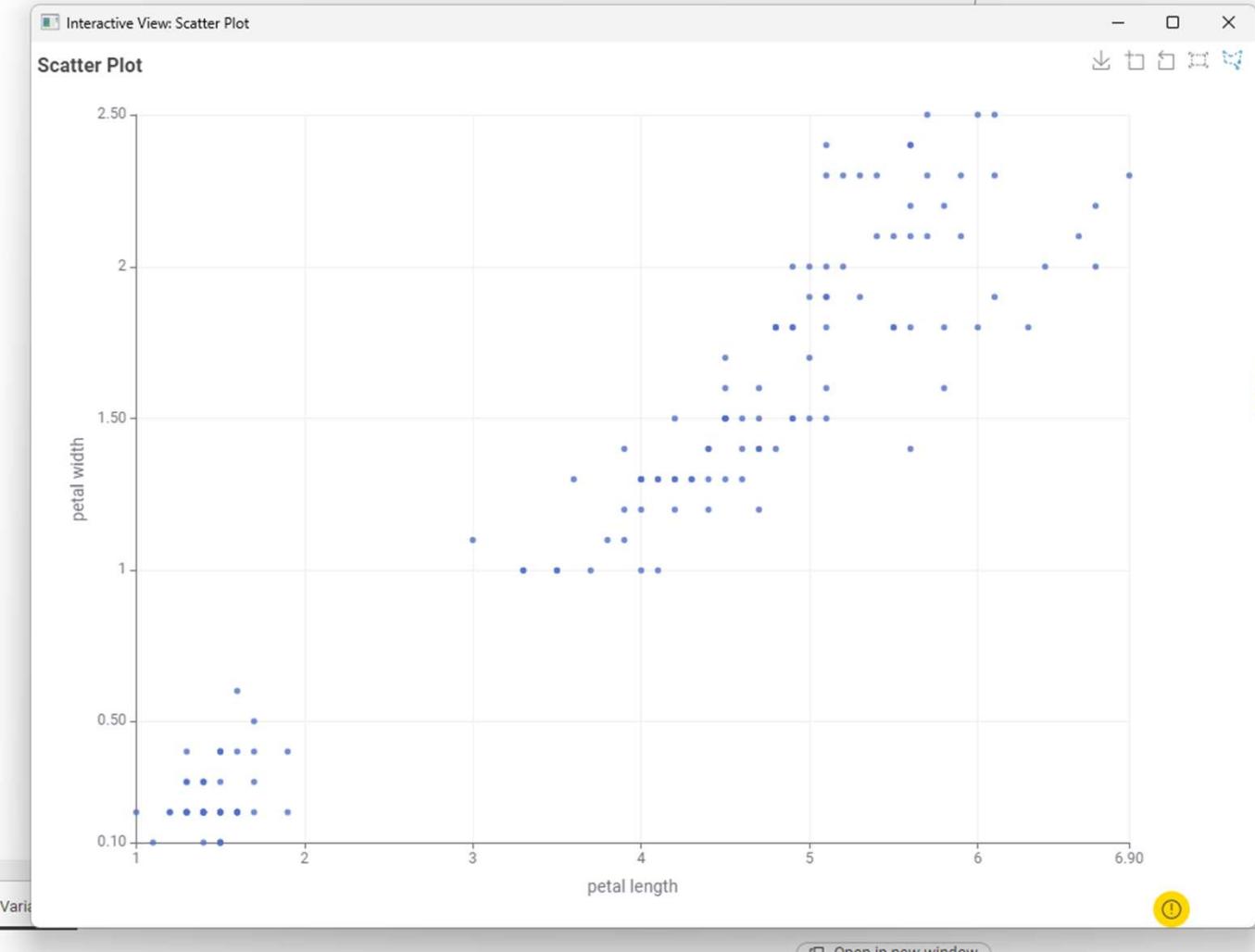






# Übung 2: Data Exploration





Sampling  
30% rows

GroupBy

Sorter

ascending on  
sepal length binned

Pivot

Parallel Coordinates Plot

Parallel Coordinates Plot

all numerical  
features

Scatter Plot with colors, no shapes

Scatter Plot

petal length  
vs. petal width

Line Plot

Line Plot

all numerical dimensions  
(selected)

Histograms & Bar Charts

Bar Chart

sepal length histogram

Bar Chart

average count for

# Übung 2: Data Exploration

Interactive View: Table View

Table View

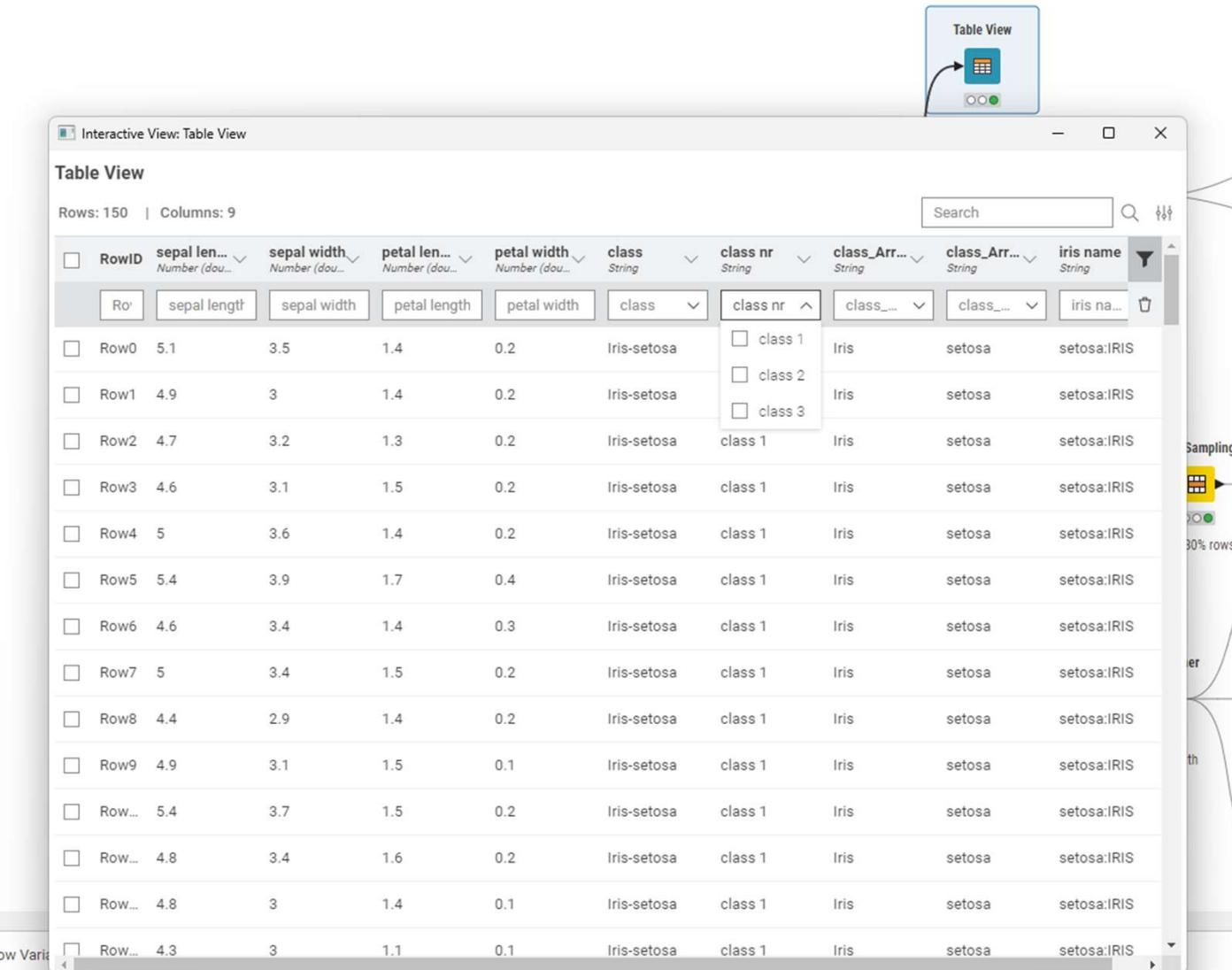
Rows: 150 | Columns: 9

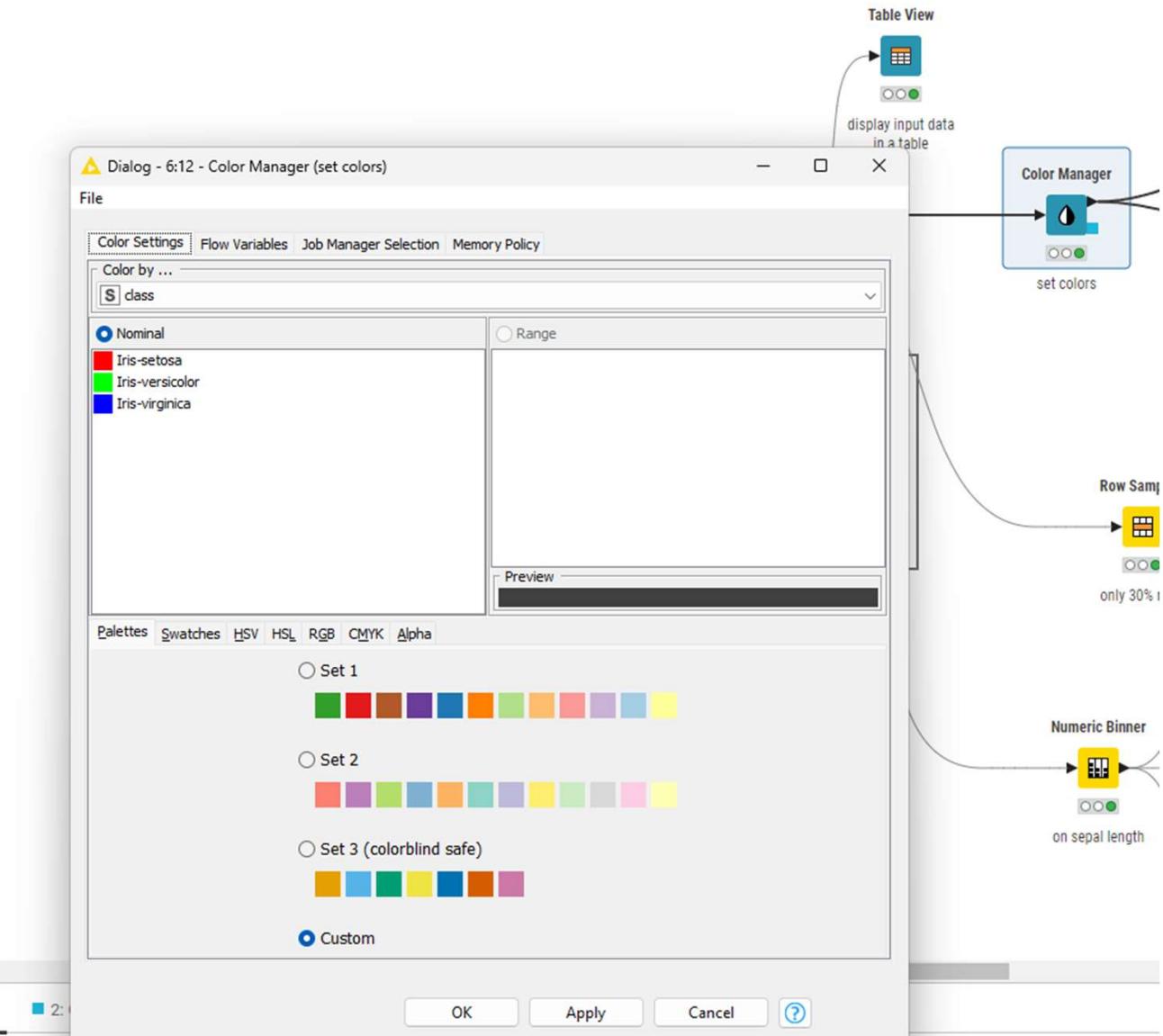
Search  Filter

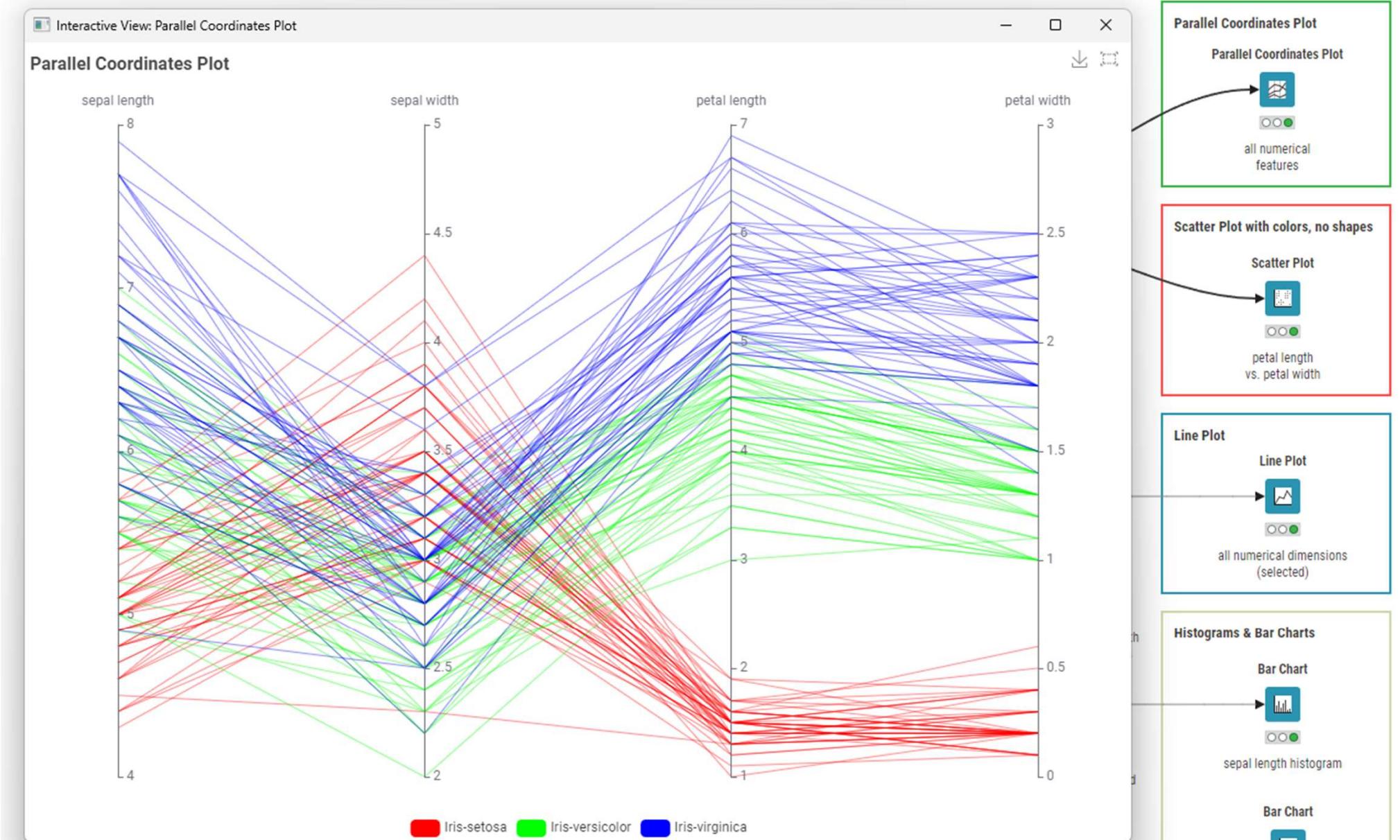
RowID	sepal length	sepal width	petal length	petal width	class	class nr	class_Arr...	class_Arr...	iris name
Row0	5.1	3.5	1.4	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row1	4.9	3	1.4	0.2	Iris-setosa	class 2	Iris	setosa	setosa:IRIS
Row2	4.7	3.2	1.3	0.2	Iris-setosa	class 3	Iris	setosa	setosa:IRIS
Row3	4.6	3.1	1.5	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row4	5	3.6	1.4	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row5	5.4	3.9	1.7	0.4	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row6	4.6	3.4	1.4	0.3	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row7	5	3.4	1.5	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row8	4.4	2.9	1.4	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row9	4.9	3.1	1.5	0.1	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row...	5.4	3.7	1.5	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row...	4.8	3.4	1.6	0.2	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row...	4.8	3	1.4	0.1	Iris-setosa	class 1	Iris	setosa	setosa:IRIS
Row...	4.3	3	1.1	0.1	Iris-setosa	class 1	Iris	setosa	setosa:IRIS

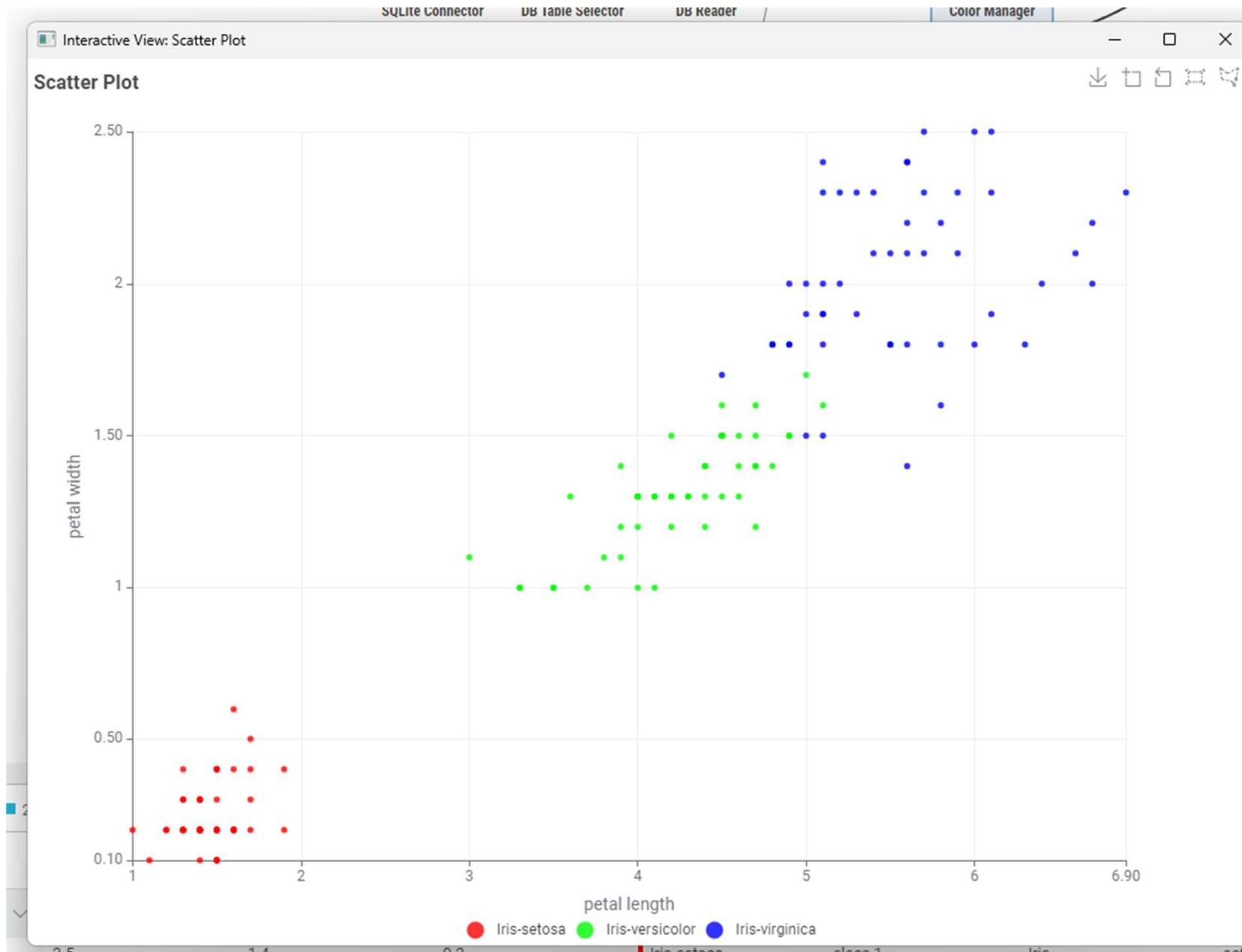
Sampling  30% rows

Flow Variables



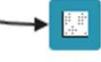






### Scatter Plot with colors, no shapes

## Scatter Plot



## petal length vs. petal width

## Line Plot

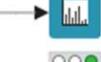
## Line Plot



all numerical dimensions  
(selected)

10 of 10

## Bar Chart



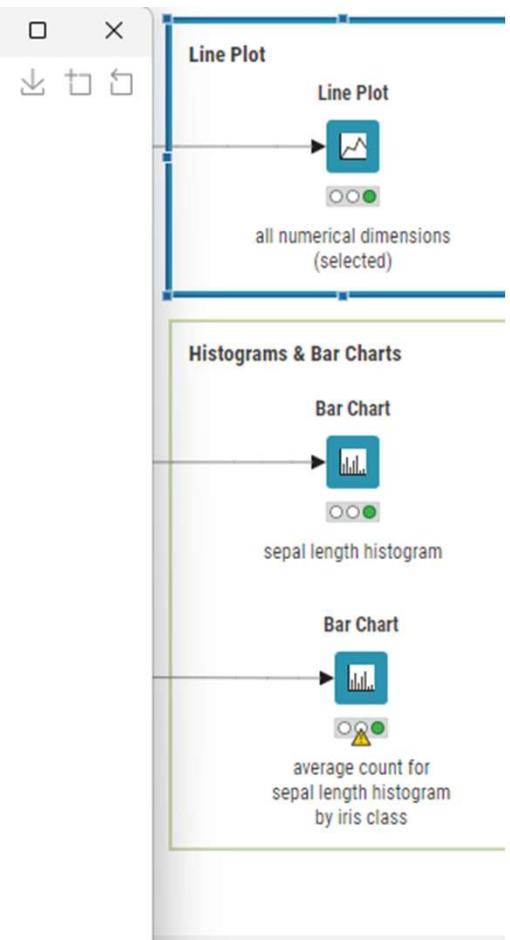
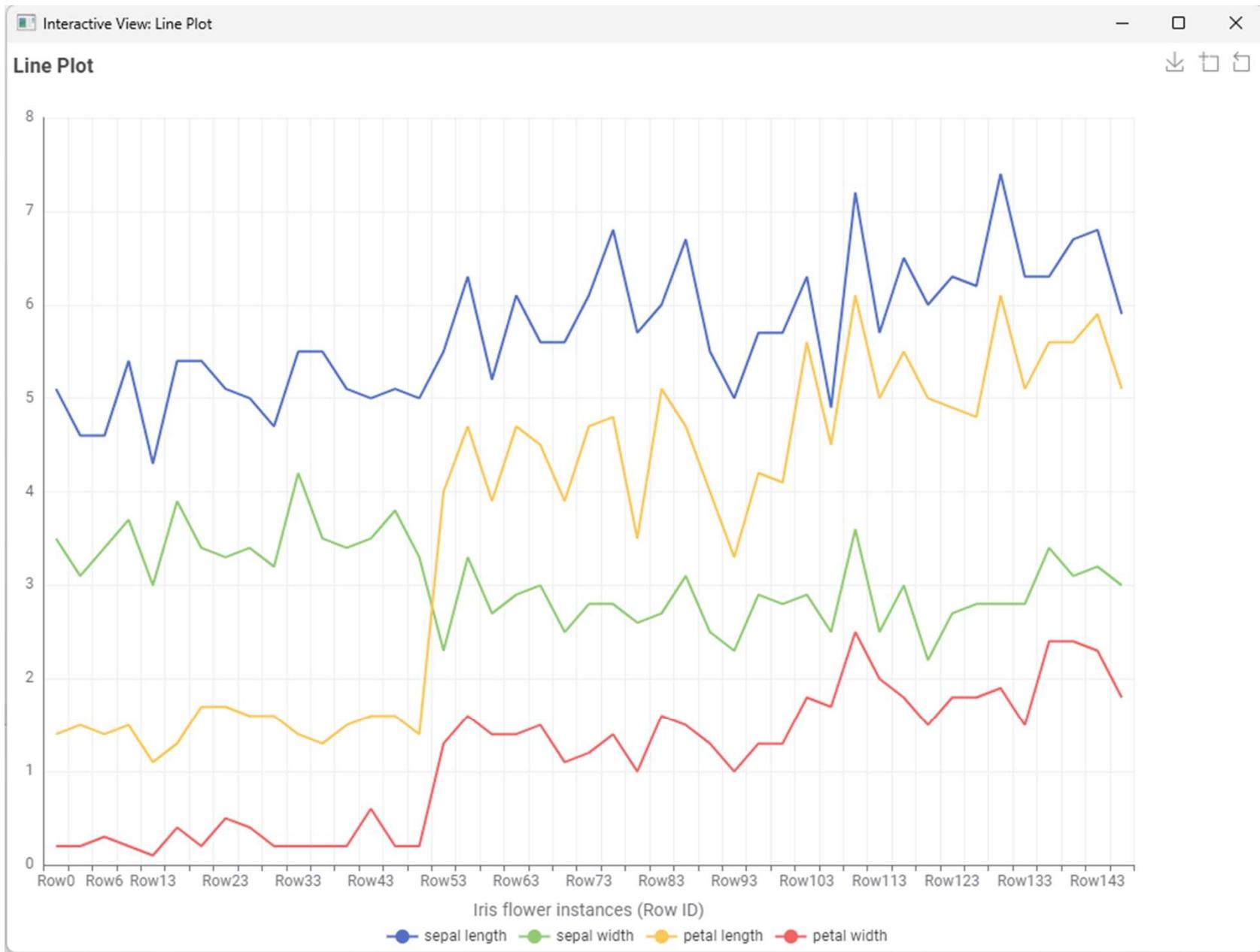
## sepal length histogram

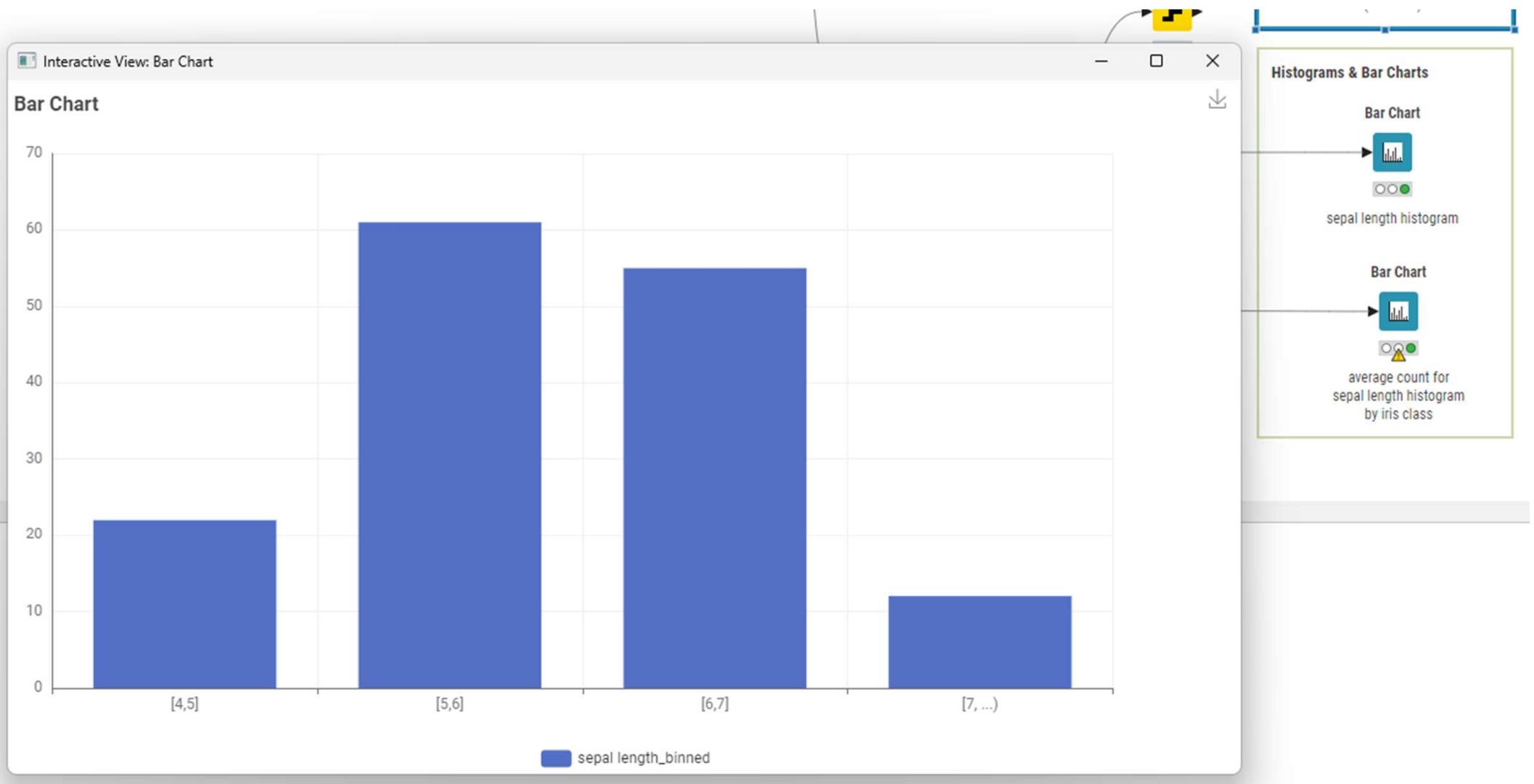
## Bar Chart



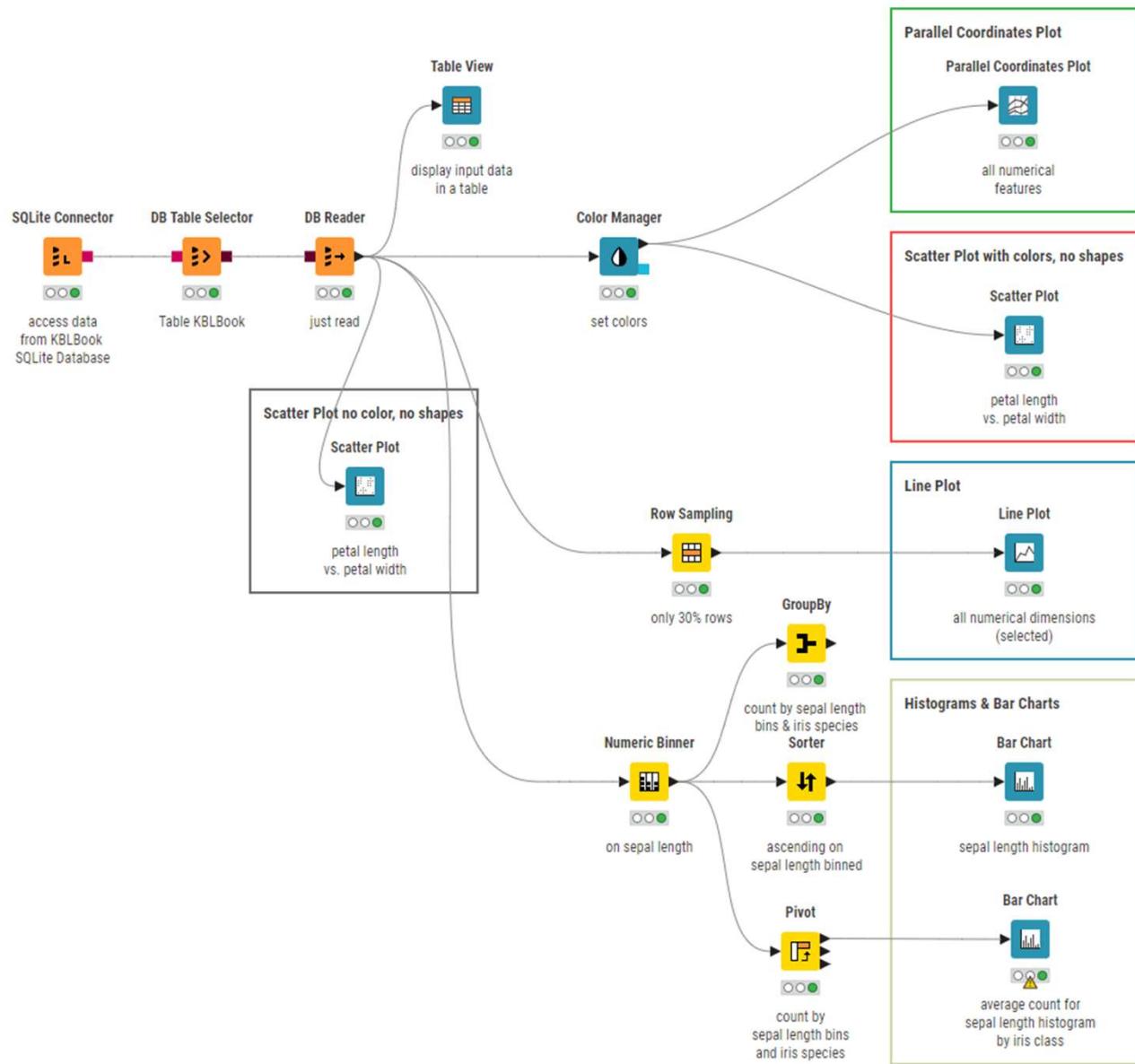
s\_Arr[1]

**iris name**  
*String*





# Übung 2: Data Exploration Ende



# Übung

## KNIME Data Exploration

### Zusammenfassung

- Datensatz Schwertlilie Data  
Exploration

Dienstag  
**Datenanalyse in der  
Praxis**

Praktische Einführung in  
die Datenanalyse:  
Methoden der  
explorativen  
Datenanalyse

Anwendungsfälle aus  
verschiedenen  
Branchen: Wie  
Datenanalyse  
branchenübergreifend  
genutzt wird?

Übungen

# Übung

## KNIME Data Exploration

### Zusammenfassung

- Spaces auf KNIME Hub mit 1000 Beispiel Workflows für alle Branchen und Anwendungsbereiche
- Workflows kompakt halten und dokumentieren
- Workflow Sunburst und Stacked Area Chart

Dienstag  
Datenanalyse in der Praxis

Praktische Einführung in die Datenanalyse:  
Methoden der explorativen Datenanalyse

Anwendungsfälle aus verschiedenen Branchen: Wie Datenanalyse branchenübergreifend genutzt wird?

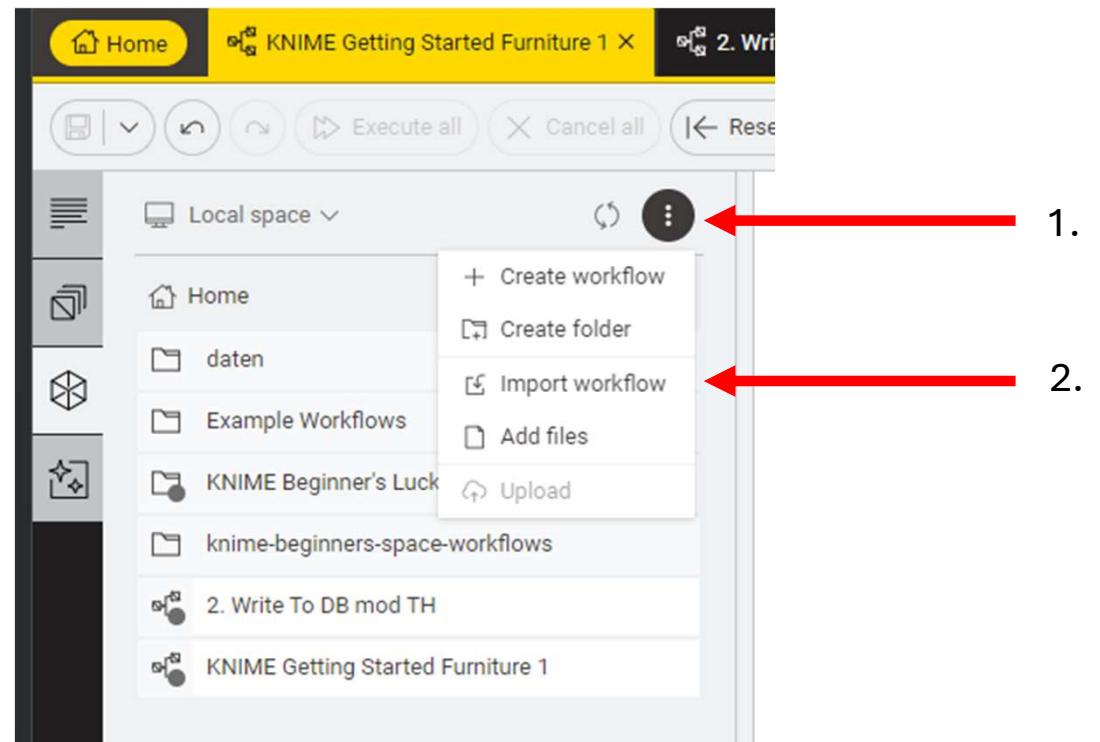
Übungen

# Übung: KNIME Beginner Space

<https://hub.knime.com/knime/spaces/Beginners%20Space/~Ln1fgQnWKKeRceeP/>

<https://hub.knime.com/knime>

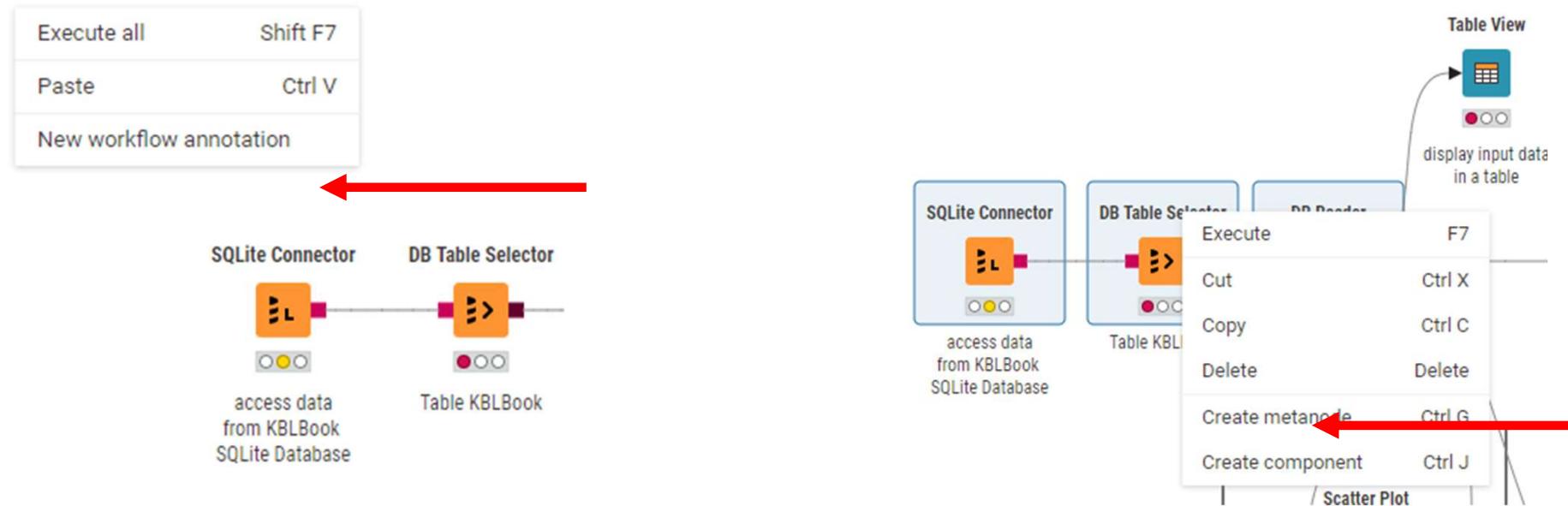
Importieren von .knar files in den workspace:



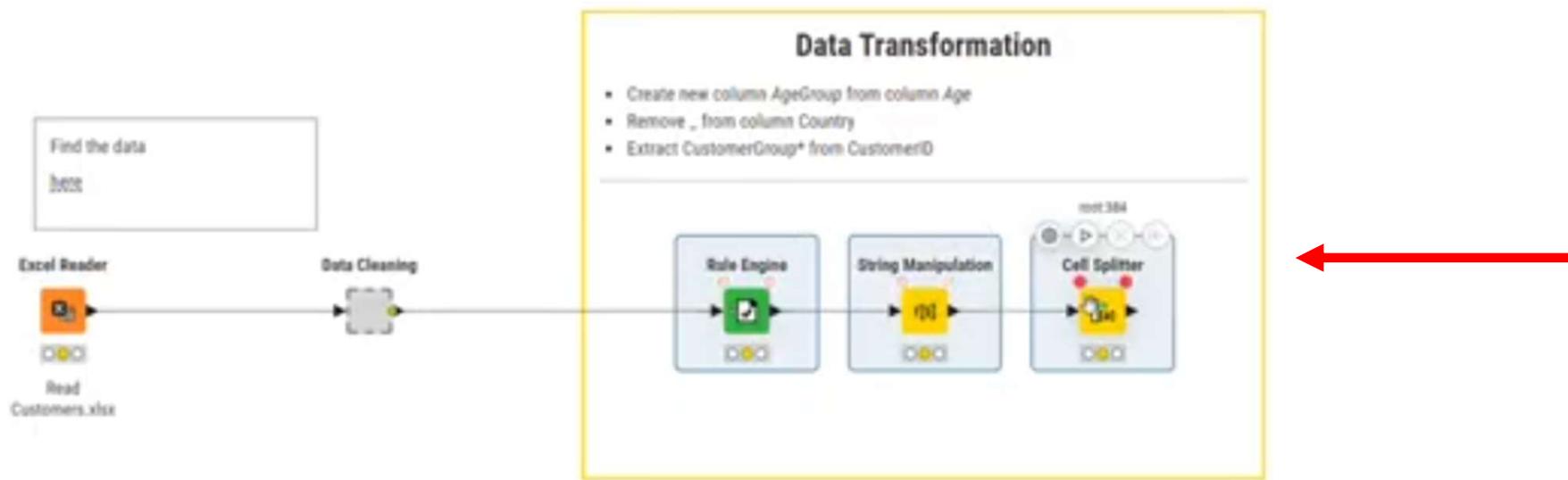
Intro to Machine Learning

<https://knime.learnupon.com/enrollments/219301013/page/848498723>

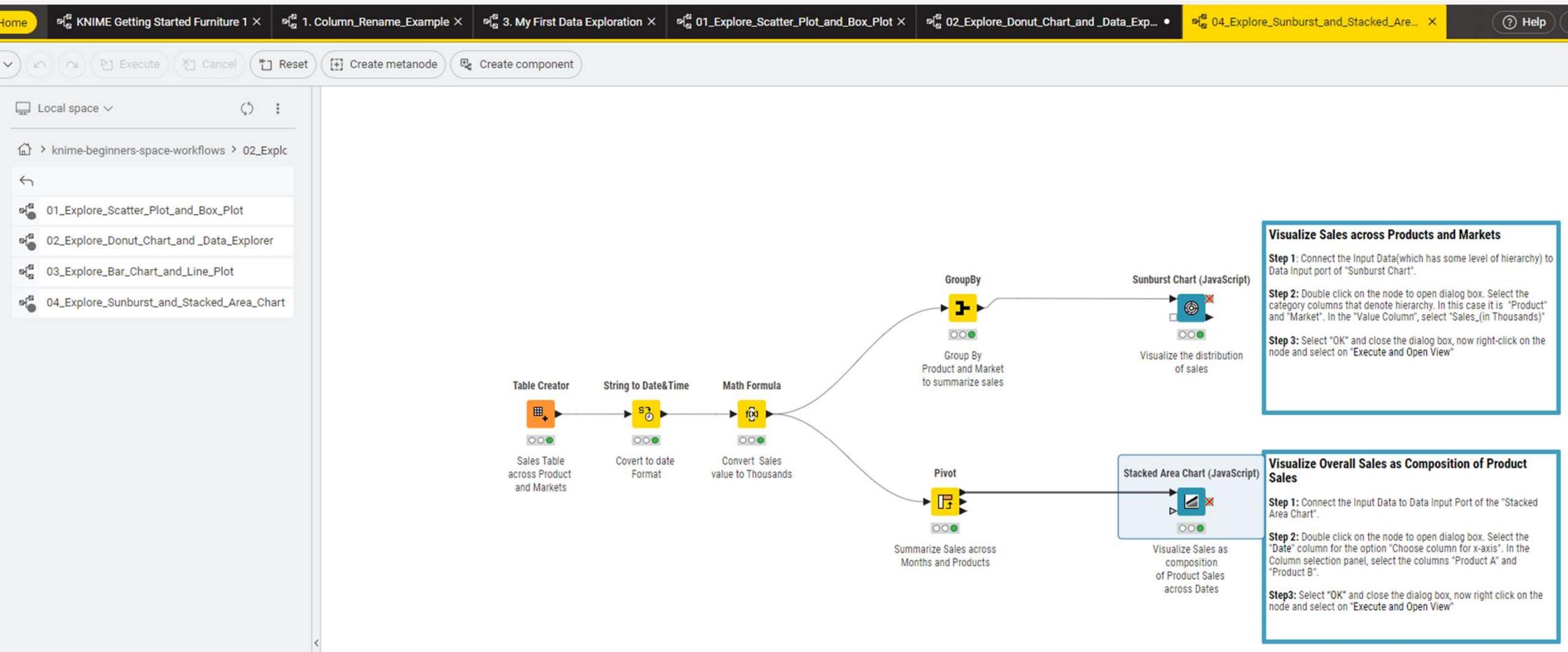
# Workflows Dokumentieren und Kompakt halten



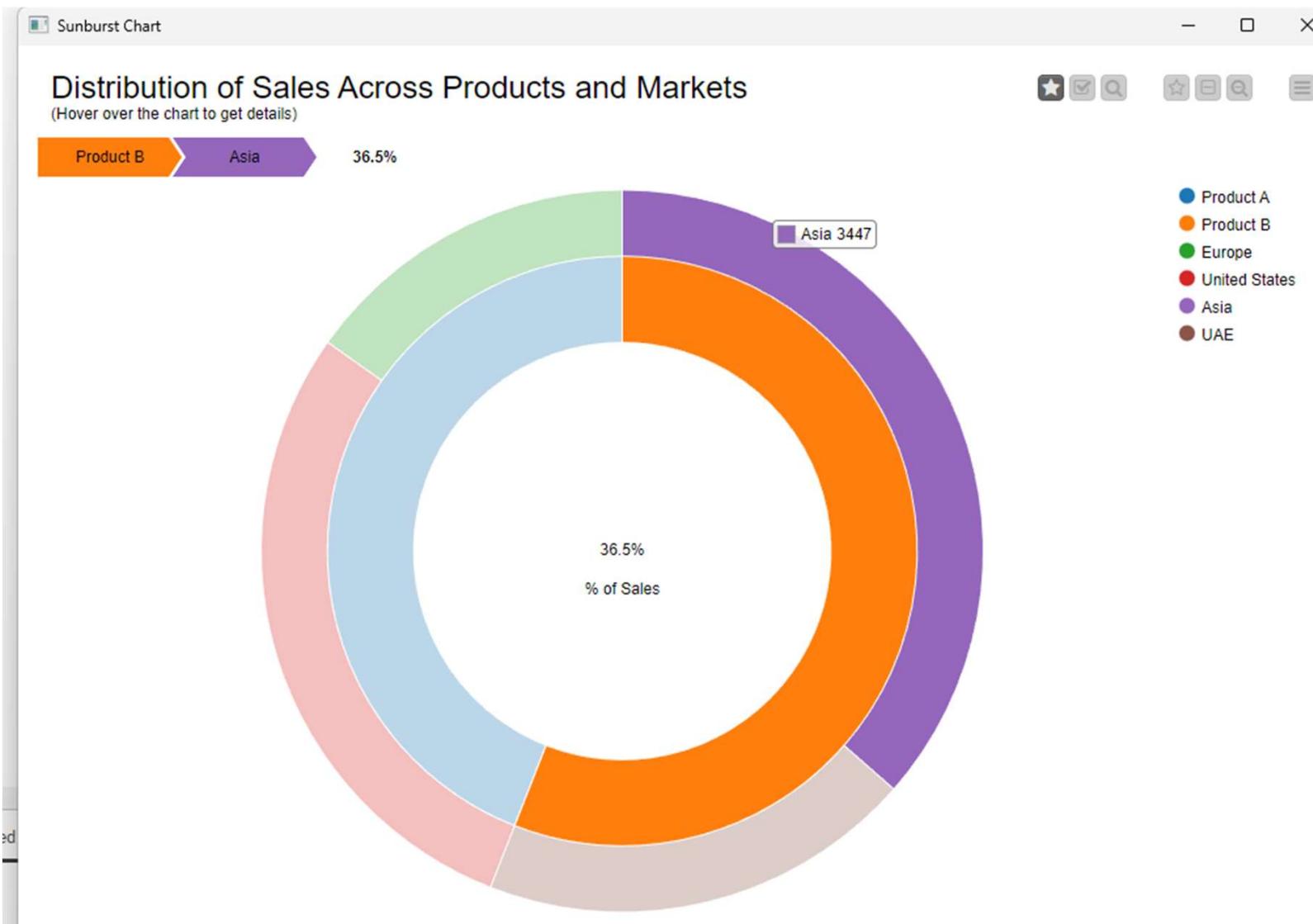
# Workflows Dokumentieren und Kompakt halten



# Data Exploration



# Data Exploration



# Übung

## KNIME Data Exploration

### Zusammenfassung

- Spaces auf KNIME Hub mit 1000 Beispiel Workflows für alle Branchen und Anwendungsbereiche
- Workflows kompakt halten und dokumentieren
- Workflow Sunburst und Stacked Area Chart

Dienstag  
Datenanalyse in der Praxis

Praktische Einführung in die Datenanalyse:  
Methoden der explorativen Datenanalyse

Anwendungsfälle aus verschiedenen Branchen: Wie Datenanalyse branchenübergreifend genutzt wird?

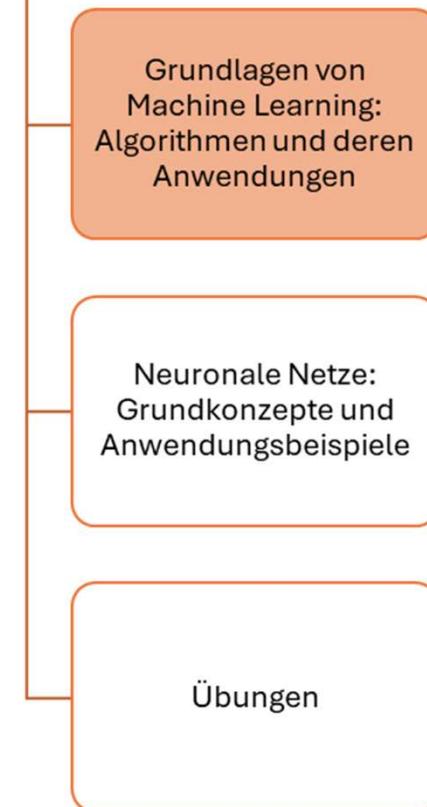
Übungen

# Grundlagen von Machine Learning: Algorithmen und deren Anwendungen

## Zusammenfassung

- Überwachtes Lernen
- Unüberwachtes Lernen
- Lineare/Multiple Regression
- KNIME Workflow Lineare Regression Hauspreise
- DS KI-Tutor Lineare Regression Hauspreise
- Klassifikation & Entscheidungsbäume
- Random Forest & Ensemble Learning
- Clustering & k-Means

**Mittwoch**  
**Maschinelles Lernen & Vorhersagemodelle**

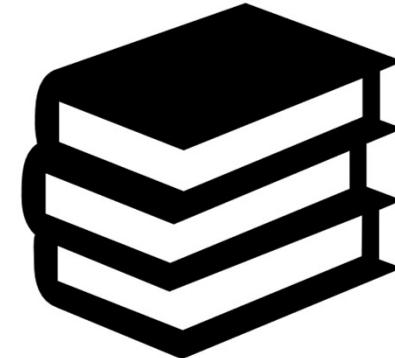


# Grundlegende Methoden Maschinelles Lernen



## Überwachtes Lernen

- Leite aus den Mustern in meinen Daten Prognosen ab !



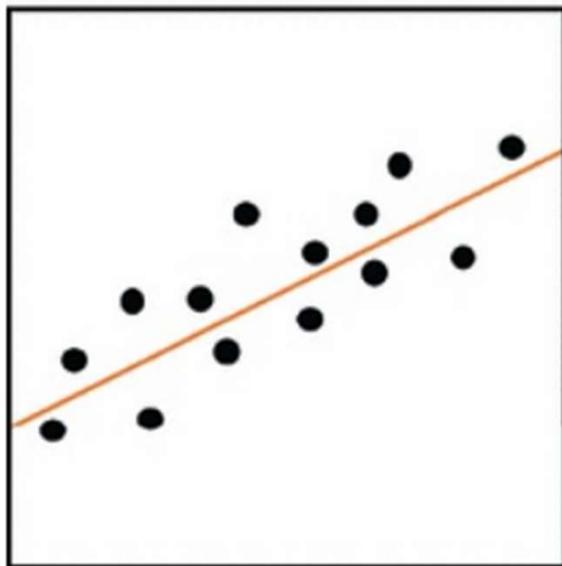
## Unüberwachtes Lernen

- Sag mir, was für Muster in meinen Daten verborgen sind !

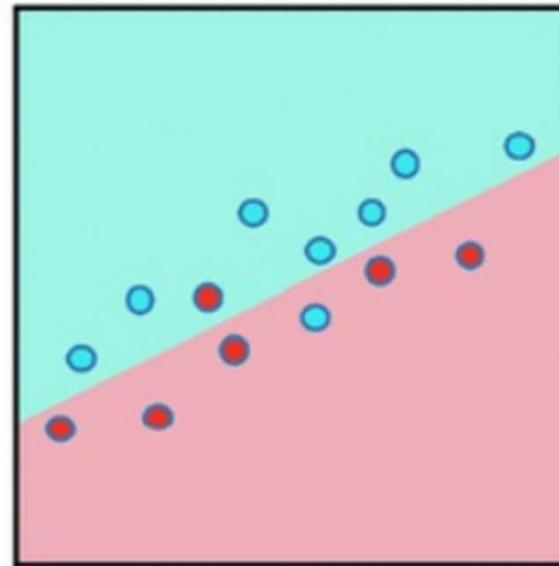


# Grundlegende Methoden Überwachtes Lernen

Leite aus den Mustern in meinen Daten Prognosen ab !



a) Regression



b) Klassifikation

**Abb. 1.1** Bei einer Regressionsanalyse wird eine Trendlinie ausgerechnet, während eine Klassifikation die Einteilung von Datenpunkten in Gruppen erfordert. Beachten Sie, dass in beiden Aufgabentypen Fehler zu erwarten sind. Bei einer Regression weichen Datenpunkte mehr oder weniger von der Trendlinie ab, während bei einer Klassifikation Datenpunkte möglicherweise in die falsche Kategorie sortiert werden



# Grundlegende Methoden Maschinelles Lernen

Kategorie	Methode	Anwendungsgebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
Überwachtes Lernen	Supervised Learning	Marketing	Kundenklassifikation im Marketing	Modell wird mit gekennzeichneten Daten trainiert. Es lernt, Vorhersagen zu treffen.	Ein Lehrer (Daten) lehrt einen Schüler (Modell) auf Basis von Beispielaufgaben.	Hohe Genauigkeit bei Vorhersagen, breite Anwendbarkeit	Benötigt große, gekennzeichnete Datensätze
	Prognosen (Lineare Regression)	Finanzen	Umsatzvorhersage	Ermittelt die Beziehung zwischen unabhängigen und abhängigen Variablen. Nutzt eine Linie, um die besten Fit für die Datenpunkte darzustellen.	Eine Linie, die den besten Fit für die Datenpunkte darstellt.	Einfach zu interpretieren, gut für lineare Beziehungen	Ungeeignet für nicht-lineare Zusammenhänge
	Klassifikation (Decision Trees)	Finanzwesen	Kreditwürdigkeitsprüfung	Entscheidungen werden basierend auf Datenattributen in einer Baumstruktur getroffen. Der Baum teilt die Daten in immer kleinere Gruppen auf.	Entscheidungsprozess wie ein Entscheidungsbaum im Alltag.	Leicht zu verstehen und zu visualisieren	Neigung zu Überanpassung (Overfitting)
	Klassifikation (Random Forest)	Zahlungsverkehr	Betrugs-erkennung	Kombination vieler Entscheidungsbäume, um Vorhersagen zu verbessern. Jeder Baum trägt zur finalen Entscheidung bei.	Abstimmung einer großen Gruppe von Experten.	Hohe Genauigkeit, reduziert Überanpassung	Komplex, schwer zu interpretieren
	Klassifikation (XGBoost)	Marketing-analyse	Zielgruppen-segmentierung	Optimierte Boosting-Algorithmus kombiniert viele Entscheidungsbäume. Er lernt aus den Fehlern vorheriger Bäume.	Training eines Expertenteams, wobei jeder Experte auf Fehler des anderen aufbaut.	Sehr hohe Leistung und Effizienz	Komplexität, erfordert gründliche Parameter-einstellung



Überwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen: Lineare Regression

Kategorie	Methode	Anwendungsgebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
	Prognosen (Lineare Regression)	Finanzen	Umsatzvorhersage	Ermittelt die Beziehung zwischen unabhängigen und abhängigen Variablen. Nutzt eine Linie, um die besten Fit für die Datenpunkte darzustellen.	Eine Linie, die den besten Fit für die Datenpunkte darstellt.	Einfach zu interpretieren, gut für lineare Beziehungen	Ungeeignet für nicht-lineare Zusammenhänge



Überwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen: Lineare Regression

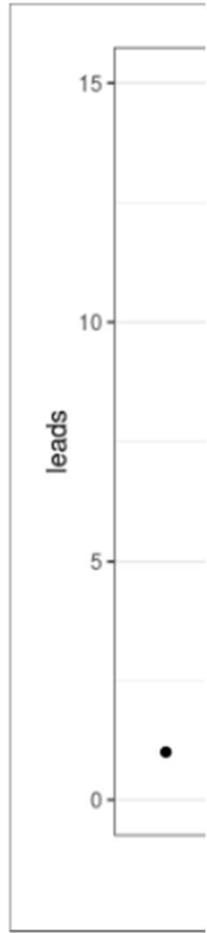


Abbildung 6.

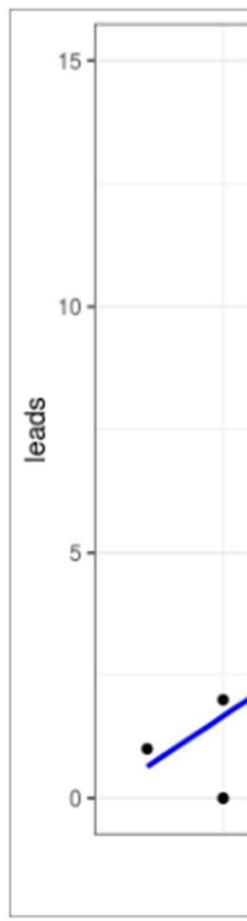


Abbildung 6.2

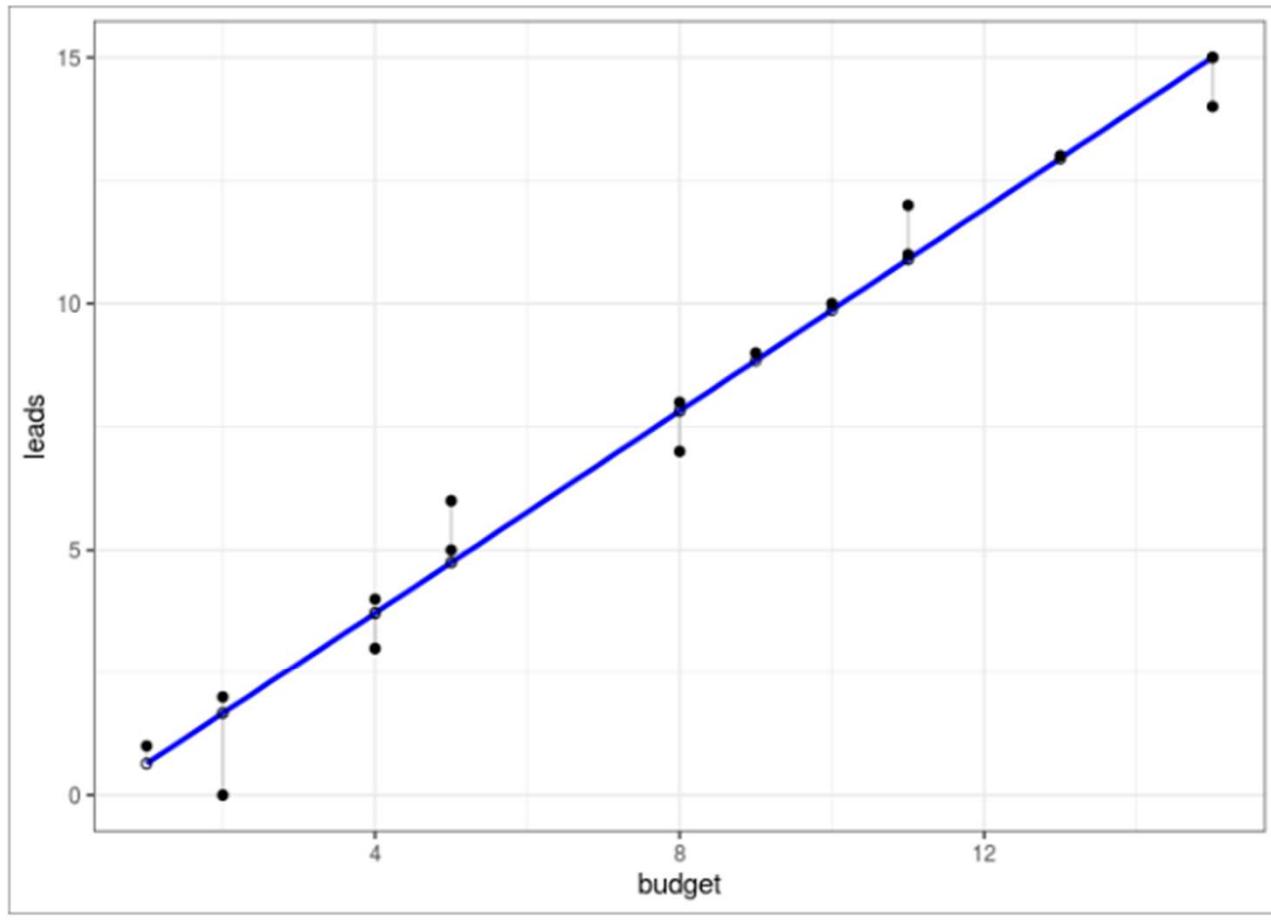


Abbildung 6.3 Dieselbe Gerade wie in Abbildung 6.2, hier mit Abständen zu den Datenpunkten

# Grundlegende Methoden Maschinelles Lernen: Lineare Regression

- **Lineare Regression:**

- "Hello World" des Machine Learnings, Ursprung bei Legendre und Gauß im 19. Jahrhundert.
- Erklärt eine abhängige Variable durch eine oder mehrere unabhängige Variablen.
- Ziel: Beste Vorhersage der abhängigen Variablen, im Gegensatz zur Korrelation (keine Kausalität).
- Beispiel: Beziehung zwischen Marketing-Budget und Leads.

- **Vorgehensweise:**

- Vielzahl von Linien durch Punkte im Plot zeichnen.
- Abstände der Punkte zur Linie berechnen (Residuen).
- Optimalste Linie: Minimale Summe der quadrierten Abstände.

- **Fehler und Abstände:**

- Fehler (Residuen) messen und minimieren ist grundlegend für viele Machine-Learning-Algorithmen.

- **Modellbewertung:**

- Residuenverteilung: Normalverteilung der Residuen ist Voraussetzung.
- Signifikanz der unabhängigen Variablen prüfen (p-Wert, R-squared).
- R-squared und Adjusted R-squared als Maß der Modellgüte.

- **Vor- und Nachteile:**

- Einfacher Algorithmus, aber Gefahr der falschen Anwendung.
- Eignung der Daten für lineare Regression prüfen.
- Beispiel: Lineare Regression bei nichtlinearen Daten ist nicht geeignet.

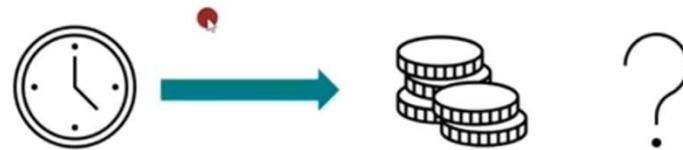
# Grundlegende Methoden Maschinelles Lernen: Lineare Regression

## Regressionsanalyse

### Reg Formen der Regressionsanalyse

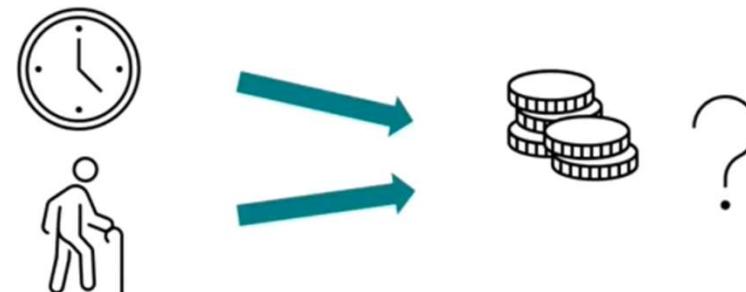
#### Einfache lineare Regression

Hat die **wöchentliche Arbeitszeit** einen Einfluss auf den **Stundenlohn** von Angestellten?



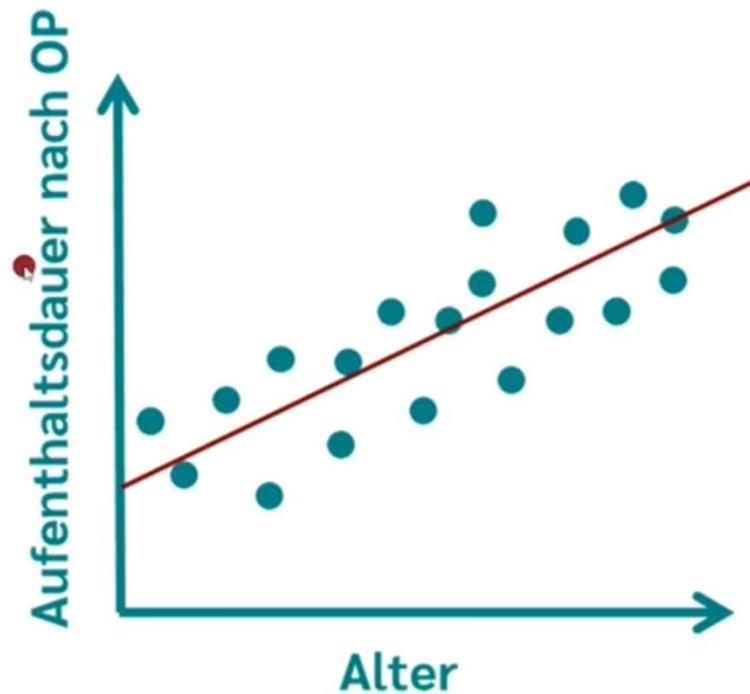
#### Multiple lineare Regression

Haben die **wöchentliche Arbeitszeit** und das **Alter** von Angestellten einen Einfluss auf deren **Stundenlohn**?



# Grundlegende Methoden Maschinelles Lernen: Lineare Regression

## E Einfache Lineare Regression



Geschätzte Aufenthaltsdauer Alter

$$\hat{y} = b \cdot x + a$$
$$\hat{y} = 0.14 \cdot x + 1.2$$
$$5.82 = 0.14 \cdot 33 + 1.2$$

Berechnung von a und b

$$b = r \frac{s_y}{s_x} \quad a = \bar{y} - b \cdot \bar{x}$$

## Grundlegende Methoden Maschinelles Lernen: Multiple Regression

# Multiple Regression

Einfachen linearen  
Regression

$$\hat{y} = b \cdot x + a$$

Multiple linearen  
Regression

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k + a$$



# Grundlegende Methoden Maschinelles Lernen: Lineare Regression KNIME

Last update: Jun 18, 2024

Analyze Data by Training a Linear Regression model for House Price Prediction

This workflow is an example of how to train a basic machine learning model - Linear Regression for a house price prediction task

If you want to give your acquired knowledge a test run, take a look at the Just KNIME It challenges. (Just KNIME It Link provided in External resources section of this workflow)

**KNIME Analytics Platform**

Home getting started • 04\_Analyze\_Linear\_Regression X

Execute all Cancel all Reset all

The diagram illustrates a KNIME workflow for linear regression. It consists of four main stages: **Read Data**, **Transform**, **Train Model**, and **Evaluation**.

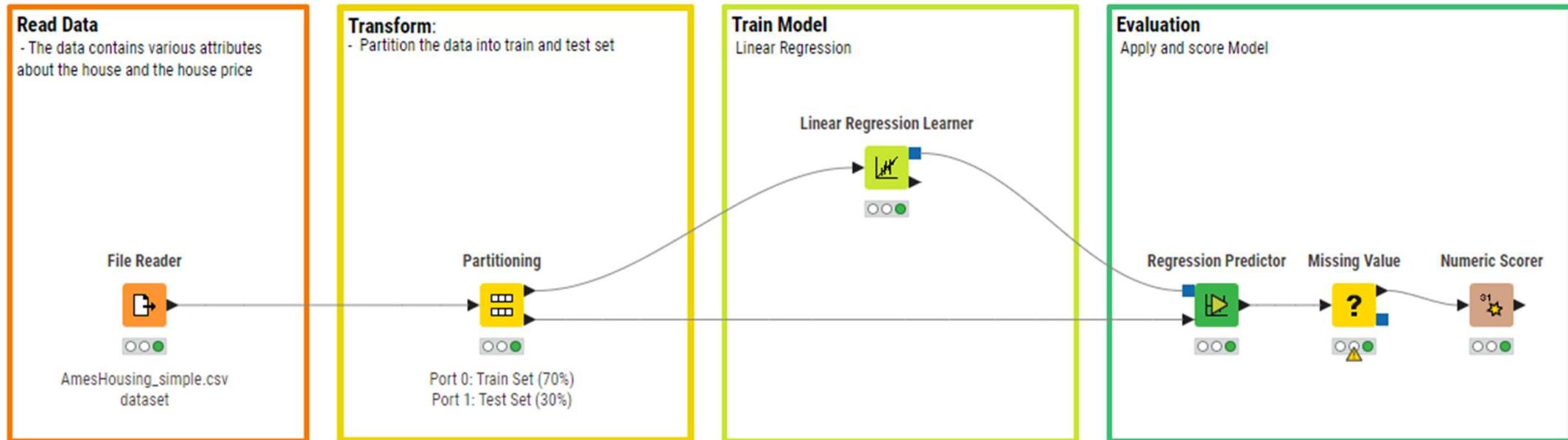
- Read Data:** A "File Reader" node reads the "AmesHousing\_simple.csv" dataset.
- Transform:** A "Partitioning" node splits the data into a "Port 0: Train Set (70%)" and a "Port 1: Test Set (30%)".
- Train Model:** A "Linear Regression Learner" node is trained on the training set.
- Evaluation:** The trained model is evaluated using a "Regression Predictor", "Missing Value" node, and "Numeric Scorer" on the test set.

Two callout boxes provide instructions:

- How to train Linear Regression Model?**
  - Drag the "Linear Regression Learner" node and double click to open the dialog.
  - Select the "Target" Column as "SalesPrice".
  - Right Click on the node and select "Execute and Open View" to train the model and to get a view of the Regression Co-efficients.
- How to evaluate Classification Model?**
  - Drag the "Regression Predictor" node.
  - Connect the output of "Linear Regression Learner" node to model input port and Test Dataset to data input port. Execute the node.
  - Connect the Predictor Output to "Missing Value" node to remove rows with missing prediction and then connect it to "Scorer" node to evaluate the model on various evaluation measures.

Workflow under : KNIME Beginners space, 04\_Analyze, 04\_Analyze Linear Regression

# Grundlegende Methoden Maschinelles Lernen: Lineare Regression KNIME



## How to train Linear Regression Model?

- Step 1: Drag the "Linear Regression Learner" node and double click to open the dialog
- Step 2: Select the "Target" Column as "SalesPrice"
- Step 3: Right Click on the node and select "Execute and Open View" to train the model and to get a view of the Regression Co-efficients

## How to evaluate Classification Model?

- Step 1: Drag the "Regression Predictor" node.
- Step 2: Connect the output of "Linear Regression Learner" node to model input port and Test Dataset to data input port. Execute the node
- Step 3: Connect the Predictor Output to "Missing Value" node to remove rows with missing prediction and then connect it to "Scorer" node to evaluate the model on various evaluation measures

# Data Science: KI Tutor Methoden

**t1p.de/ds-tutor1**

Free Account 10 Messages free (evtl. mit 2 Accounts arbeiten)

Paid Account (23€ / Monat / monatlich kündbar) 40-80 Messages per Day inkludiert

Weitere Alternative Bing Chat (6 Messages pro Chat free)

# Grundlegende Methoden Maschinelles Lernen: Lineare Regression KNIME

Data Science Tutor v6 ▾

AmesHousing Simple

	BsmtFin SF 1	Bsmt Unf SF	Total Bsmt SF	1st Flr SF	2nd Flr SF	Gr Liv Area	SalePrice
1	639	441	1080	1656	0	1656	215000
2	468	270	882	896	0	896	105000
3	923	406	1329	1329	0	1329	172000
4	1065	1045	2110	2110	0	2110	244000
5	791	137	928	928	701	1629	189900
6	602	324	926	926	678	1604	195500

analysiere den angehängten Datensatz

**Beschreibe und Analysieren den Datensatz. Wie kann ich die Verkaufspreise aufgrund der Flächenangaben prognostizieren. Beginne mit einer linearen Regression.**

Der Datensatz enthält Informationen über 2930 Immobilienverkäufe mit den folgenden 7 Spalten:

1. BsmtFin SF 1: Fläche des ausgebauten Kellers in Quadratfuß
2. Bsmt Unf SF: Fläche des unausgebauten Kellers in Quadratfuß
3. Total Bsmt SF: Gesamtfläche des Kellers in Quadratfuß
4. 1st Flr SF: Fläche des Erdgeschosses in Quadratfuß
5. 2nd Flr SF: Fläche des Obergeschosses in Quadratfuß
6. Gr Liv Area: Gesamtwohnfläche in Quadratfuß
7. SalePrice: Verkaufspreis des Hauses

### Analyse der Daten

1. Fehlende Werte: Es gibt einige fehlende Werte in den Spalten `BsmtFin SF 1`, `Bsmt Unf SF` und `Total Bsmt SF` (jeweils ein fehlender Wert).
2. Datentypen: Die meisten Spalten sind numerisch (`float64` und `int64`).

Lass uns die Verteilung der Verkaufspreise und die Beziehung zwischen den verschiedenen Flächenangaben und den Verkaufspreisen untersuchen.

## Schritt 1: Verteilung der Verkaufspreise

## Lineare Regression Hausverkauf

Ich werde die Verteilung der Verkaufspreise visualisieren. [x]

Analysiert

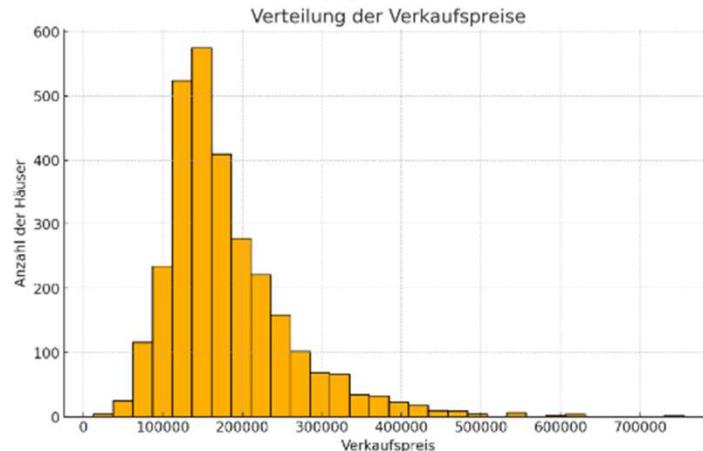
python

Details immer anzeigen  |  Code kopieren

```
import matplotlib.pyplot as plt

# Verteilung der Verkaufspreise
plt.figure(figsize=(10, 6))
plt.hist(data['SalePrice'], bins=30, edgecolor='black')
plt.title('Verteilung der Verkaufspreise')
plt.xlabel('Verkaufspreis')
plt.ylabel('Anzahl der Häuser')
plt.grid(True)
plt.show()
```

Verteilung Der Verkaufspreise



Verteilung der Verkaufspreise

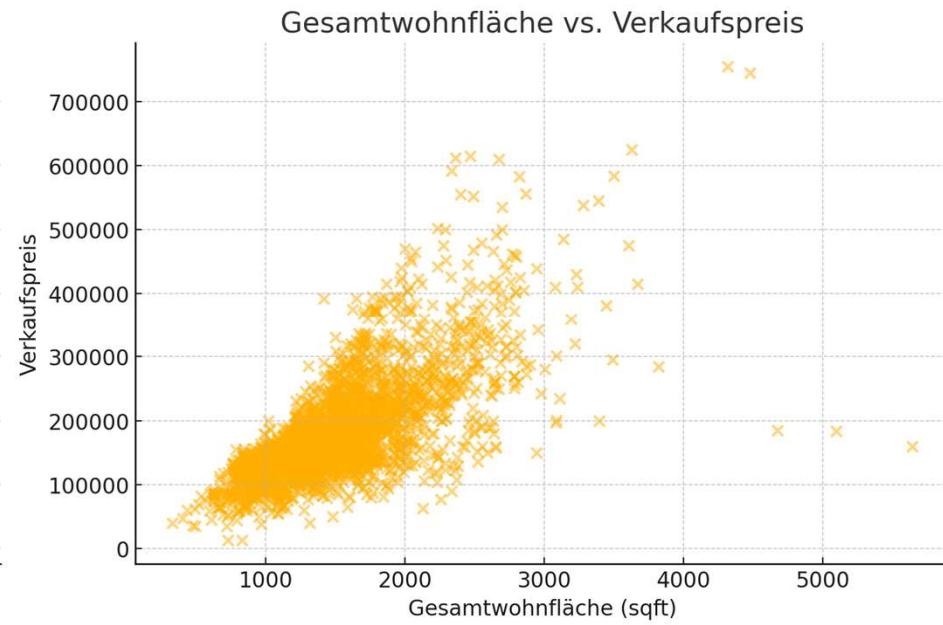
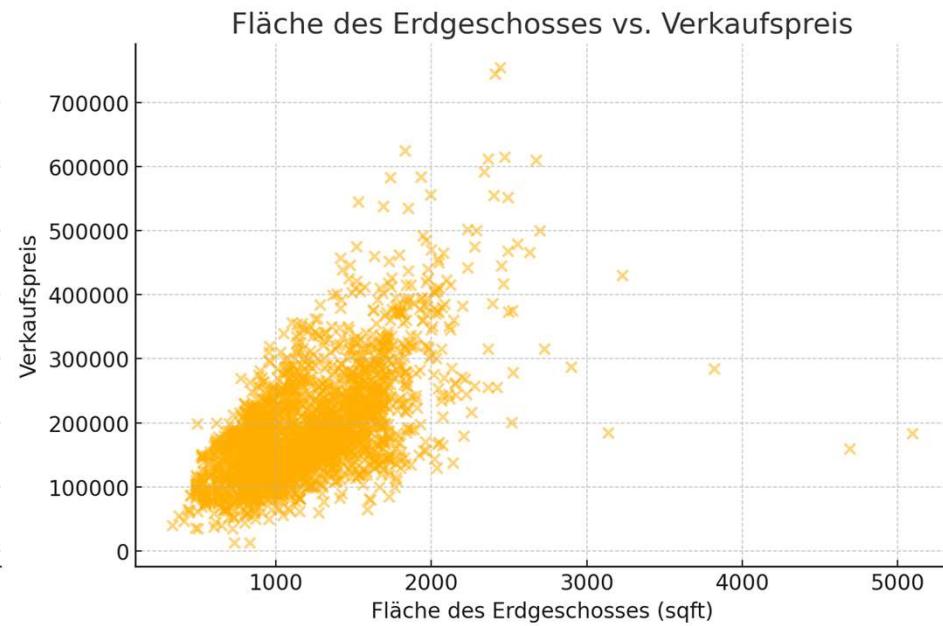
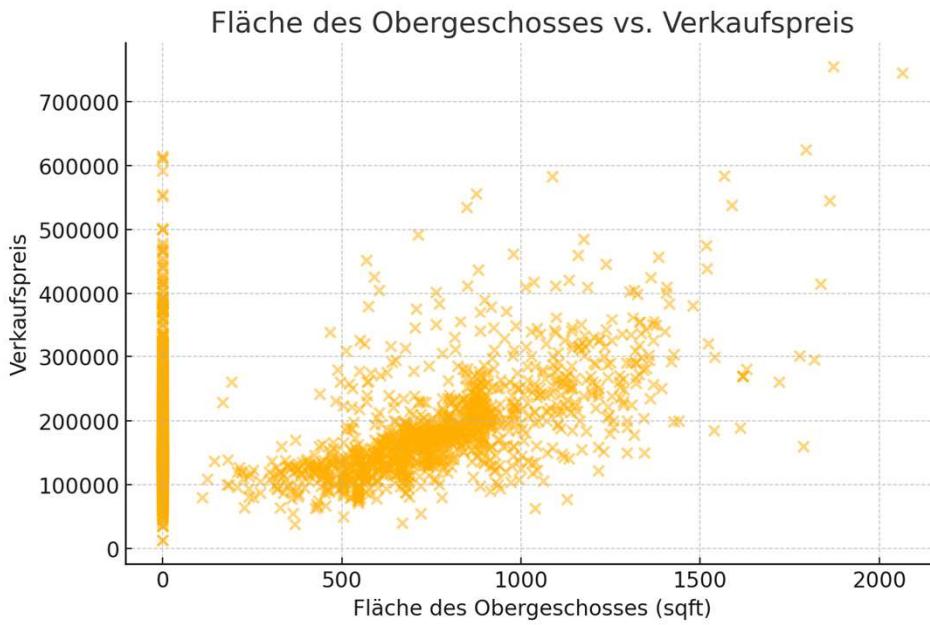
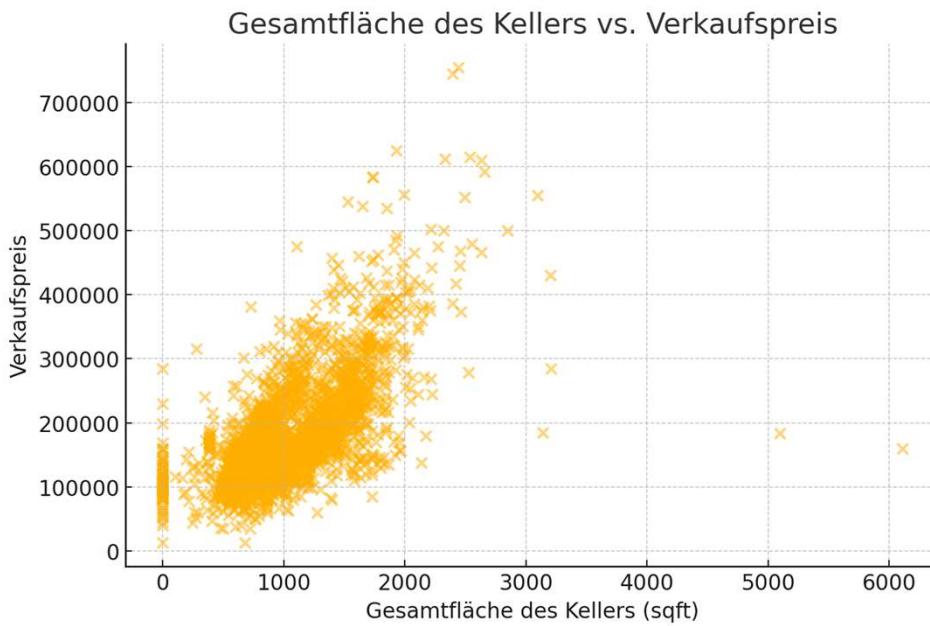
Die Verteilung der Verkaufspreise zeigt, dass die meisten Häuser zu Preisen zwischen 100.000 und 200.000 USD verkauft wurden, mit einem langen rechten Schwanz, was auf einige teurere Häuser hinweist.

## Schritt 2: Beziehung zwischen Flächenangaben und Verkaufspreis

Lass uns die Korrelationen zwischen den verschiedenen Flächenangaben und dem Verkaufspreis untersuchen und grafisch darstellen. Zuerst werde ich die Korrelationen berechnen und dann Scatterplots erstellen. [→]

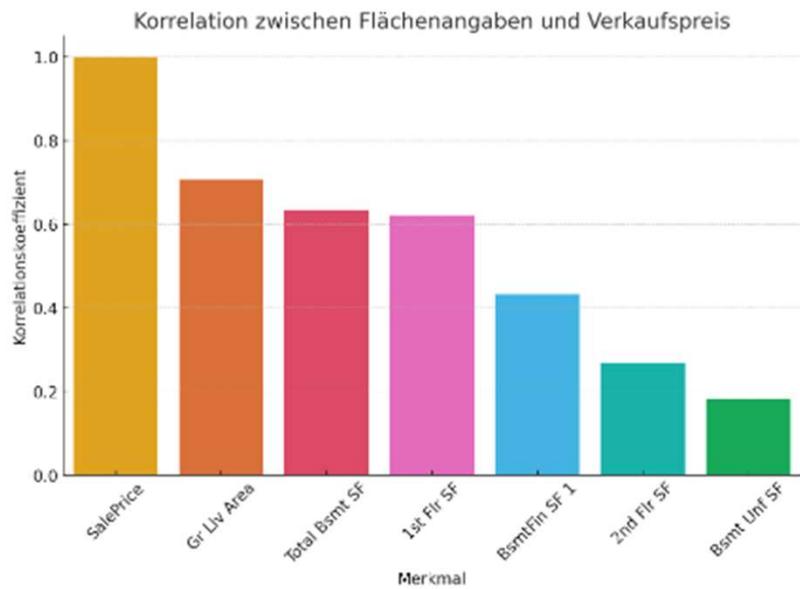
Korrelationen Der Variablen Mit Dem Verkaufspreis		
		SalePrice
1	SalePrice	1.0
2	Gr Liv Area	0.706779920976626 8
3	Total Bsmt SF	0.632280457106512
4	1st Flr SF	0.621676063270252 2
5	BsmtFin SF 1	0.432914410607395 2
6	2nd Flr SF	0.269373356589591 55

Bitte untersuche die Korrelation zwischen den verschiedenen Flächenangaben und dem Verkaufspreis und stelle dies grafisch dar.



Bitte untersuche die Korrelation zwischen den verschiedenen Flächenangaben und dem Verkaufspreis und stelle dies grafisch dar.

### Korrelation Zwischen Flächenangaben Und Verkaufspreis



Die grafische Darstellung zeigt die Korrelationen zwischen den verschiedenen Flächenangaben und dem Verkaufspreis. Hier sind die wichtigsten Erkenntnisse:

- Gr Liv Area hat die höchste positive Korrelation mit dem Verkaufspreis, was darauf hinweist, dass größere Wohnflächen tendenziell höhere Verkaufspreise erzielen.
- 1st Flr SF und Total Bsmt SF zeigen ebenfalls eine starke positive Korrelation mit dem Verkaufspreis.
- BsmtFin SF 1 und Bsmt Unf SF haben eine geringere, aber immer noch positive Korrelation.

Diese Ergebnisse bestätigen, dass verschiedene Flächenangaben, insbesondere die Wohnfläche über dem Boden, wichtige Faktoren für den Verkaufspreis sind.

## Analyse der Beziehungen

Die Scatterplots zeigen die Beziehung zwischen verschiedenen Flächenangaben und dem Verkaufspreis:

### 1. Gesamtfläche des Kellers vs. Verkaufspreis:

- Es gibt eine positive Korrelation: Häuser mit größerer Kellerfläche neigen dazu, teurer zu sein.

### 2. Fläche des Erdgeschosses vs. Verkaufspreis:

- Auch hier gibt es eine positive Korrelation: Größere Erdgeschossflächen sind mit höheren Verkaufspreisen verbunden.

### 3. Fläche des Obergeschosses vs. Verkaufspreis:

- Eine weniger klare, aber immer noch positive Korrelation: Häuser mit größeren Obergeschossen können teurer sein, obwohl die Streuung hier größer ist.

### 4. Gesamtwohnfläche vs. Verkaufspreis:

- Starke positive Korrelation: Größere Gesamtwohnflächen führen typischerweise zu höheren Verkaufspreisen.

Die Korrelationen mit dem Verkaufspreis unterstützen diese Beobachtungen. Die Gesamtwohnfläche zeigt die stärkste Korrelation mit dem Verkaufspreis, gefolgt von der Erdgeschossfläche und der Kellerfläche.

## Nächste Schritte

Möchtest du weitere spezifische Analysen oder Visualisierungen zu diesem Datensatz durchführen? Beispielsweise könnten wir eine einfache lineare Regression zur Vorhersage der Verkaufspreise basierend auf einer oder mehreren dieser Flächenangaben durchführen. [→]



**Führe eine einfache lineare Regression zur Vorhersage der Verkaufspreise basierend auf einer oder mehreren der Flächenangaben durch.**



Super, wir werden eine lineare Regression durchführen, um den Verkaufspreis basierend auf der Gesamtwohnfläche (Gr Liv Area) vorherzusagen. Dies ist die Variable, die die höchste Korrelation mit dem Verkaufspreis aufweist.

## Schritt 1: Daten vorbereiten

Wir teilen die Daten in Trainings- und Testdaten auf, um die Modelleistung zu bewerten.

## Schritt 2: Modell erstellen und trainieren

Wir erstellen ein lineares Regressionsmodell und trainieren es mit den Trainingsdaten.

## Schritt 3: Modell bewerten

Wir bewerten die Leistung des Modells anhand der Testdaten und interpretieren die Ergebnisse.

Lass uns beginnen! 😊

### Schritt 1: Daten vorbereiten

Wir teilen die Daten in Trainings- und Testdaten auf.

Analysiert

```
python
from sklearn.model_selection import train_test_split

# Merkmale und Zielvariable definieren
X = data[['Gr Liv Area']]
y = data['SalePrice']

# Daten in Trainings- und Testdaten aufteilen
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

X_train.shape, X_test.shape, y_train.shape, y_test.shape

Ergebnis
((2344, 1), (586, 1), (2344,), (586,))
```

Die Daten wurden erfolgreich in Trainings- und Testdatensätze aufgeteilt. Wir haben 2344 Datenpunkte für das Training und 586 Datenpunkte für den Test.

## Schritt 2: Modell erstellen und trainieren

Jetzt erstellen wir das lineare Regressionsmodell und trainieren es mit den Trainingsdaten. [→]

Analysiert

python

Details immer anzeigen

Code kopieren

```
from sklearn.linear_model import LinearRegression

# Lineares Regressionsmodell erstellen
model = LinearRegression()

# Modell mit den Trainingsdaten trainieren
model.fit(X_train, y_train)

# Koeffizienten und Achsenabschnitt des Modells anzeigen
beta_1 = model.coef_[0]
beta_0 = model.intercept_

beta_1, beta_0
```

Ergebnis

(106.72915508271505, 19250.559555372573)

Das trainierte lineare Regressionsmodell hat folgende Gleichung:

Verkaufspreis = 19250.56 + 106.73 × Gesamtwohnfläche

Das bedeutet, dass für jede zusätzliche Quadratfuß Wohnfläche der Verkaufspreis um etwa 106.73 USD steigt.

### Schritt 3: Modell bewerten

Jetzt bewerten wir die Leistung des Modells anhand der Testdaten. Dazu berechnen wir den R<sup>2</sup>-Wert (Bestimmtheitsmaß) und den mittleren quadratischen Fehler (MSE).

Lass uns das durchführen. [→]

Analysiert

python

Details immer anzeigen

Code kopieren

```
from sklearn.metrics import mean_squared_error, r2_score

# Vorhersagen mit dem Testdatensatz
y_pred = model.predict(X_test)

# Modellbewertung
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

mse, r2
```

Ergebnis

(3821184066.2726417, 0.5233974153691151)

### Modellbewertung

#### 1. Mittlerer quadratischer Fehler (MSE): 3,821,184,066

- Dieser Wert gibt den durchschnittlichen quadratischen Fehler zwischen den vorhergesagten und den tatsächlichen Verkaufspreisen an. Ein niedrigerer Wert bedeutet eine bessere Vorhersagequalität.

#### 2. Bestimmtheitsmaß (R<sup>2</sup>): 0.523

- Dieser Wert gibt an, wie gut das Modell die Variabilität der Daten erklärt. Ein Wert von 0.523 bedeutet, dass etwa 52.3% der Variabilität im Verkaufspreis durch die Gesamtwohnfläche erklärt werden kann.

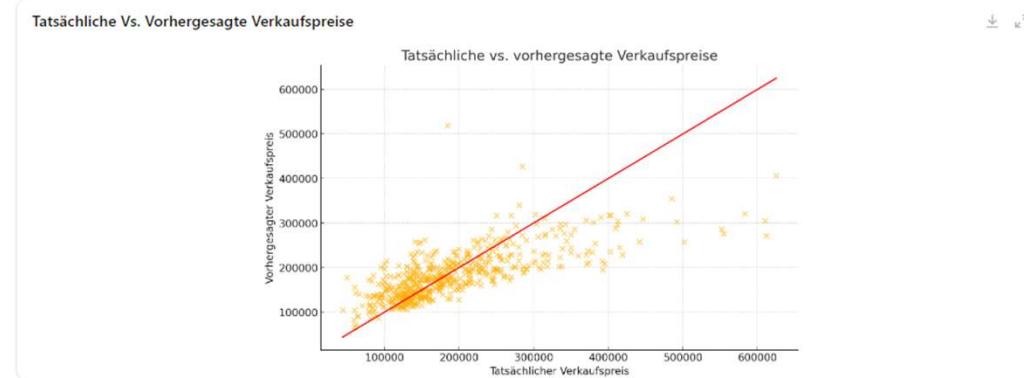
## Visualisierung der Vorhersagen

Lass uns die tatsächlichen Verkaufspreise und die vorhergesagten Verkaufspreise grafisch darstellen, um die Modellleistung visuell zu überprüfen. [→]

Analysiert

```
python
# Scatterplot der tatsächlichen vs. vorhergesagten Verkaufspreise
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red') # Diagonale Linie
plt.title('Tatsächliche vs. vorhergesagte Verkaufspreise')
plt.xlabel('Tatsächlicher Verkaufspreis')
plt.ylabel('Vorhergesagter Verkaufspreis')
plt.grid(True)
plt.show()
```

Details immer anzeigen  |  Code kopieren



## Interpretation der Visualisierung

Die Punkte im Scatterplot repräsentieren die tatsächlichen und vorhergesagten Verkaufspreise. Die rote Linie stellt die ideale Linie dar, bei der die vorhergesagten Werte genau den tatsächlichen Werten entsprechen würden.

Die Punkte sollten sich möglichst nahe an dieser Linie befinden, was auf eine gute Modellleistung hinweist. In unserem Fall sind die Punkte relativ gut verteilt, aber es gibt einige Abweichungen, was auf die eingeschränkte Modellgenauigkeit hinweist. Der R<sup>2</sup>-Wert von 0.523 zeigt, dass unser Modell die Variabilität im Verkaufspreis moderat gut erklärt.

## Zusammenfassung

- Modellgleichung: Verkaufspreis = 19250.56 + 106.73 × Gesamtwohnfläche
- Modellbewertung:
  - Mittlerer quadratischer Fehler (MSE): 3,821,184,066
  - Bestimmtheitsmaß (R<sup>2</sup>): 0.523
- Visualisierung: Die Vorhersagen sind moderat gut, aber es gibt Raum für Verbesserungen.

Möchtest du noch tiefer in die Analyse einsteigen, z.B. andere Variablen einbeziehen oder andere Modelle ausprobieren? 😊 [→]

Möchtest du noch tiefer in die Analyse einsteigen, z.B. andere Variablen einbeziehen oder andere Modelle ausprobieren? 😊 [→]



Inkludiere mehr Variablen im Modell.

Untersuche nichtlineare Regressionsmodelle.

✉️ Sende eine Nachricht an Data Science Tutor v6

**Inkludiere mehr Variablen im Modell**

Inkludiere mehr Variablen im Modell.



Super, wir werden mehr Variablen in das Modell aufnehmen, um die Vorhersagegenauigkeit zu verbessern. Lass uns die folgenden Variablen verwenden:

- Gesamtfläche des Kellers (Total Bsmt SF)
- Fläche des Erdgeschosses (1st Flr SF)
- Fläche des Obergeschosses (2nd Flr SF)
- Gesamtwohnfläche (Gr Liv Area)

### Schritt 1: Daten vorbereiten

Wir werden die oben genannten Variablen in das Modell aufnehmen und dann das Modell trainieren und bewerten.

Analysefehler

```
python
# Details immer anzeigen  | 
# Merkmale und Zielvariable definieren (mit mehr Variablen)
X = data[['Total Bsmt SF', '1st Flr SF', '2nd Flr SF', 'Gr Liv Area']]
y = data['SalePrice']

# Daten in Trainings- und Testdaten aufteilen
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Neues Modell erstellen und trainieren
model_multi = LinearRegression()
model_multi.fit(X_train, y_train)

# Koeffizienten und Achsenabschnitt des neuen Modells anzeigen
coefficients = model_multi.coef_
intercept = model_multi.intercept_

coefficients, intercept
125      NOT_NAN_NAN
126      and estimator_name
(...)

130      # Improve the error message on how to handle missing values in
131      # scikit-learn.
132      msg_err += (
133          f"\n{estimator_name} does not accept missing values"
134          " encoded as NaN natively. For supervised learning, you might want"
(...)
144          "#estimators-that-handle-nan-values"
145      )
--> 146      raise ValueError(msg_err)
148 # for object dtype data, we only check for NaNs (GH-13254)
149 elif X.dtype == np.dtype("object") and not allow_nan:
ValueError: Input X contains NaN.
LinearRegression does not accept missing values encoded as NaN natively. For supervised learning, you might want to consider sklearn.ensemble.HistGradientBoostingClassifier and Regressor which accept
```

Es gibt fehlende Werte in den Daten, die von dem linearen Regressionsmodell nicht direkt verarbeitet werden können. Wir müssen die fehlenden Werte im Datensatz entweder entfernen oder durch sinnvolle Werte ersetzen (Imputation).

### Schritt 1: Fehlende Werte behandeln

Ich werde die fehlenden Werte durch den Durchschnitt der jeweiligen Spalte ersetzen.

## Schritt 2: Modell neu erstellen und trainieren

Nach der Behandlung der fehlenden Werte werden wir das Modell erneut erstellen und trainieren. [\[>\]](#)

Analysiert

```
python
from sklearn.impute import SimpleImputer

# Imputer erstellen, um fehlende Werte durch den Mittelwert zu ersetzen
imputer = SimpleImputer(strategy='mean')

# Fehlende Werte in den Trainings- und Testdaten ersetzen
X_train_imputed = imputer.fit_transform(X_train)
X_test_imputed = imputer.transform(X_test)

# Neues Modell erstellen und trainieren
model_multi = LinearRegression()
model_multi.fit(X_train_imputed, y_train)

# Koeffizienten und Achsenabschnitt des neuen Modells anzeigen
coefficients = model_multi.coef_
intercept = model_multi.intercept_

coefficients, intercept

Ergebnis
(array([ 64.40899552, 139.06858351, 137.50423972, -54.43866522]),
 -14068.09333632767)
```

Das trainierte lineare Regressionsmodell mit mehreren Variablen hat folgende Gleichung:

$$\text{Verkaufspreis} = -14068.09 + 64.41 \times \text{Total Bsmt SF} + 139.07 \times \text{1st Flr SF} + 137.50 \times \text{2nd Flr SF} - 54.44 \times \text{Gr Liv Area}$$

### Interpretation der Koeffizienten

- **Total Bsmt SF (Kellerfläche):** Für jede zusätzliche Quadratfuß Kellerfläche steigt der Verkaufspreis um etwa 64.41 USD.
- **1st Flr SF (Erdgeschossfläche):** Für jede zusätzliche Quadratfuß Erdgeschossfläche steigt der Verkaufspreis um etwa 139.07 USD.
- **2nd Flr SF (Obergeschossfläche):** Für jede zusätzliche Quadratfuß Obergeschossfläche steigt der Verkaufspreis um etwa 137.50 USD.
- **Gr Liv Area (Gesamtwohnfläche):** Interessanterweise zeigt diese Variable einen negativen Koeffizienten, was auf mögliche Multikollinearität hinweisen könnte. Das bedeutet, dass die Gesamtwohnfläche in Anwesenheit der anderen Variablen den Verkaufspreis negativ beeinflusst.

## Schritt 3: Modell bewerten

Wir bewerten nun die Leistung des neuen Modells anhand der Testdaten. [-]

Analysiert

python

Details immer anzeigen

Code kopieren

```
# Vorhersagen mit dem Testdatensatz
y_pred_multi = model_multi.predict(X_test_imputed)

# Modellbewertung
mse_multi = mean_squared_error(y_test, y_pred_multi)
r2_multi = r2_score(y_test, y_pred_multi)

mse_multi, r2_multi
```

Ergebnis

(2484081871.1410866, 0.6901693769032654)

## Modellbewertung

### 1. Mittlerer quadratischer Fehler (MSE): 2,484,081,871

- Dieser Wert gibt den durchschnittlichen quadratischen Fehler zwischen den vorhergesagten und den tatsächlichen Verkaufspreisen an. Der MSE ist im Vergleich zum vorherigen Modell niedriger, was auf eine bessere Vorhersagequalität hinweist.

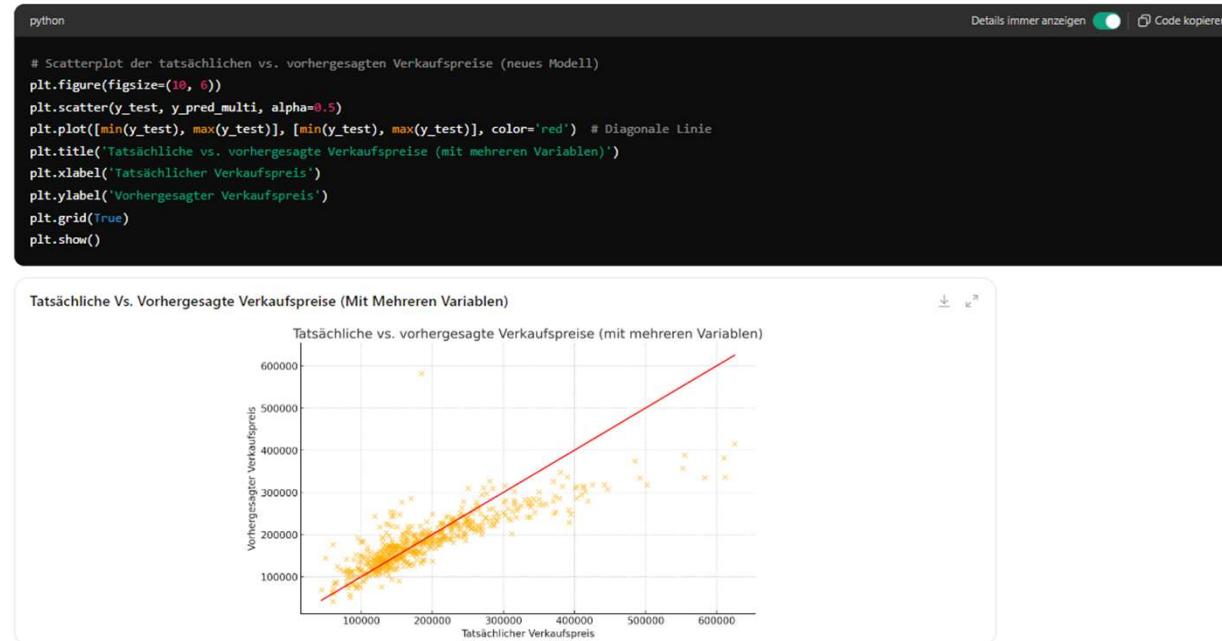
### 2. Bestimmtheitsmaß ( $R^2$ ): 0.690

- Dieser Wert gibt an, wie gut das Modell die Variabilität der Daten erklärt. Ein Wert von 0.690 bedeutet, dass etwa 69.0% der Variabilität im Verkaufspreis durch die ausgewählten Variablen erklärt werden kann. Dies ist eine Verbesserung gegenüber dem vorherigen Modell.

## Visualisierung der Vorhersagen

Lass uns die tatsächlichen Verkaufspreise und die vorhergesagten Verkaufspreise des neuen Modells grafisch darstellen. [→]

Analysiert



## Interpretation der Visualisierung

Die Punkte im Scatterplot repräsentieren die tatsächlichen und vorhergesagten Verkaufspreise des neuen Modells. Die rote Linie stellt die ideale Linie dar, bei der die vorhergesagten Werte genau den tatsächlichen Werten entsprechen würden.

Die Punkte liegen näher an der roten Linie im Vergleich zum vorherigen Modell, was auf eine verbesserte Vorhersageleistung hinweist.

## Zusammenfassung

- Modellgleichung:  $\text{Verkaufspreis} = -14068.09 + 64.41 \times \text{Total Bsmt SF} + 139.07 \times \text{1st Flr SF} + 137.50 \times \text{2nd Flr SF} - 54.44 \times \text{Gr Liv Area}$
- Modellbewertung:
  - Mittlerer quadratischer Fehler (MSE): 2,484,081,871
  - Bestimmtheitsmaß ( $R^2$ ): 0.690
- Visualisierung: Die Vorhersagen sind genauer als beim einfachen Modell, aber es gibt immer noch einige Abweichungen.

Möchtest du noch tiefer in die Analyse einsteigen oder ein anderes Modell ausprobieren? 😊 [→]



Versuch ein Random Forest Modell.

Wie wäre es mit Feature Scaling?

## Versuche Random Forrest für bessere Vorhersagen

Versuche Random Forest für bessere Vorhersagen.

Ein Random Forest ist ein leistungsstarkes Ensemble-Modell, das mehrere Entscheidungsbäume kombiniert, um die Vorhersagegenauigkeit zu erhöhen und Überanpassung zu reduzieren. Lassen Sie uns ein Random Forest-Modell erstellen und die Ergebnisse bewerten.

### Schritte:

1. Datenaufteilung: Aufteilung der Daten in Trainings- und Testdatensätze.
2. Modellbildung: Erstellen und Trainieren eines Random Forest-Modells.
3. Modellbewertung: Evaluierung des Modells anhand von MSE und R<sup>2</sup>.

Lassen Sie uns beginnen.

Die Ergebnisse des Random Forest-Modells sind wie folgt:

- Mean Squared Error (MSE): 2,507,867,952.74
- R<sup>2</sup>-Wert: 0.687

### Zusammenfassung

Das Random Forest-Modell zeigt eine Verbesserung gegenüber den linearen Regressionsmodellen und der Ridge-Regression. Der R<sup>2</sup>-Wert von 0.687 zeigt, dass das Modell etwa 68.7% der Variabilität im Verkaufspreis erklären kann.

## Prompt: Versuche Random Forrest für bessere Vorhersagen

# Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume

Kategorie	Methode	Anwendungsgebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
	Klassifikation (Decision Trees)	Finanzwesen	Kreditwürdigkeitsprüfung	Entscheidungen werden basierend auf Datenattributen in einer Baumstruktur getroffen. Der Baum teilt die Daten in immer kleinere Gruppen auf.	Entscheidungsprozess wie ein Entscheidungsbau im Alltag.	Leicht zu verstehen und zu visualisieren	Neigung zu Überanpassung (Overfitting)



Überwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen: Klassifikation

## Klassifikation bei Zahlungen

- Überprüfung von Transaktionen auf verdächtige Aktivitäten bei der Nutzung von Kreditkarten
- Identifikation ungewöhnlicher Käufe oder untypischer Orte für den Karteninhaber
- Gute Datensätze durch schnelle Meldungen von Kreditkarteninhabern bei unautorisierten Umsätzen
- Abwägung von Fehlern:
  - Falscher Alarm: peinlich, führt aber nur zu Unannehmlichkeiten beim Bezahlen
  - Verpasster Betrug: Konto könnte geleert werden, Erstattungsprobleme mit dem Kreditkarteninstitut
- Fraud-Systeme:
  - Regelbasierte Systeme: Alarm bei bestimmten Summen oder Shop-Kategorien
  - Betrügerische Anpassungen: Notwendigkeit, neue Variablen zur Klassifikation zu finden

# Grundlegende Methoden Maschinelles Lernen: Klassifikation

## Klassifikation bei Kreditwürdigkeit

- Überprüfung der Kreditwürdigkeit bei Transaktionen wie Rechnungskauf oder Ratenzahlung
- Nutzung historischer Daten für das Training von Algorithmen (erfolgreiche und problematische Transaktionen)
- Abfrage von Auskunfteien (z.B. Schufa) zur Einschätzung der Kreditwürdigkeit
- Abwägung von Fehlern:
  - Kreditverweigerung: Kunde kauft eventuell nicht
  - Kreditvergabe ohne Rückzahlung: finanzieller Verlust
- Grenzwertbestimmung für Kreditvergabe: Kompromiss zwischen Geschäftsabschluss und Verhinderung von Kreditausfall

# Grundlegende Methoden Maschinelles Lernen: Klassifikation

## Klassifikation bei Versicherungsbetrug

- Aufdeckung unberechtigter Schadensanträge durch Algorithmen
- Vorgetäuschte Schäden oder falsche Darstellung des Schadenshergangs
- Finanzielle Verluste für die Versicherungswirtschaft und höhere Prämien für alle
- Automatisierte Prüfung von Schadensanträgen auf Verdachtsmomente
- Training der Algorithmen mit früheren Betrugsfällen

## Grundlegende Methoden Maschinelles Lernen: Klassifikation

### **Kundendatenbank und Kündigungswahrscheinlichkeit:**

- Algorithmus überprüft vergangene Kündigungen zur Vorhersage zukünftiger Kündigungen
- Identifikation von Mustern bei Kunden, die gekündigt haben (z.B. Häufung von Beschwerden)
- Proaktive Ansprache gefährdeter Kunden mit Gesprächen zur Zufriedenheit und speziellen Angeboten
- Klassifikation von Kunden für neue Produkte oder Services

# Grundlegende Methoden Maschinelles Lernen: Klassifikation

## **Klassifikation von Muttermalen:**

- Algorithmus analysiert Bilder von Muttermalen zur Diagnose von Melanomen
- Fehler 2. Art (Übersehen von Melanomen) besonders kritisch
- Unterstützung von Hautärzten durch Systeme wie die der Firma Fotofinder
- Umwandlung von Bildmaterial in Zahlen zur Analyse

# Grundlegende Methoden Maschinelles Lernen: Klassifikation mit Entscheidungsbäumen

## Funktionsweise der Decision Trees:

- Ein Entscheidungsbaum wächst von oben nach unten.
- An der Wurzel werden die Daten in zwei Teile geteilt, um den größtmöglichen Unterschied bei der Zielvariablen zu erzielen.
- Beispiel Titanic-Datensatz:
  - Variable wird genutzt, die die Daten am besten in Bezug auf die Zielvariable (überlebt/nicht überlebt) aufteilt.
  - Wahrscheinlich wird das Geschlecht die beste Variable sein.
- Schritt wird wiederholt, bis keine Verbesserung durch weiteres Aufteilen möglich ist.
- Ergebnis wird an den Blättern der Verzweigungen abgelesen.
- Vorgehensweise wird als rekursive Partitionierung bezeichnet.

## Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume

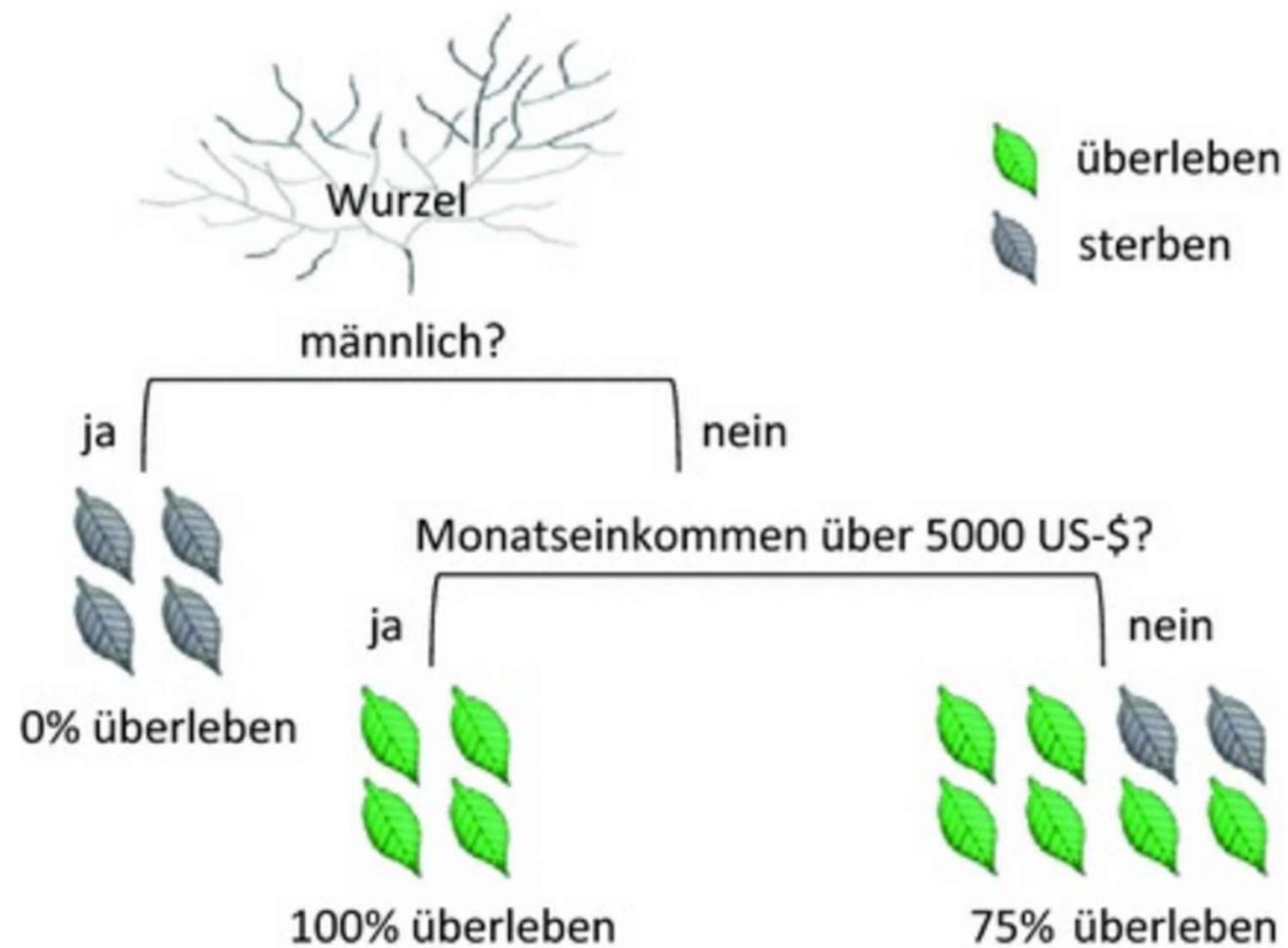
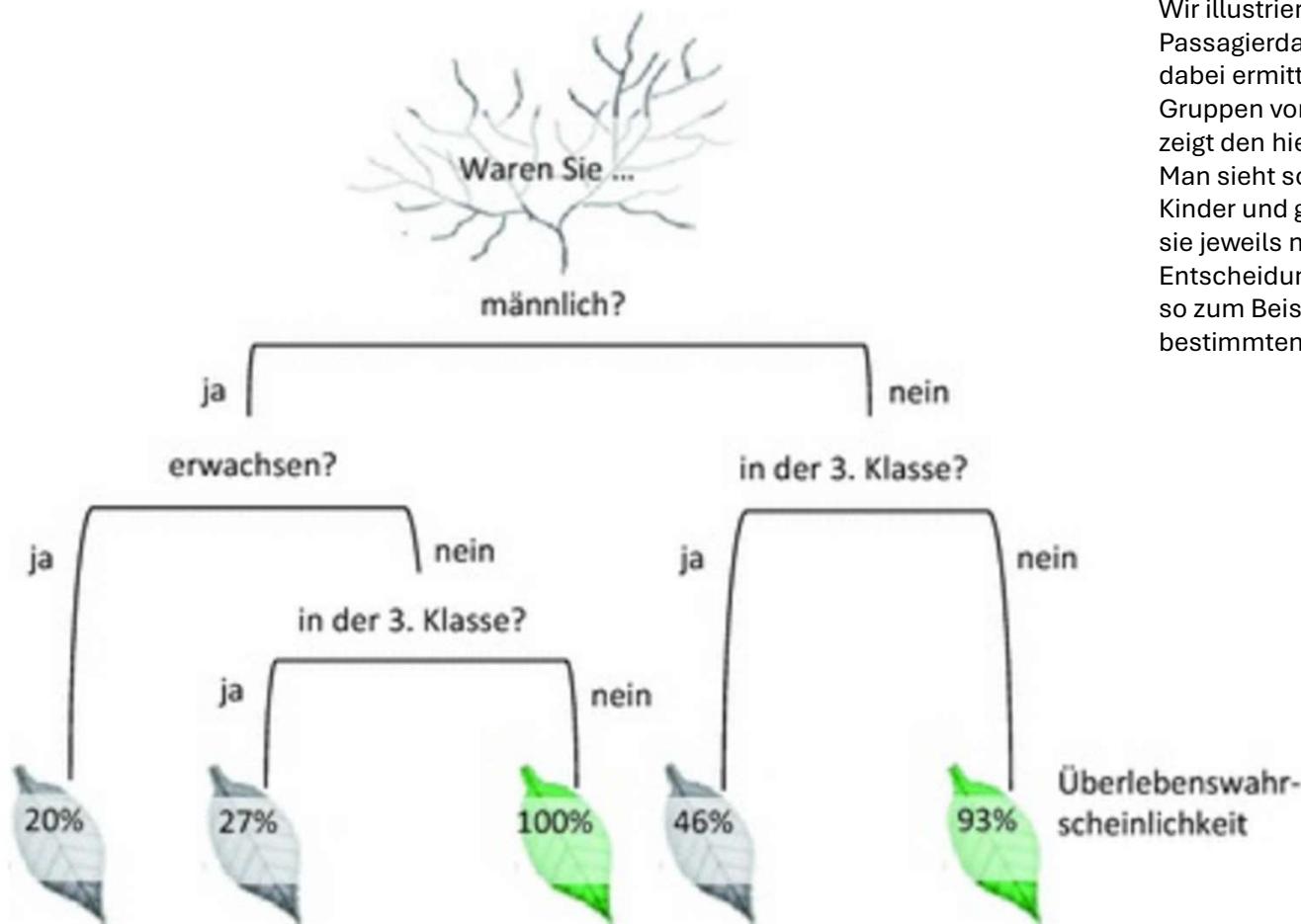


Abb. 9.1 Beispiel für einen Entscheidungsbaum

## Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume



Wir illustrieren die Verwendung von Entscheidungsbäumen anhand von Passagierdaten des 1912 gesunkenen Ozeandampfers Titanic. Es soll dabei ermittelt werden, mit welcher Wahrscheinlichkeit verschiedene Gruppen von Passagieren die Katastrophe überlebt haben. Abbildung zeigt den hierfür berechneten Entscheidungsbaum. Man sieht sofort, dass von den dargestellten Gruppen männliche Kinder und generell Frauen eine hohe Überlebenschance hatten, sofern sie jeweils nicht das Pech hatten, in der 3. Klasse zu reisen. Entscheidungsbäume haben ausgesprochen vielfältige Anwendungen, so zum Beispiel die Prognose von Heilungschancen nach einer bestimmten medizinischen Diagnose,

**Abb. 9.2** Entscheidungsbäum für die Überlebenswahrscheinlichkeit auf der Titanic

## Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume

- Entscheidungsbäume werden gerne eingesetzt, weil sie sehr leicht zu interpretieren sind.
- Frage ist , wie wir einen Entscheidungsbaum am besten konstruieren.

# Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume

## „Pflanzen“ eines Entscheidungsbaums:

- Einteilung aller Datenpunkte in zwei Gruppen, sodass ähnliche Datenpunkte beieinander liegen
- Weiteres binäres Aufsplitten dieser Gruppen
- Ergebnis: Unter jeder Verzweigung (Knoten) liegen weniger und homogener Datenpunkte

## Grundidee:

- Datenpunkte, die durch denselben Pfad von der Wurzel erreicht werden, ähneln sich wahrscheinlich sehr

## Rekursive Partitionierung:

- Wiederholtes Aufteilen zur Erzielung homogenerer Untergruppen
- Schritte:
  1. Finde die binäre Frage, welche die Datenpunkte am besten in zwei möglichst homogene Gruppen teilt
  2. Wiederhole Schritt 1 für jeden Knoten, bis ein Abbruchkriterium erfüllt ist

## Abbruchkriterien:

- Datenpunkte an jedem Blatt sind von derselben Kategorie oder haben denselben Wert
- Blätter enthalten weniger als fünf Datenpunkte
- Weitere Verzweigungen verbessern die Homogenität nicht weiter oder eine vorgegebene Mindesthomogenität ist erreicht

## Eigenschaften der rekursiven Partitionierung:

- Benutzt nur aufschlussreiche binäre Fragen, nicht-signifikante Variablen beeinträchtigen das Ergebnis nicht
- Datenpunkte tendieren dazu, sich um zentrale Werte zu versammeln
- Entscheidungsbäume sind robust gegen Ausreißer.

# Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäum Titanic Datensatz

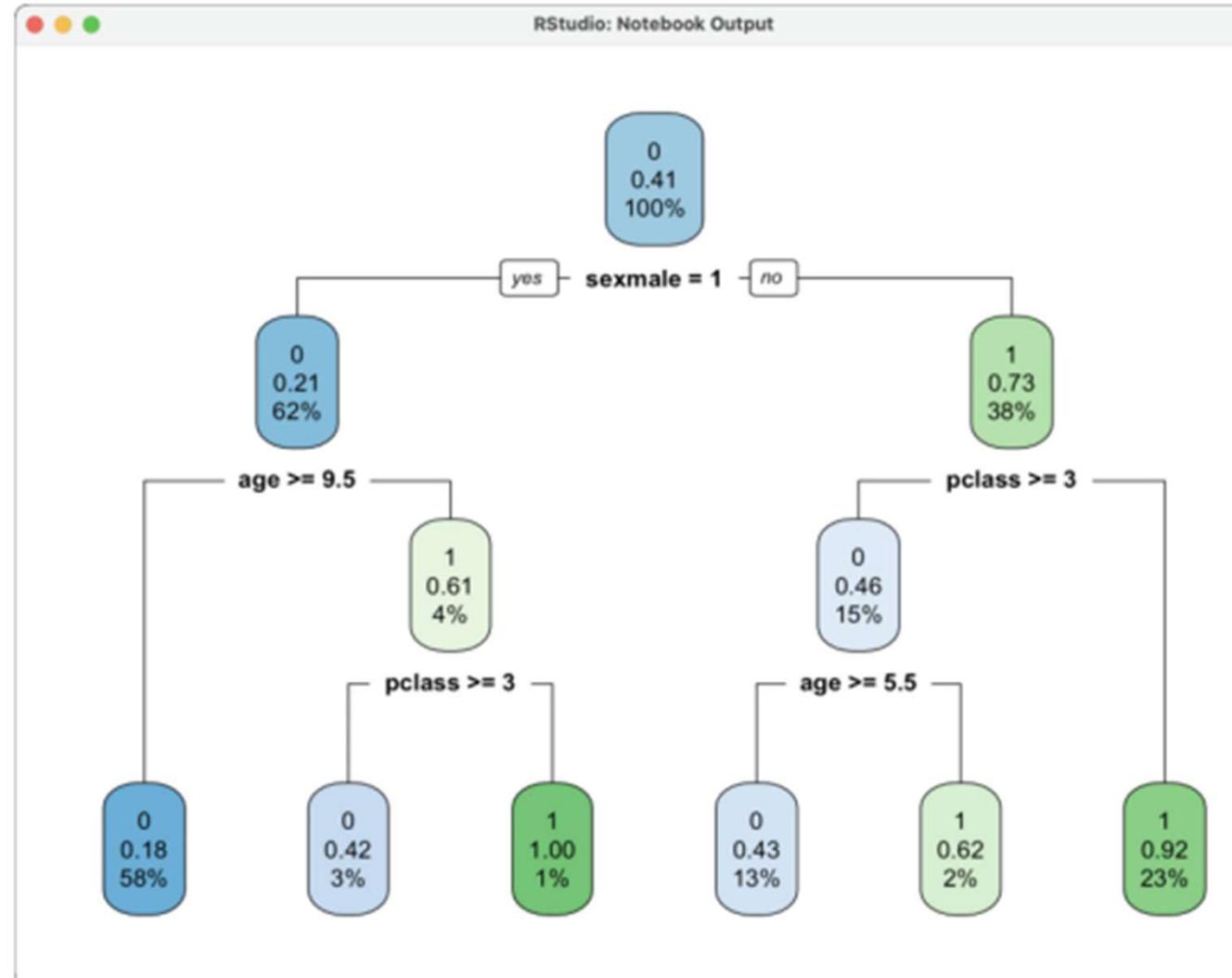


Abbildung 8.5 Das Ergebnis unseres ersten Decision Trees

Quelle: Tom Alby, Data Science in der Praxis, 2022

# Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäum Datensatz Titanic

• **Regelwerk:** Der Decision Tree dient als nachvollziehbares Regelwerk, das das Vertrauen der Anwender stärkt.

• **Variable Geschlecht:** Geschlecht ist die Variable an der Wurzel, die den größten Unterschied macht.

• **Knoteninformationen:**

- Vorhergesagte Klasse
- Vorhergesagte Überlebenswahrscheinlichkeit
- Prozentanteil der Beobachtungen im Knoten

**Detaillierte Verzweigungen:**

• **Wurzelknoten:**

- Überlebenswahrscheinlichkeit: 41%
- 100% der Beobachtungen

• **Geschlecht (rechts: Frauen):**

- Überlebenswahrscheinlichkeit: 73%
- 38% der Beobachtungen
- **Passagierklasse < 3:**
  - Überlebenswahrscheinlichkeit: 92%
- **Passagierklasse = 3:**
  - Überlebenswahrscheinlichkeit: 46%
  - **Alter ≥ 5,5 Jahre:**
    - Überlebenswahrscheinlichkeit: 43%
  - **Alter < 5,5 Jahre:**
    - Überlebenswahrscheinlichkeit: 62%

• **Geschlecht (links: Männer):**

- Überlebenswahrscheinlichkeit: 21%
- 62% der Beobachtungen
- **Alter ≥ 9,5 Jahre:**
  - Überlebenswahrscheinlichkeit: 18%
  - 58% der Beobachtungen
- **Alter < 9,5 Jahre:**
  - Überlebenswahrscheinlichkeit: 61%
  - 4% der Beobachtungen

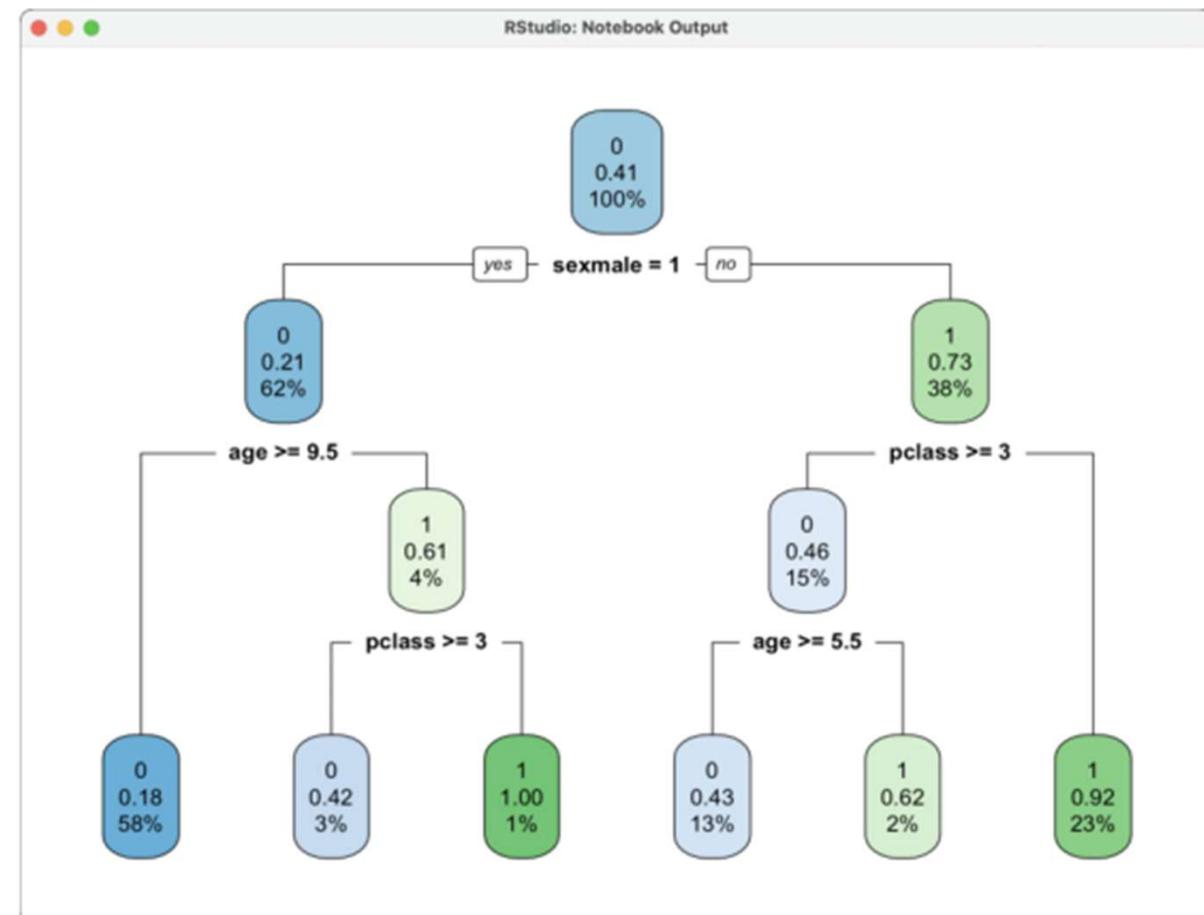


Abbildung 8.5 Das Ergebnis unseres ersten Decision Trees

# Grundlegende Methoden Maschinelles Lernen: Entscheidungsbäume - Grenzen

**Stärken von Entscheidungsbäumen:** Leicht zu interpretieren

**Schwächen von Entscheidungsbäumen:**

- **Instabilität:**

- Kleine Änderungen der Daten können die Aufteilung stark verändern
- Anfällig für Überanpassungen

**Ungenauigkeit:**

- Beste binäre Frage für die erste Aufteilung führt nicht immer zu den genauesten Vorhersagen
- Weniger effiziente erste Aufteilungen können manchmal zu besseren Vorhersagen führen

**Methoden zur Verbesserung:**

- **Diversifikation von Entscheidungsbäumen:**

- Kombinieren von Vorhersagen unterschiedlicher Bäume zur Erhöhung der Stabilität und Genauigkeit

**Methoden zur Diversifikation:**

- **Random Forest:**

- Erzeugung vieler unterschiedlicher Bäume durch zufällige Kombinationen von binären Fragen
- Aggregation der Vorhersagen (z.B. Mittelwert)

- **Gradient Boosting:**

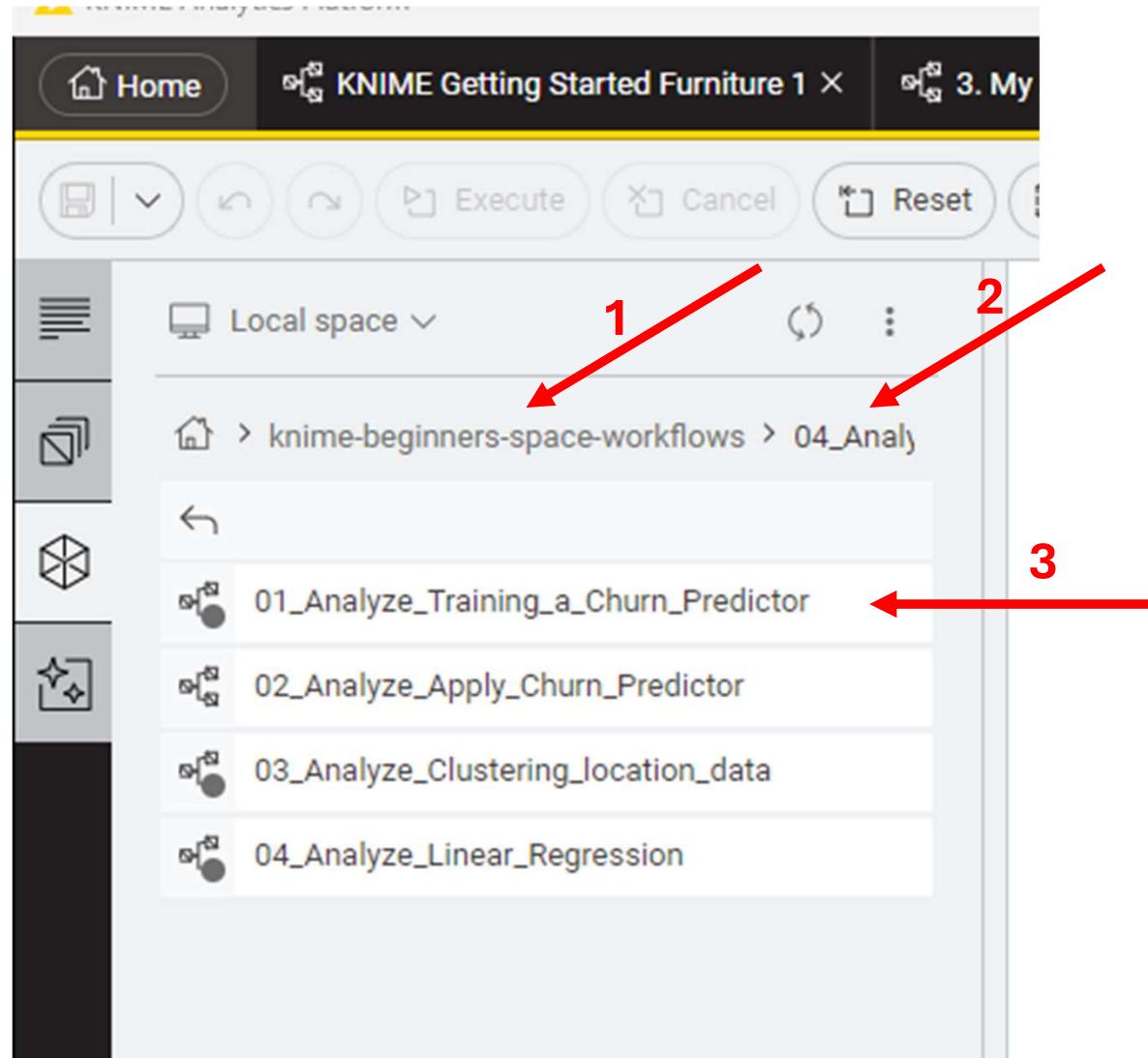
- Auswahl von binären Fragen zur schrittweisen Verbesserung der Vorhersagegenauigkeit
- Verwendung eines gewichteten Mittels der Vorhersagen aller Bäume

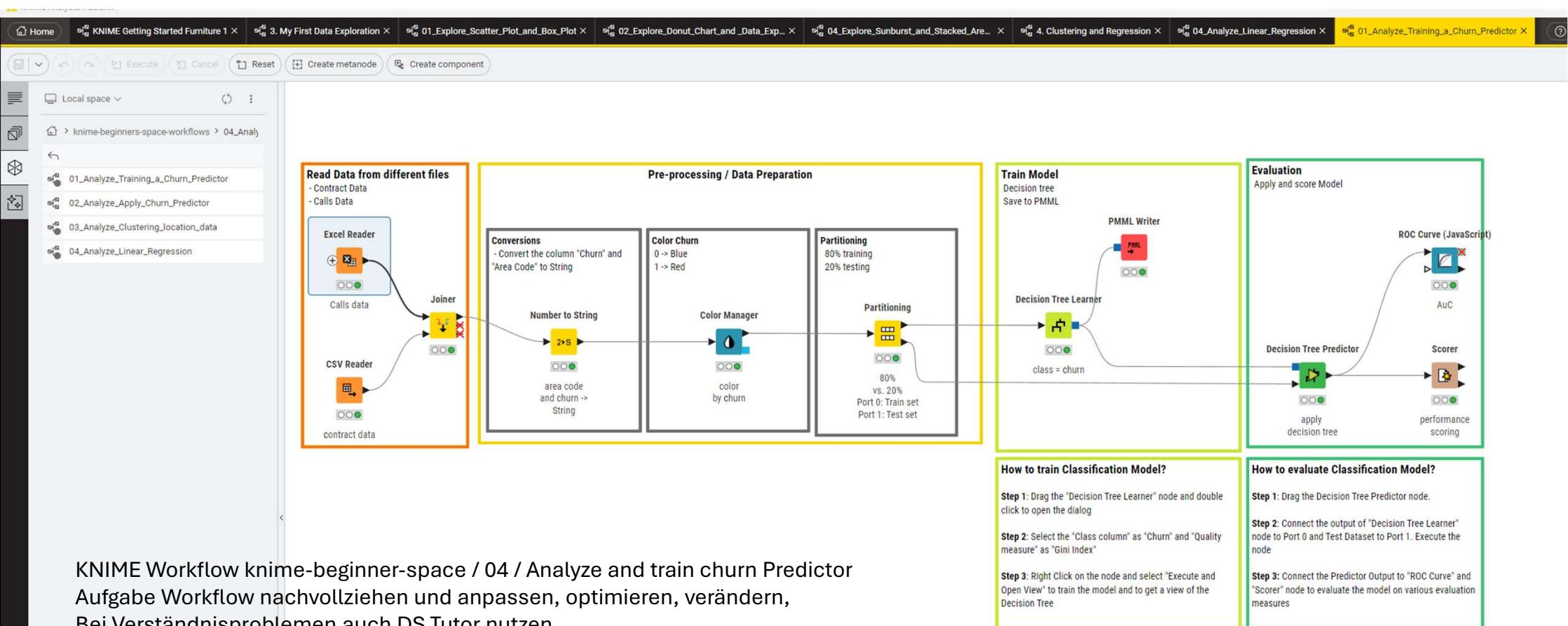
**Vor- und Nachteile der Methoden:**

- **Random Forest und Gradient Boosting:**

- Tendenziell bessere Vorhersagen
- Höhere Komplexität, schwerer zu visualisieren („Blackboxes“)

## KNIME Workflow Kundenabwanderung: Entscheidungsbaum





KNIME Workflow knime-beginner-space / 04 / Analyze and train churn Predictor  
Aufgabe Workflow nachvollziehen und anpassen, optimieren, verändern,  
Bei Verständnisproblemen auch DS Tutor nutzen

1: File Table Flow Variables

Rows: 3333 | Columns: 16

#	RowID	VMail Messa...	Day Mins	Eve Mins	Night Mins	Intl Mins	CustServ Calls	Day Calls	Day Charge	Eve Calls	Eve Charge	Night Calls	Night Charge	Intl Calls	Intl Charge	Area C
1	Row0	25	265.1	197.4	244.7	10	1	110	45.07	99	16.78	91	11.01	3	2.7	415
2	Row1	26	161.6	195.5	254.4	13.7	1	123	27.47	103	16.62	103	11.45	3	3.7	415

# Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest

Kategorie	Methode	Anwendungsgebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
	Klassifikation (Random Forest)	Zahlungsverkehr	Betrugs-erkennung	Kombination vieler Entscheidungsbäume, um Vorhersagen zu verbessern. Jeder Baum trägt zur finalen Entscheidung bei.	Abstimmung einer großen Gruppe von Experten.	Hohe Genauigkeit, reduziert Überanpassung	Komplex, schwer zu interpretieren



Überwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest

## Weisheit der Crowd:

- Kann oft falsch richtig sein
- Obwohl es Intuition zuwiderläuft, ist dies bei einigen der besten Vorhersagemodele möglich und zu erwarten

## Kernidee:

- Viele falsche Vorhersagen, aber nur eine korrekte
- Kombination von Modellen mit unterschiedlichen Stärken und Schwächen
- Zutreffende Vorhersagen bestärken sich gegenseitig
- Falsche Vorhersagen heben sich gegenseitig auf

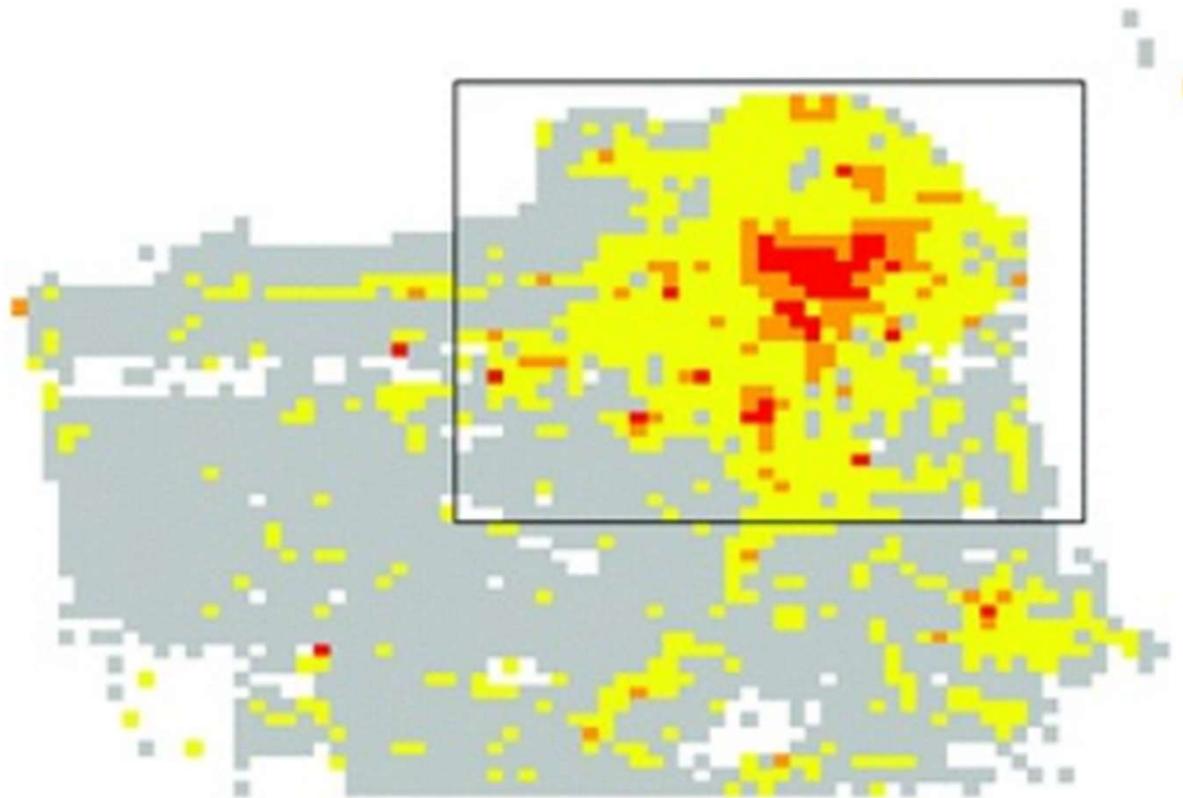
## Ensembling (Ensemble-Lernen):

- Verbesserung der Vorhersagegenauigkeit durch Kombination von Modellen
- Beispiel: Random Forest
  - Kombination von Entscheidungsbäumen
  - Random Forest arbeitet besser als einzelne Entscheidungsbäume

## Experiment:

- Erzeugung von 1000 möglichen Entscheidungsbäumen
- Vorhersage der Häufigkeit von Verbrechen in einer US-amerikanischen Großstadt
- Vergleich der Vorhersagegenauigkeit der einzelnen Entscheidungsbäume mit einem Random Forest

## Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest



**Abb. 10.1** Eine sogenannte Heatmap der Verbrechenshäufigkeit in San Francisco; grau: sehr selten, gelb: selten, orange: mäßig oft und rot: hoch

# Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest

## **Verbrechenvorhersage:**

- Nutzung offener Daten des San Francisco Police Department (2014-2016)
- Daten enthalten Informationen über Ort, Datum und Schwere von Verbrechen
- Hinzufügen lokaler Wetterdaten (Tagestemperaturen und Niederschläge)

## **Modellannahme:**

- Begrenzte Personalressourcen der Polizei
- Modell priorisiert die 30% der Gebiete mit den höchsten Wahrscheinlichkeiten für tägliche Gewaltverbrechen

## **Analyseergebnisse:**

- Verbrechen konzentrieren sich vor allem im Nordosten der Stadt
- Region in kleinere Rechtecke ( $260\text{ m} \times 220\text{ m}$ ) unterteilt

## Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest

### **Vorhersagemodell:**

- Erzeugung von 1000 möglichen Entscheidungsbäumen basierend auf Verbrechenszahlen & Wetterdaten
- Zusammenfassung der Bäume zu einem Random Forest
- Training der Modelle mit Daten von 2014 und 2015
- Testen der Genauigkeit mit Daten von 2016 (Januar bis August)

### **Vorhersagegenauigkeit:**

- Random-Forest-Modell sagte 72% aller Gewaltverbrechen korrekt voraus
- Durchschnittliche Vorhersagegenauigkeit der 1000 Einzelbäume: 67%
- Nur 12 von 1000 Entscheidungsbäumen hatten eine bessere Genauigkeit als der Random Forest

### **Beispielvorhersagen:**

- Polizei sollte Ressourcen in rot markierten Bereichen konzentrieren
- Modell konnte Verbrechen auch in nicht-roten Bereichen vorhersagen
- Beispiel: korrekt vorhergesagtes Verbrechen in einem grauen Bezirk am Tag 4

## Grundlegende Methoden Maschinelles Lernen Klassifikation – Random Forest



Quelle: Data Science – Was ist das eigentlich?! Algorithmen des maschinellen Lernens verständlich erklärt, 2018

## Grundlegende Methoden Maschinelles Lernen: Ensemble

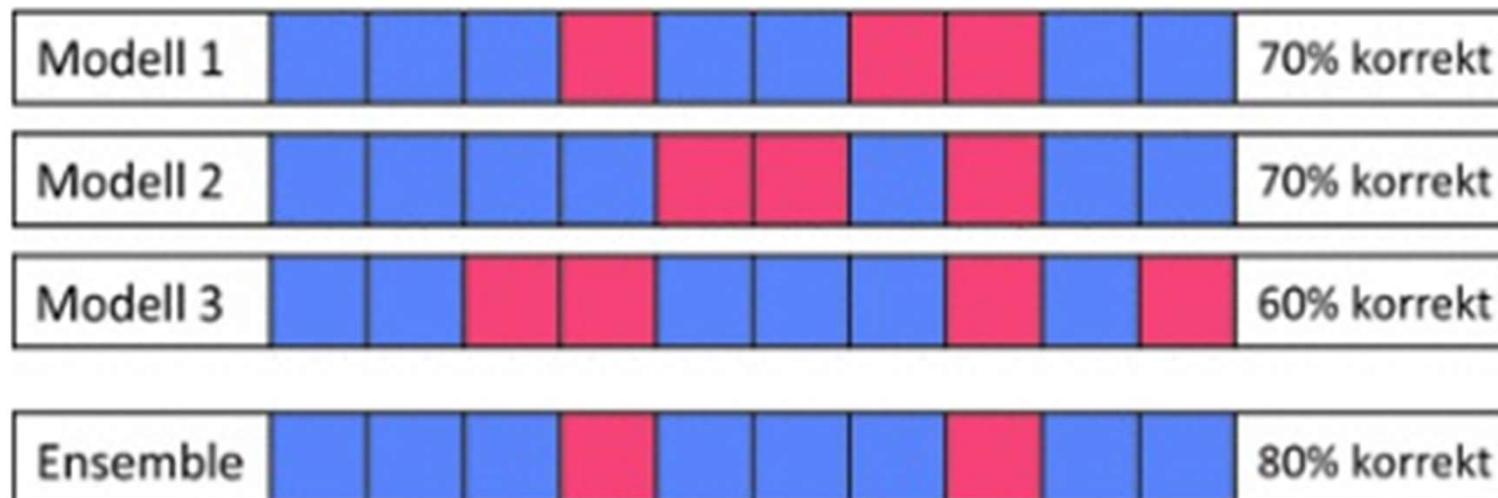
### **Ensembles:**

- Ein Random Forest ist ein Ensemble von Entscheidungsbäumen.
- Ein Ensemble kombiniert die Vorhersagen vieler Einzelmodelle.
- Kombinationsmethoden: Mehrheitsentscheid oder Mittelwertbildung.

### **Vorteile von Ensembles:**

- Genauere Vorhersagen durch Mehrheitsentscheid.
- Korrekte Vorhersagen bestärken sich gegenseitig.
- Fehler heben sich gegenseitig auf.
- Voraussetzung: Modelle müssen unkorreliert sein, d.h., sie dürfen nicht alle dieselben Fehler machen.

## Grundlegende Methoden Maschinelles Lernen: Ensemble



**Abb. 10.5** Drei Beispielmodelle für die Vorhersage von zehn Antworten auf eine binäre Frage

Drei Beispielmodelle für die Vorhersage von zehn Antworten auf eine binäre Frage (blau oder rot).

- Korrekte Vorhersage war in allen zehn Fällen blau.
- Einzelmodelle hatten Vorhersagegenauigkeiten von 70%, 70% und 60%.
- Ensemble aus diesen drei Modellen hatte eine höhere Vorhersagegenauigkeit von 80% durch b Mehrheitsentscheid.

# Grundlegende Methoden Maschinelles Lernen: Unüberwachtes Lernen

Kategorie	Methode	Anwendungs-gebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
Unüberwachtes Lernen	Unsupervised Learning	Einzelhandel	Kunden-segmentierung	Modell lernt selbstständig Muster und Strukturen in den Daten. Es wird nicht mit gekennzeichneten Daten trainiert.	Entdecker erkundet unbekanntes Land ohne Karte.	Erkennt verborgene Muster ohne Vorwissen	Ergebnisse können schwer interpretierbar sein
	Clustering (Hierarchisch)	Marktsegmentierung	Markt-segmentierung	Daten werden in eine Baumstruktur von Clustern gruppiert. Jede Gruppe wird weiter unterteilt.	Sortieren von Dokumenten in immer kleinere Ordner.	Identifiziert hierarchische Beziehungen zwischen Daten	Rechenintensiv, schwer zu skalieren
	Clustering (k-Means)	Zielgruppen analyse	Zielgruppen-segmentierung	Daten werden in k Cluster aufgeteilt. Jeder Datenpunkt wird dem nächsten Zentrum zugewiesen.	Gruppierung von Menschen auf Basis von Ähnlichkeiten.	Schnell, einfach zu implementieren	Anzahl der Cluster muss vorab festgelegt werden



Unüberwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen: Unüberwachtes Lernen

Kategorie	Methode	Anwendungs-gebiet	Beispiel aus der Wirtschaft	Funktionsweise	Analogie für intuitives Verständnis	Vorteile	Nachteile
	Clustering (k-Means)	Zielgruppenanalyse	Zielgruppen-segmentierung	Daten werden in k Cluster aufgeteilt. Jeder Datenpunkt wird dem nächsten Zentrum zugewiesen.	Gruppierung von Menschen auf Basis von Ähnlichkeiten.	Schnell, einfach zu implementieren	Anzahl der Cluster muss vorab festgelegt werden



Unüberwachtes Lernen

# Grundlegende Methoden Maschinelles Lernen: Unüberwachtes Lernen - Clustering k-Means

## **Clustering:**

- Ermittelt gemeinsame Vorlieben oder Eigenschaften von Personen
- Teilt diese in Gruppen für gezielte Werbekampagnen

## **Herausforderungen:**

- Schwierigkeit, relevante Kundengruppen zu bilden
- Unklarheit, wie Kunden eingeteilt werden sollten
- Ungewissheit über die Anzahl der Gruppen

## **k-Means-Clustering:**

- Verfahren zur Bildung von Clustern
- k gibt die Anzahl der Cluster an

## **Datenanforderungen:**

- Notwendigkeit quantifizierbarer Informationen über die Kunden
- Beispiel: Einkommen als oft benutzte Variable

## **Anwendungsbeispiel:**

- Gruppen mit höherem Einkommen neigen dazu, Markenprodukte zu erwerben
- Kaufhäuser nutzen diese Information für gezieltes Direktmarketing hochpreisiger Artikel

# Unüberwachtes Lernen - Clustering k-Means

## **Cluster definieren:**

- Zwei Fragen beantworten: Wie viele Cluster gibt es? Was gehört zu welchem Cluster?

## **Anzahl der Cluster:**

- Subjektive Frage
- Mehr Cluster = ähnliche Mitglieder, aber schwerer zu unterscheidende benachbarte Cluster
- Extremfall: Jeder Datenpunkt ist ein eigener Cluster

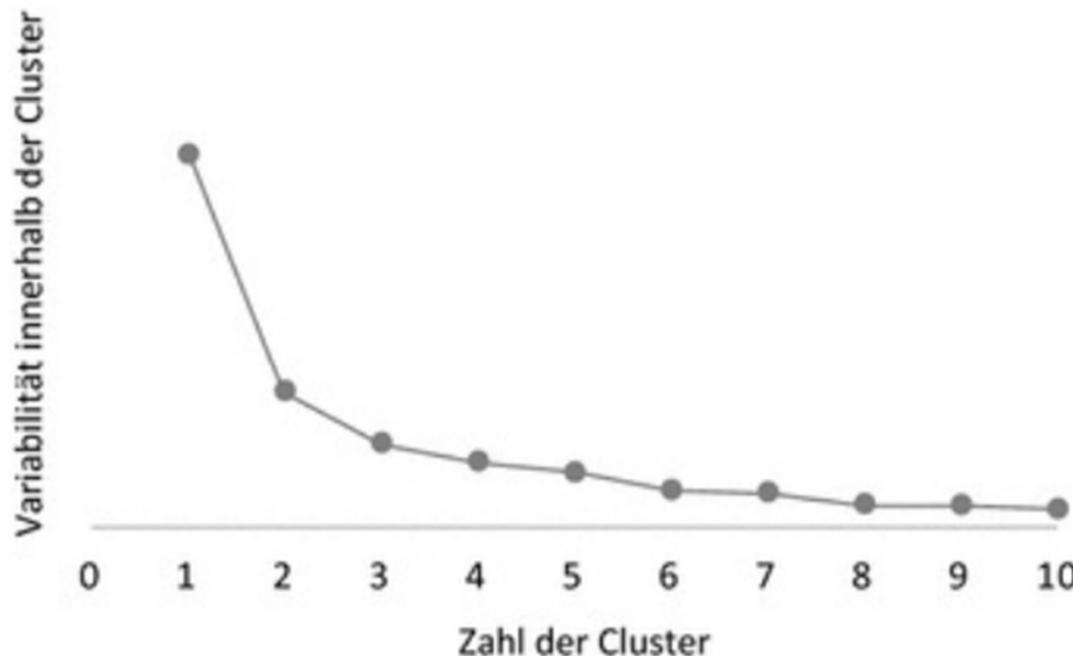
## **Mittelweg finden:**

- Anzahl der Cluster groß genug für aussagekräftige Muster
- Anzahl der Cluster klein genug für deutliche Unterscheidung

## **Scree-Test:**

- Methode zur Ermittlung der optimalen Anzahl von Clustern
- Scree-Plot: Streuung/Variabilität innerhalb der Cluster gegen die Anzahl der Cluster
- Abfallende Kurve: Variabilität sinkt mit steigender Clusteranzahl
- Knick in der Kurve deutet auf optimale Clusteranzahl hin

## Unüberwachtes Lernen k-Means



**Abb. 2.2** In diesem Scree-Plot finden wir einen Knick bei zwei bis drei Clustern

### Scree-Plot:

- Methode zur Ermittlung der optimalen Anzahl von Clustern
- Scree-Plot: Streuung/Variabilität innerhalb der Cluster gegen die Anzahl der Cluster
- Abfallende Kurve: Variabilität sinkt mit steigender Clusteranzahl
- Knick in der Kurve deutet auf optimale Clusteranzahl hin

# Unüberwachtes Lernen: Clustering k-Means

## Clusterzugehörigkeit:

- Iterativer Prozess zur Feststellung der Clusterzugehörigkeit
- Ziel: Datenpunkte in einem Cluster sollen möglichst dicht beieinander liegen und fern von anderen Clustern sein

## Iterativer Prozess beim k-Means-Clustering:

### • Schritt 0:

- Schätzung der (zwei) vorläufigen Clustermittelpunkte

### • Schritt 1:

- Zuordnung jedes Datenpunkts zum nächstgelegenen vorläufigen Mittelpunkt
- Ergebnis: Bildung von zwei Clustern (rot und blau)

### • Schritt 2:

- Berechnung der tatsächlichen Mittelpunkte der gebildeten Cluster
- Ersetzung der vorläufigen Mittelpunkte durch die tatsächlichen

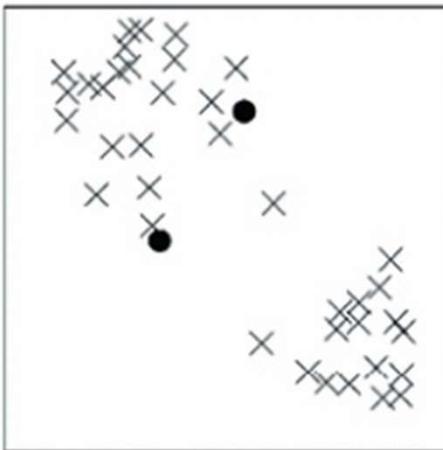
### • Schritt 3:

- Wiederholung der Schritte 1 und 2
- Fortsetzung bis keine Änderungen der Clusterzusammensetzung mehr auftreten

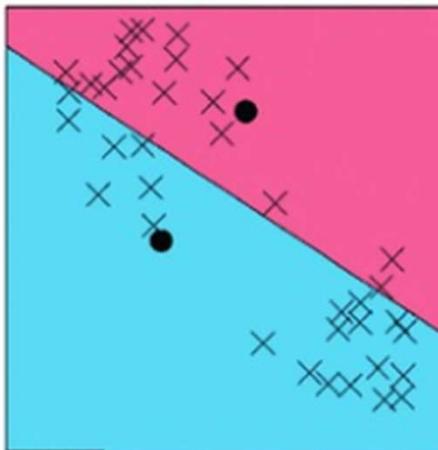
## Weitere Informationen:

- Clustering in drei oder mehr Dimensionen möglich
- Zusätzliche Dimensionen (z.B. Alter der Kunden, Häufigkeit der Besuche) können genutzt werden
- Schwierige Visualisierung in höheren Dimensionen, aber einfache Berechnung für Computerprogramme

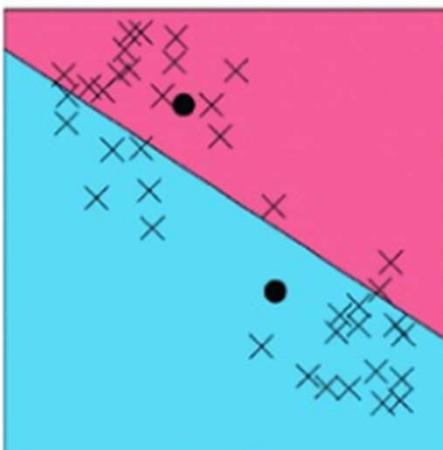
# Unüberwachtes Lernen: Clustering k-Means



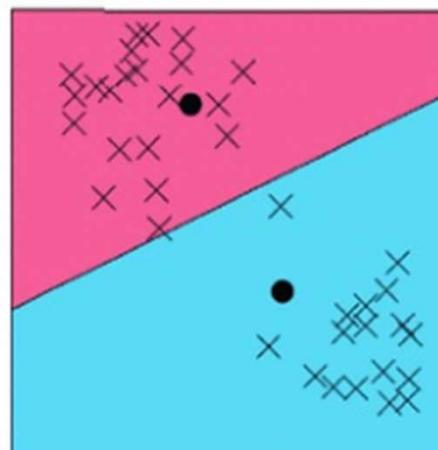
a) Schritt 0



b) Schritt 1



c) Schritt 2



d) Schritt 3

## Schritt 0:

- Schätzung der (zwei) vorläufigen Clustermittelpunkte

## Schritt 1:

- Zuordnung jedes Datenpunkts zum nächstgelegenen vorläufigen Mittelpunkt
- Ergebnis: Bildung von zwei Clustern (rot und blau)

## Schritt 2:

- Berechnung der tatsächlichen Mittelpunkte der gebildeten Cluster
- Ersetzung der vorläufigen Mittelpunkte durch die tatsächlichen

## Schritt 3:

- Wiederholung der Schritte 1 und 2
- Fortsetzung bis keine Änderungen der Clusterzusammensetzung mehr auftreten

# Unüberwachtes Lernen: Clustering k-Means - Herausforderungen

## Schwächen von k-Means-Clustering:

- Jeder Datenpunkt kann nur zu einem Cluster gehören.
  - Problematisch, wenn ein Datenpunkt zwischen zwei Clustern liegt.
- Annahme, dass Cluster kreisförmig (sphärisch in n Dimensionen) sind.
  - Iterative Zuordnung führt zu kompakten, kugelförmigen Clustern.
  - Problematisch bei tatsächlicher ellipsenförmiger oder ellipsoidischer Clusterform.
- Cluster schließen sich gegenseitig aus.
  - Keine Überlappungen oder Einschlüsse von Clustern erlaubt.

## Alternative Ansätze:

- Robuste Clustering-Verfahren:
  - Berechnen Wahrscheinlichkeiten, dass ein Datenpunkt zu mehreren Clustern gehört.
  - Ermöglichen nicht-sphärische oder überlappende Cluster.

## Stärken von k-Means-Clustering:

- Elegante Einfachheit des Algorithmus.
- Gute Strategie:
  - Mit k-Means-Clustering grundlegendes Verständnis der Datenstruktur gewinnen.
  - Schwächen des Algorithmus mit komplexeren Methoden abhelfen.

# Unüberwachtes Lernen: Clustering k-Means - Zusammenfassung

## **k-Means-Clustering:**

- Methode zur Gruppierung ähnlicher Datenpunkte
- Anzahl k der Cluster muss vorher festgelegt werden

## **Vorgehensweise:**

- Zuweisung jedes Datenpunkts zu einem vorläufigen Clustermittelpunkt
- Korrektur des Clustermittelpunkts anhand der zugewiesenen Punkte
- Wiederholung der Schritte bis keine Änderungen mehr auftreten

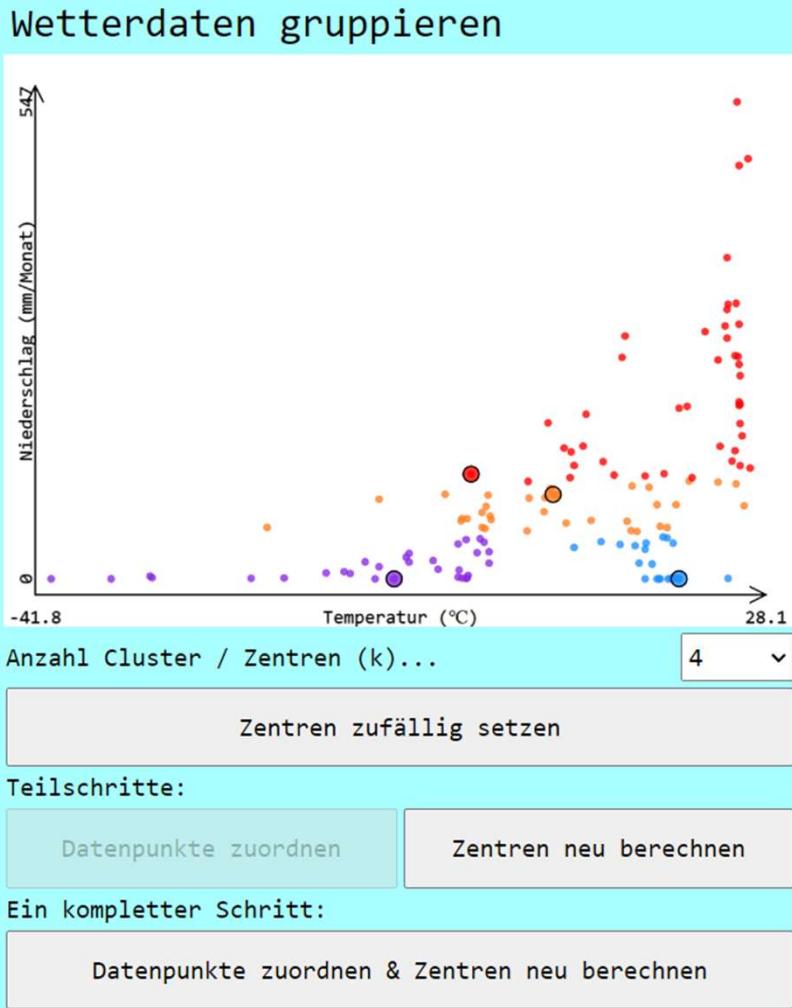
## **Eignung:**

- Funktioniert am besten bei sphärischen, nicht-überlappenden Clustern

# Unüberwachtes Lernen: Clustering k-Means – interaktives Beispiel

p5\*

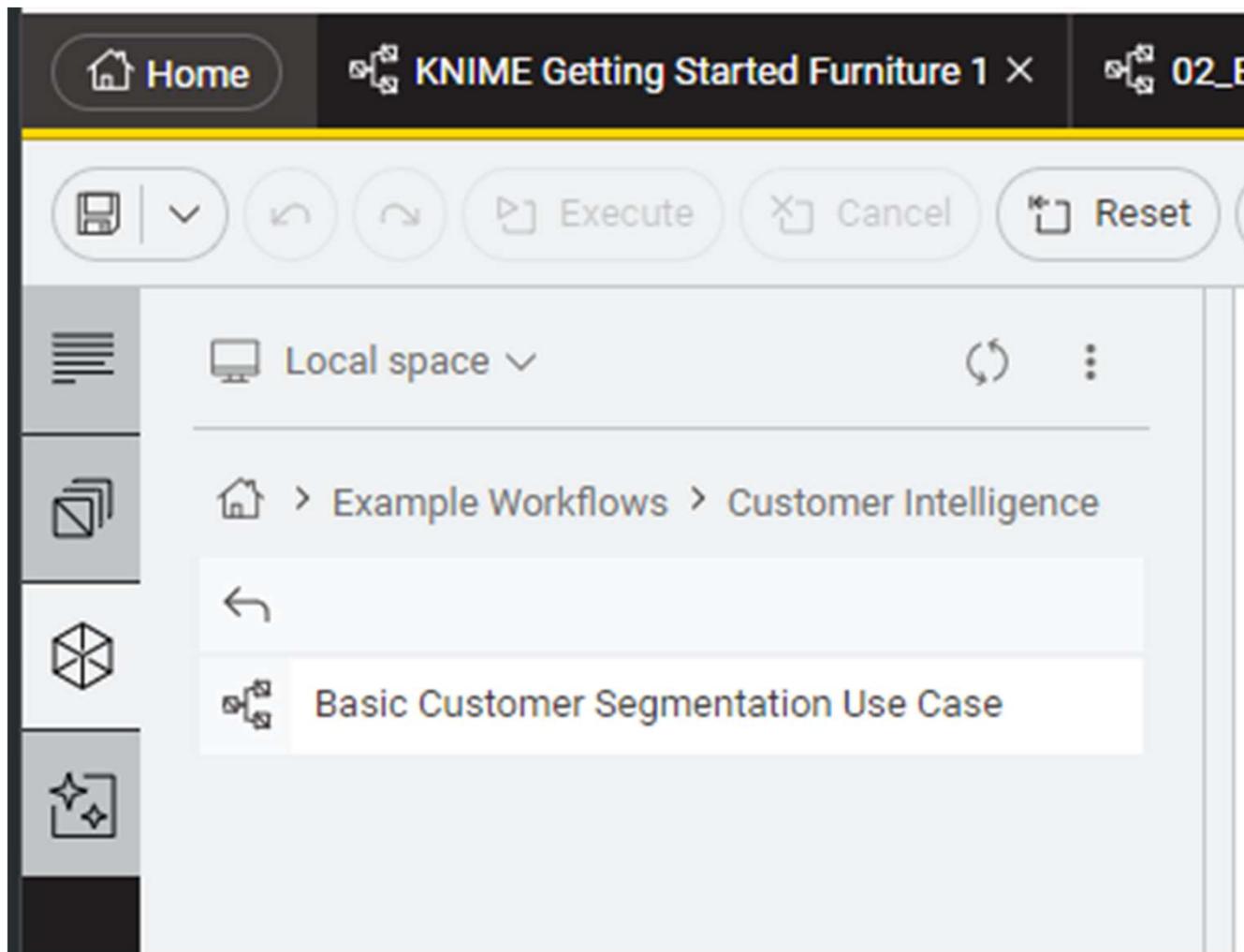
09\_01\_wetterdaten\_gruppieren by MaschinenNah



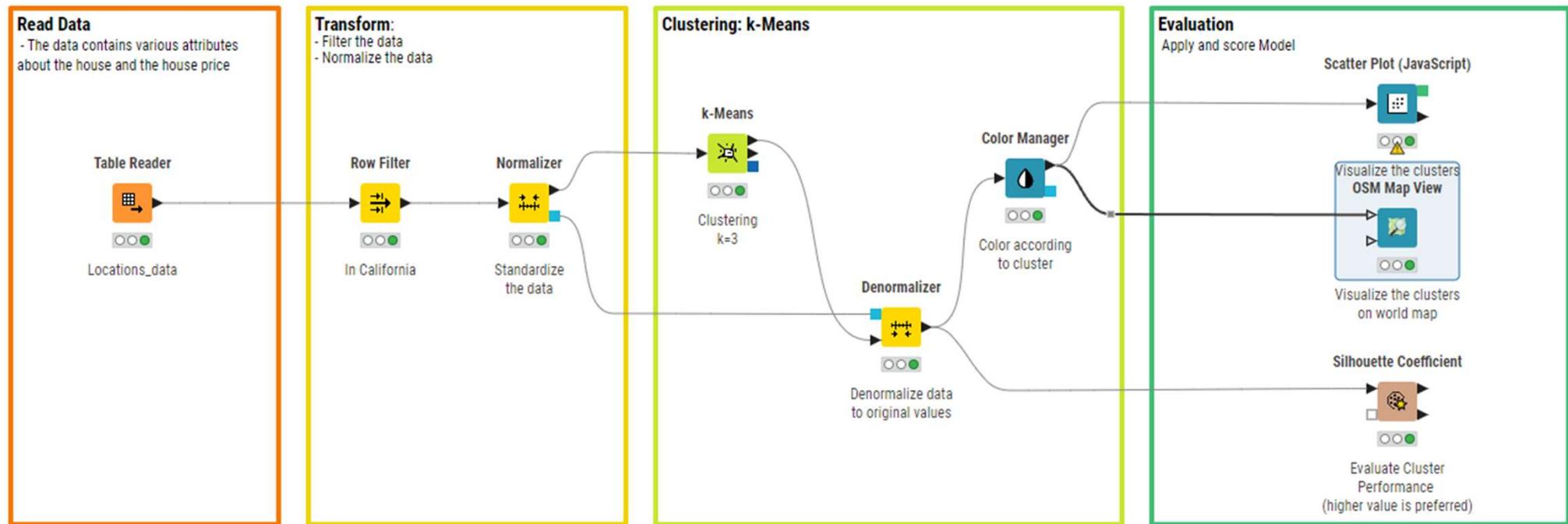
[https://www.maschinennah.de/ki-buch/  
Kapitel 9](https://www.maschinennah.de/ki-buch/Kapitel 9)

Direktlink:  
<https://editor.p5js.org/MaschinenNah/full/MIpq-o0q8>

# KNIME Workflow: Clustering Location Data



# KNIME Workflow: Clustering Location Data



## How to train Clustering Model?

- Step 1: Drag the k-Means node and double click to open the dialog
- Step 2: Select the "Number of clusters" as 3. In the "Column Selection" select the Columns "Lat" and "Long"
- Step 3: Right Click on the node and select "Execute" to perform clustering

## How to evaluate Clustering Model?

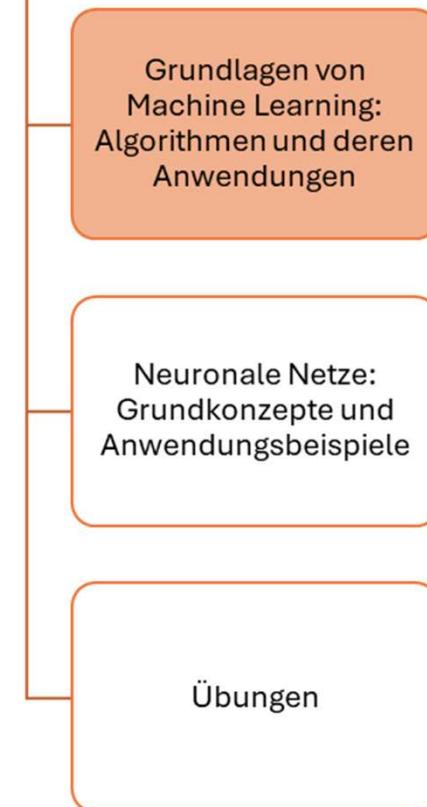
- Step 1: To visualize the clusters drag the Scatter Plot node and the OSM Map View. These nodes should be connected to Color Manager to visualize clusters with colors
- Step 2: To evaluate the clustering task, connect the clustering output to "Silhouette Coefficient" node. Select the "Clustering Column Selection" as "Cluster"
- Step 3: Execute the node to get Silhouette Coefficients for each instance, each cluster and for overall clustering task

# Grundlagen von Machine Learning: Algorithmen und deren Anwendungen

## Zusammenfassung

- Überwachtes Lernen
- Unüberwachtes Lernen
- Lineare/Multiple Regression
- KNIME Workflow Lineare Regression Hauspreise
- DS KI-Tutor Lineare Regression Hauspreise
- Klassifikation & Entscheidungsbäume
- Random Forest & Ensemble Learning
- Clustering & k-Means

**Mittwoch**  
**Maschinelles Lernen & Vorhersagemodelle**



# Neuronale Netze: Grundkonzepte und Anwendungsbeispiele

## Zusammenfassung

- Maschinelles Lernen vs Deep Learning
- Neuronale Netze – Intuition
- Neuronale Netze – Funktionsweise
- Neuronale Netze – Backpropagation

**Mittwoch**  
**Maschinelles Lernen & Vorhersagemodelle**

Grundlagen von  
Machine Learning:  
Algorithmen und deren  
Anwendungen

Neuronale Netze:  
Grundkonzepte und  
Anwendungsbeispiele

Übungen

# KNIME Kurs Machine Learning

<https://knime.learnupon.com/enrollments/219301013/page/848498723>

## Part 2: Data Science with KNIME Analytics Platform

COMPLETE

NCE

Introduction to Data

ack: Data

nd Learning

upervised

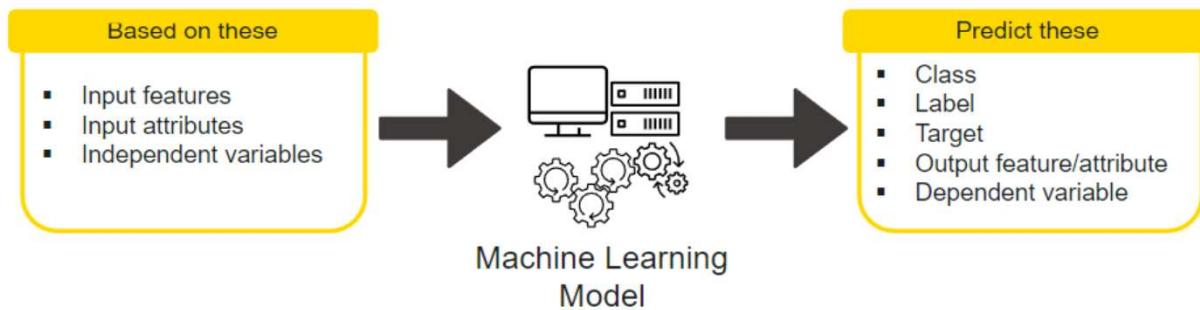
ed

## What is Machine Learning?

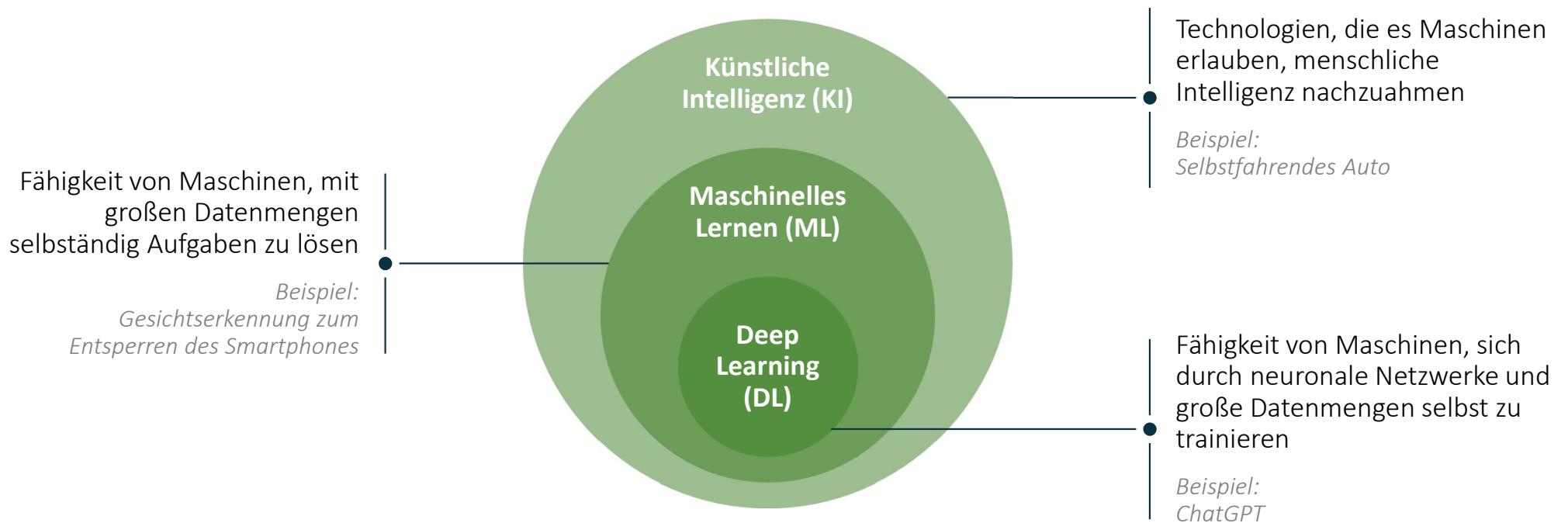
### 💡 Machine learning

is a field of data science focusing on developing algorithms to understand data using past observations without explicitly programming mathematical or statistical models.

Based on input features, or independent variables, machine learning models aim to predict output features, target classes, or dependent variables. In other words, the machine (or the model) learns (or adjusts itself) from past observations. For example, you can use the historical data containing days of the week, city districts, weather conditions, and the number of rented bikes, to train a model to learn to predict the number of rented bikes (target) given a day of the week, district, and weather conditions on that day (features).



# Künstliche Intelligenz, Machine Learning und Deep Learning



# Was ist KI - Begriffe

## Künstliche Intelligenz

Beschreibt Computerintelligenz ganz allgemein. Computersysteme erledigen Aufgaben, die normalerweise menschliche Intelligenz erfordern.

## Machine Learning

### (Maschinelles Lernen)

Ein Ansatz um KI zu ermöglichen. Computer bekommen die Fähigkeit aus Daten zu lernen ohne explizit programmiert zu werden. Die Algorithmen verbessern sich selbstständig durch Trainingsdaten.

## Deep Learning

### (tiefe neuronale Netze)

Technik für Umsetzung von ML mit Hilfe von tiefen neuronalen Netzen. Diese extrahieren komplexe Muster aus großen Datenmengen

## Generative KI

### (Transformer & Sprachmodelle ChatGPT/Dall-E)

Große Modelle generieren aus Eingabedaten (Prompt) und Trainingsdaten (Texte, Bilder) neue Inhalte.

Transformer = Deep Learning Architektur.

1950

1980

2010

2017

2023

# Was ist KI - Beispiele

## Künstliche Intelligenz

Beschreibt Computerintelligenz ganz allgemein. Computersysteme erledigen Aufgaben, die normalerweise menschliche Intelligenz erfordern.

### Machine Learning

*(Maschinelles Lernen)*

Ein Ansatz um KI umzusetzen. Computer bekommen die Fähigkeit aus Daten zu lernen ohne explizit programmiert zu werden. Die Algorithmen verbessern sich selbstständig durch Training.



### Empfehlungssysteme

- Wie sie von Online-Shops und Streaming-Diensten verwendet werden, um Produktempfehlungen basierend auf den Vorlieben des Benutzers zu machen.



### Betrugserkennung

- Banken und Finanzinstitute nutzen ML, um verdächtige Aktivitäten auf Konten zu erkennen.



### Kundenanalyse

- Unternehmen analysieren Kundenverhalten und -präferenzen, um Marketingstrategien zu optimieren.

# KI Begriffe Überblick

## Künstliche Intelligenz

Beschreibt Computerintelligenz ganz allgemein. Computersysteme erledigen Aufgaben, die normalerweise menschliche Intelligenz erfordern.

## Machine Learning

(*Maschinelles Lernen*)

Ein Ansatz um KI zu erreichen. Computer bekommen die Fähigkeit aus Daten zu lernen ohne explizit programmiert zu werden. Die Algorithmen verbessern sich selbstständig durch Training.

## Deep Learning

(*tiefe neuronale Netze*)

Technik für Umsetzung von ML mit Hilfe von tiefen neuronalen Netzen. Diese extrahieren komplexe Muster aus großen Datenmengen



### Autonome Fahrzeuge

- verwenden DL für die Bildverarbeitung und Entscheidungsfindung beim Fahren.



### Medizinische Bildanalyse

- Erkennen von Anomalien oder Krankheiten in medizinischen Bildern wie Röntgenbildern.



### Videospiele

- verwenden DL, um realistische Umgebungen oder intelligente NPCs zu schaffen.

# Was ist KI - Beispiele

## Künstliche Intelligenz

Beschreibt Computerintelligenz ganz allgemein. Computersysteme erledigen Aufgaben, die normalerweise menschliche Intelligenz erfordern.

### Machine Learning

(*Maschinelles Lernen*)

Ein Ansatz um KI zu erreichen. Computer bekommen die Fähigkeit aus Daten zu lernen ohne explizit programmiert zu werden. Die Algorithmen verbessern sich selbstständig durch Training.

### Deep Learning

(*tiefe neuronale Netze*)

Technik für Umsetzung von ML mit Hilfe von tiefen neuronalen Netzen. Diese extrahieren komplexe Muster aus großen Datenmengen

### Generative KI

(*Transformer & Sprachmodelle ChatGPT/Dall-E*)

Große Modelle generieren aus Eingabedaten (Prompt) und Trainingsdaten (Texte, Bilder) neue Inhalte. Transformer – Deep Learning Architektur.

- ❖ Chatbots
- ❖ Sprachsynthese
- ❖ Audiosynthese

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze



Abbildung 1-2: Schema einer einfachen Vorhersagemaschine

Stellen Sie sich eine simple Maschine vor, die eine Frage entgegennimmt, etwas »nachdenkt« und eine Antwort ausgibt. Das läuft genau wie im obigen Beispiel ab, in dem wir selbst die Eingaben über die Augen aufnehmen, mit unserem Gehirn die Szene analysieren und daraus ableiten, was die Objekte in dieser Szene bedeuten. Abbildung 1-2 stellt dies schematisch dar.

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze



Abbildung 1-3: Alternative Beschreibung der Vorhersagemaschine

Computer denken nicht wirklich, sie sind lediglich bessere Taschenrechner. Deshalb wollen wir die Vorgänge mit treffenderen Worten beschreiben (siehe Abbildung 1-3).

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze



Abbildung 1-4: Beispiel für die Verarbeitung einer Multiplikation

Ein Computer nimmt eine Eingabe entgegen, führt bestimmte Berechnungen aus und liefert dann eine Ausgabe. Das folgende Beispiel soll das veranschaulichen. Es wird eine Eingabe von » $3 \times 4$ « verarbeitet. Das geschieht möglicherweise dadurch, dass die Multiplikation in einen einfacheren Satz von Additionen überführt wird. Die ausgegebene Antwort lautet »12«.

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze



Abbildung 1-5: Umrechnung von Kilometern in Meilen

Fahren wir die Komplexität jetzt eine winzige Stufe höher.  
Stellen Sie sich eine Maschine vor, die Kilometer in Meilen umrechnet  
(siehe Abbildung 1-5).

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze

Tabelle 1-2: Wertepaare für die Umrechnung zwischen Kilometern und Meilen

Wahrheitsbeispiel	Kilometer	Meilen
1	0	0
2	100	62,137



Abbildung 1-6: Zufällig gewählte Konstante  $c$

# Grundlegende Methoden Maschinelles Lernen: Neuronale Netze

Tabelle 1-2: Wertepaare für die Umrechnung zwischen Kilometern und Meilen

Wahrheitsbeispiel	Kilometer	Meilen
1	0	0
2	100	62,137

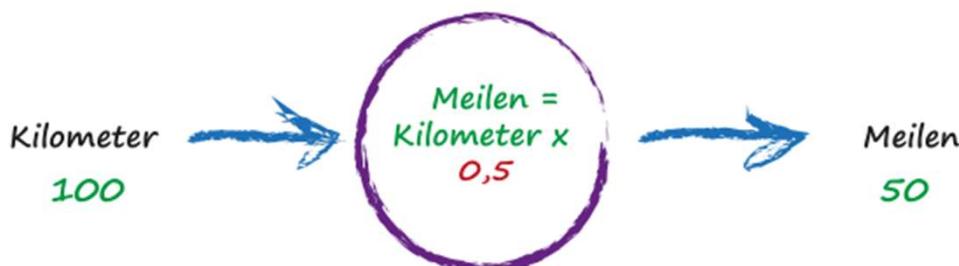


Abbildung 1-6: Zufällig gewählte Konstante c

Nun nehmen wir an, dass wir die Formel für die Umrechnung zwischen Kilometern und Meilen nicht kennen. Wir wissen lediglich, dass die Beziehung zwischen beiden linear ist. Wenn man also die Anzahl der Meilen verdoppelt, wird die gleiche Entfernung in Kilometern ebenfalls verdoppelt. Das ist intuitiv verständlich.

Diese lineare Beziehung zwischen Kilometern und Meilen liefert uns einen Anhaltspunkt über diese geheimnisvolle Berechnung – sie muss die Form haben: Meilen = Kilometer  $\cdot$  c, wobei c eine Konstante ist. Den Wert dieser Konstanten c kennen wir aber noch nicht.

Die einzigen anderen Anhaltspunkte liefern einige Beispiele, die Kilometer und Meilen paarweise angeben. Diese sind wie Beobachtungen der Wirklichkeit, mit denen man wissenschaftliche Theorien überprüft – sie sind Beispiele für die Wahrheit der echten Welt.

Was sollten wir tun, um die fehlende Konstante c zu ermitteln? Setzen wir einfach einmal einen zufälligen Wert ein und probieren wir es aus! Versuchen wir es mit c = 0,5 und schauen wir, was passiert.

Hier haben wir Meilen = Kilometer  $\cdot$  c, wobei Kilometer gleich 100 und c unsere derzeitige Schätzung 0,5 sind. Damit erhalten wir 50 Meilen.

Nun gut. Das ist gar nicht mal so schlecht unter dem Aspekt, dass wir c = 0,5 zufällig ausgewählt haben! Doch wir wissen, dass das Ergebnis nicht genau ist, weil das Wahrheitsbeispiel Nummer 2 uns sagt, dass die Antwort 62,137 sein sollte.

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze



Abbildung 1-7: Der Fehler bei unserer ersten Schätzung

Wir liegen um 12,137 daneben. Das ist der Fehler, die Differenz zwischen unserer berechneten Antwort und der tatsächlichen Wahrheit aus unserer Beispielliste.

$$\begin{aligned}\text{Das heißt, Fehler} &= \text{wahr} - \text{berechnet} \\ &= 62,137 - 50 \\ &= 12,137\end{aligned}$$

Was kommt als Nächstes? Wir wissen, dass wir falsch liegen und wie groß die Abweichung ist. Anstatt nun aufgrund dieses Fehlers zu verzweifeln, nutzen wir ihn, um zu einer zweiten, besseren Schätzung für  $c$  zu gelangen.

Sehen Sie sich diesen Fehler noch einmal an. Wir haben 12,137 zu wenig geschätzt Beziehung darstellt ( $\text{Meilen} = \text{Kilometer} \cdot c$ ), wissen wir, dass bei einer Erhöhung von  $c$  auch der Ausgabewert größer wird.tzt. Da die Formel für die Umrechnung von Kilometern in Meilen eine lineare Beziehung darstellt ( $\text{Meilen} = \text{Kilometer} \cdot c$ ), wissen wir, dass bei einer Erhöhung von  $c$  auch der Ausgabewert größer wird.

## Grundlegende Methoden Maschinelles Lernen: Neuronale Netze

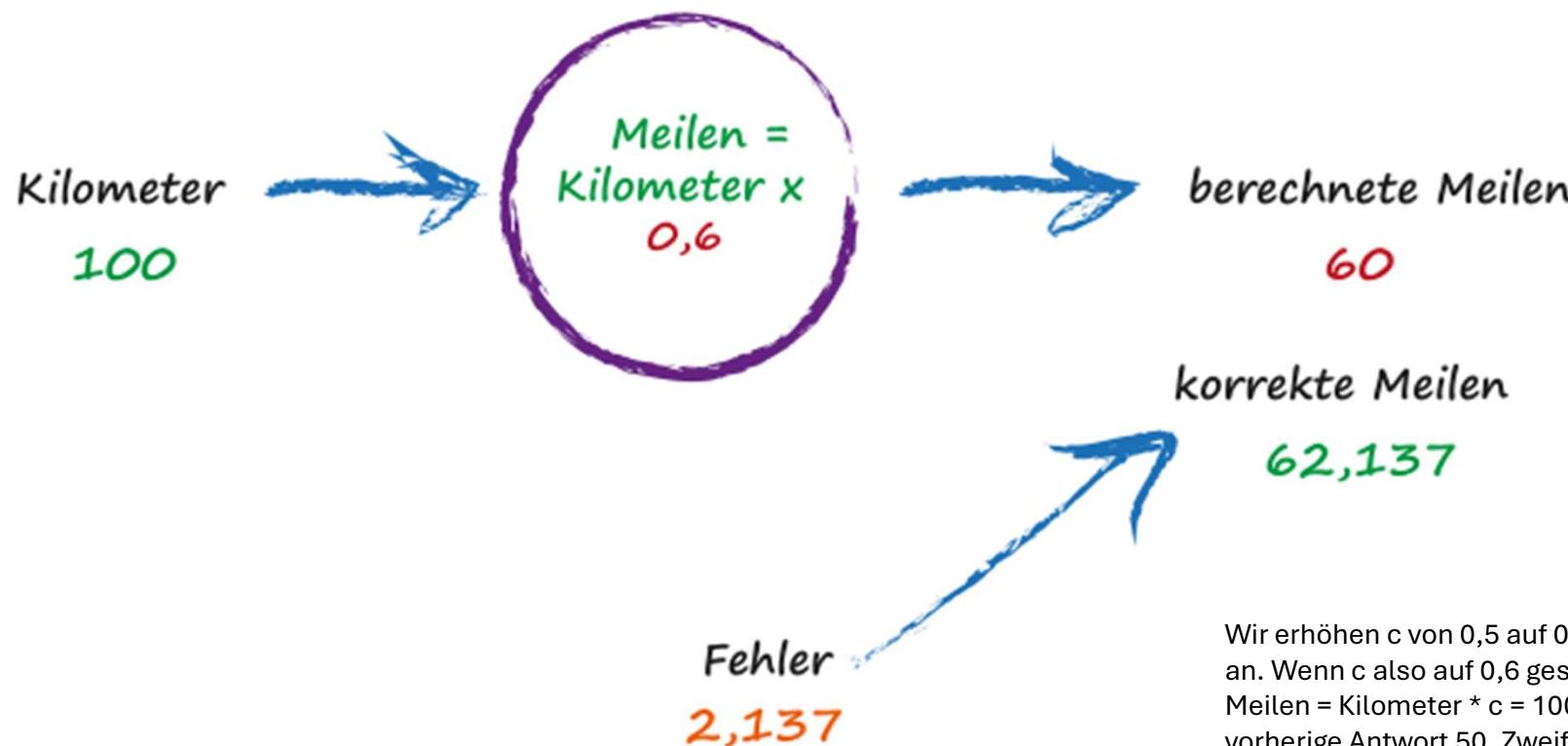


Abbildung 1-8: Die zweite Schätzung ergibt einen kleineren Fehler.

Wir erhöhen  $c$  von 0,5 auf 0,6 und sehen uns das neue Ergebnis an. Wenn  $c$  also auf 0,6 gesetzt ist, erhalten wir  $\text{Meilen} = \text{Kilometer} * c = 100 * 0,6 = 60$ . Das ist besser als die vorherige Antwort 50. Zweifellos haben wir einen Fortschritt gemacht!  
Der Fehler ist nun mit 2,137 viel kleiner. Es könnte sogar ein Fehler sein, mit dem wir durchaus leben können.

Wichtig ist hier, dass wir uns bei der Entscheidung, um wie viel der Wert von  $c$  angehoben werden soll, am Fehler orientiert haben. Wir wollten die Ausgabe 50 vergrößern, also haben wir  $c$  ein wenig erhöht.

# Grundlegende Methoden Maschinelles Lernen: Neuronale Netze

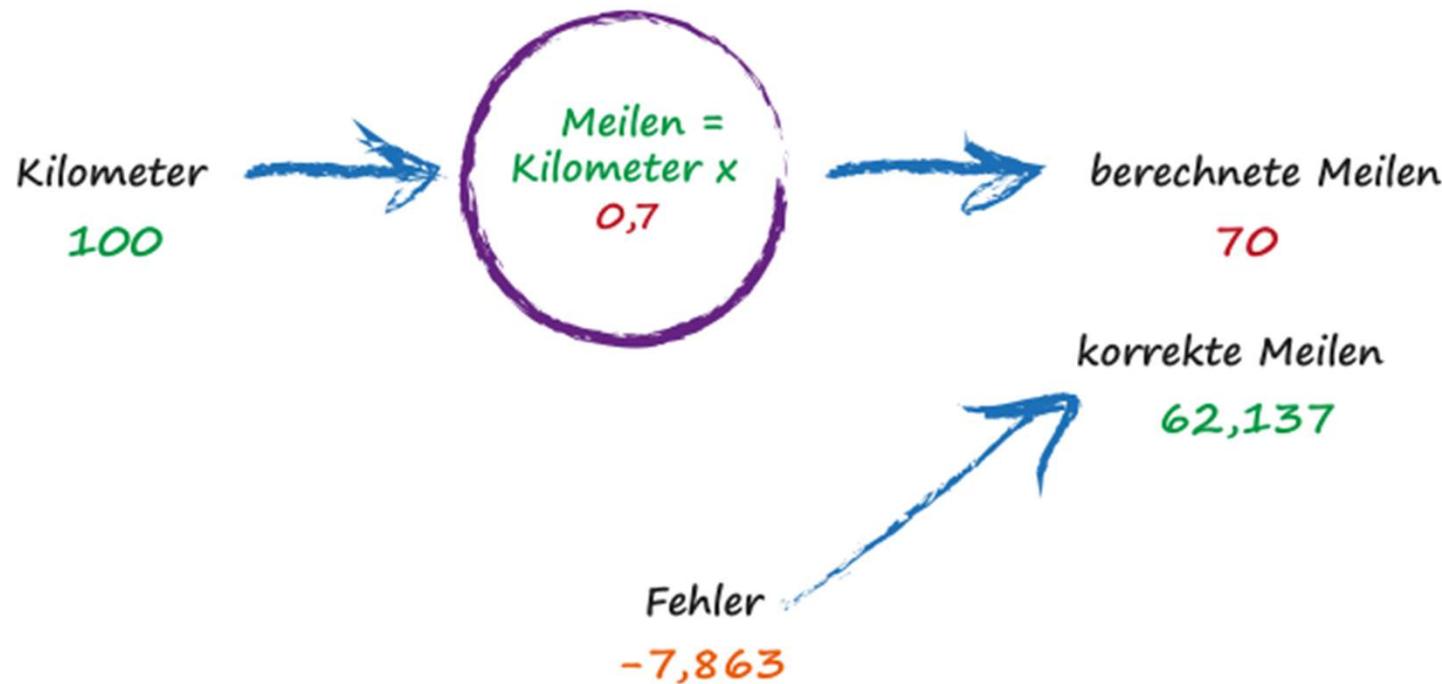


Abbildung 1-9: Die nächste Schätzung liefert einen negativen Fehler.

Fahren wir also fort. Die Ausgabe von 60 ist immer noch zu klein. Wir schrauben den Wert von  $c$  ein weiteres Mal nach oben, und zwar von 0,6 auf 0,7.

Oh, nein! Wir sind zu weit gegangen und über die korrekte Antwort hinausgeschossen. Der vorherige Fehler war 2,137, nun liegt er aber bei -7,863. Das Minuszeichen besagt lediglich, dass wir den Zielwert überschritten statt unterschritten haben. Denn der Fehler ergibt sich als korrekter Wert – berechneter Wert. Da nun  $c = 0,6$  besser als  $c = 0,7$  war, könnten wir uns mit dem kleinen Fehler von  $c = 0,6$  zufriedengeben und diese Übung jetzt beenden. Doch wir wollen noch ein wenig weitergehen. Wir könnten  $c$  doch auch in kleineren Schritten verändern, beispielsweise von 0,6 auf 0,61.

# Grundlegende Methoden Maschinelles Lernen: Neuronale Netze

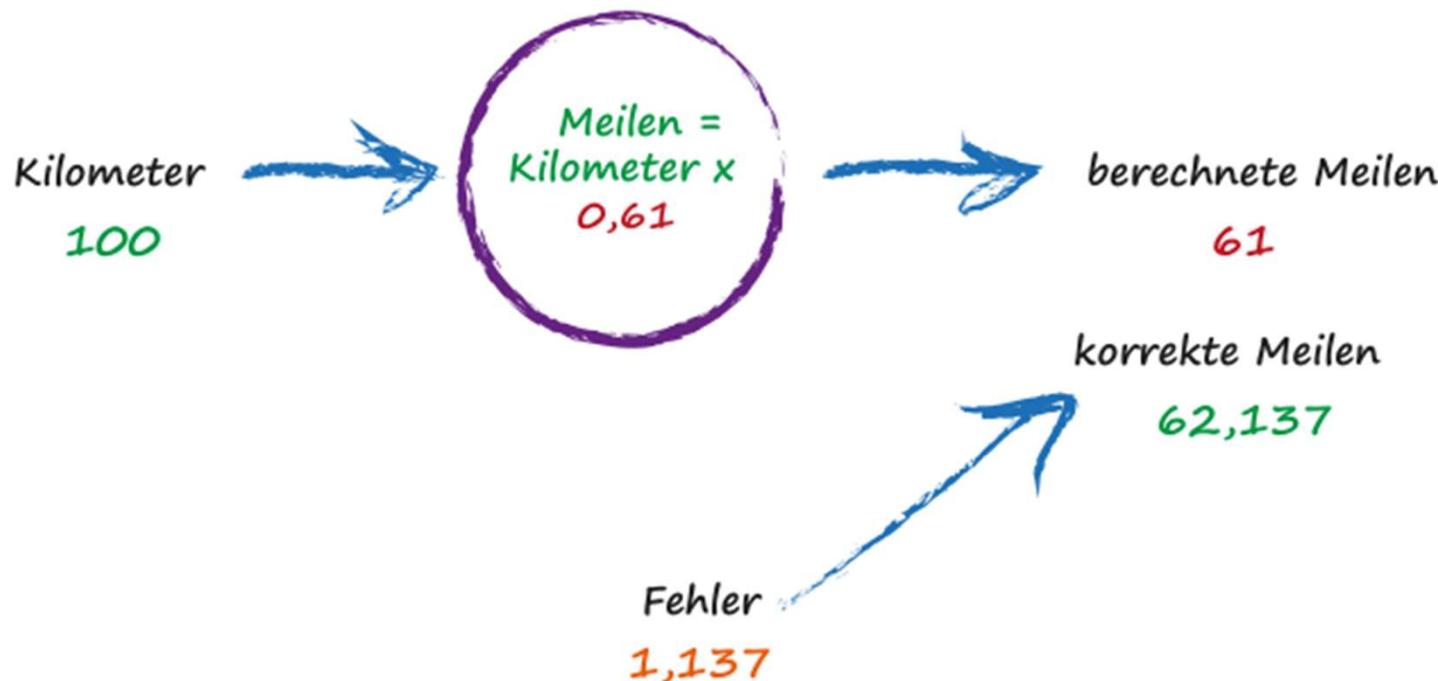


Abbildung 1-10: Die Konstante  $c$  wird jetzt in kleineren Schritten verändert.

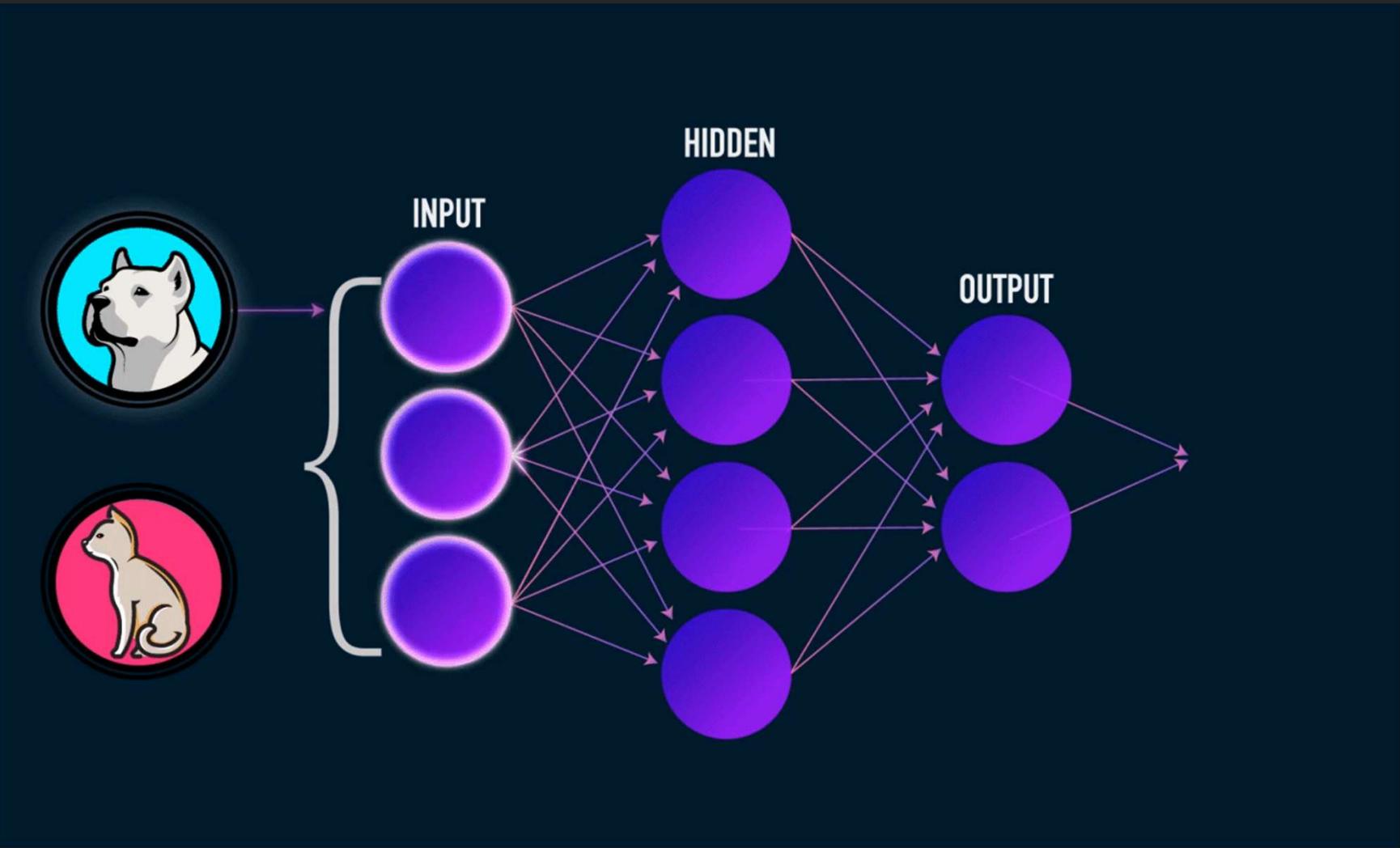
Das ist schon viel besser als vorher. Wir haben einen Ausgabewert von 61, der nur 1,137 vom exakten Wert 62,137 abweicht.

Dieser letzte Versuch hat uns also gelehrt, dass wir den Wert von  $c$  moderat verändern sollten. Wenn die Ausgaben der korrekten Antwort näher kommen – d. h. der Fehler kleiner wird –, sollte man die veränderbare Komponente nicht so stark anheben. Auf diese Weise lässt sich vermeiden, dass über den richtigen Wert hinausgeschossen wird, wie es weiter oben passiert ist.

Das ist intuitiv richtig – ein großer Fehler bedeutet, dass eine größere Korrektur erforderlich ist, und ein winziger Fehler heißt, wir brauchen die kleinsten Anpassungsschritte für  $c$ .

Ob Sie es glauben oder nicht, wir sind gerade den prinzipiellen Lernprozess in einem neuronalen Netz durchgegangen – wir haben die Maschine trainiert, damit sie immer besser dabei wird, die richtige Antwort zu geben. Es lohnt sich, kurz innezuhalten und darüber nachzudenken – wir haben ein Problem nicht in einem einzigen Schritt genau gelöst, wie wir es oftmals in der Schulmathematik oder bei wissenschaftlichen Problemen tun. Stattdessen haben wir einen gänzlich anderen Weg eingeschlagen, indem wir eine Antwort ausprobiert und sie wiederholt verbessert haben. Man spricht auch von iterativ und meint damit, dass eine Antwort wiederholt Stück für Stück verbessert wird.

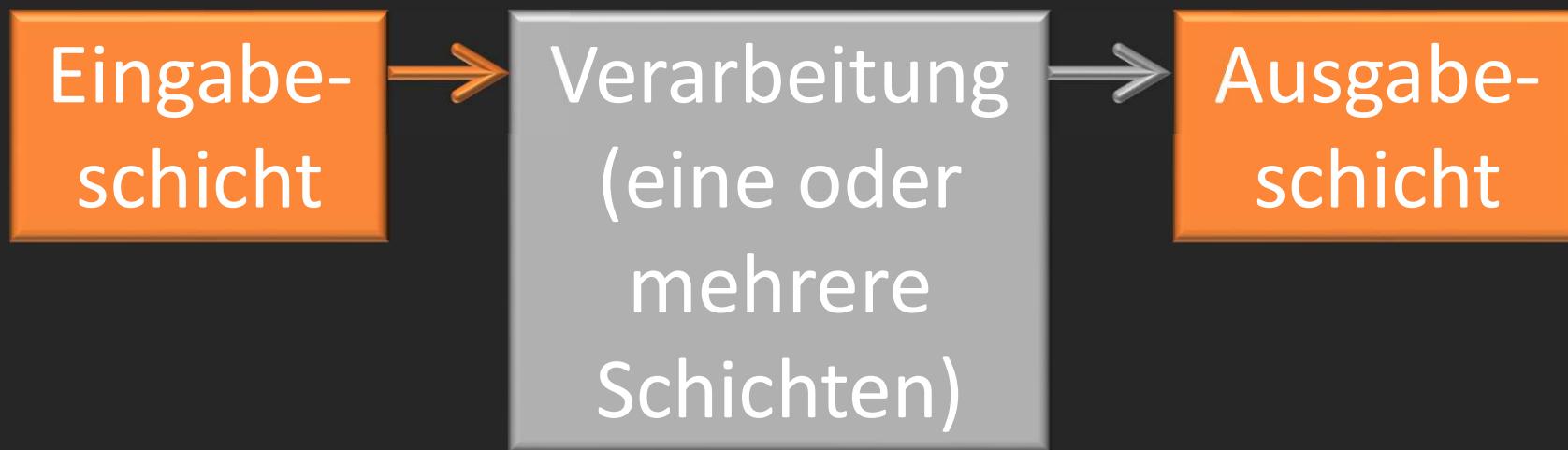
# Was sind neuronale Netze?



# Aufbau neuronaler Netze

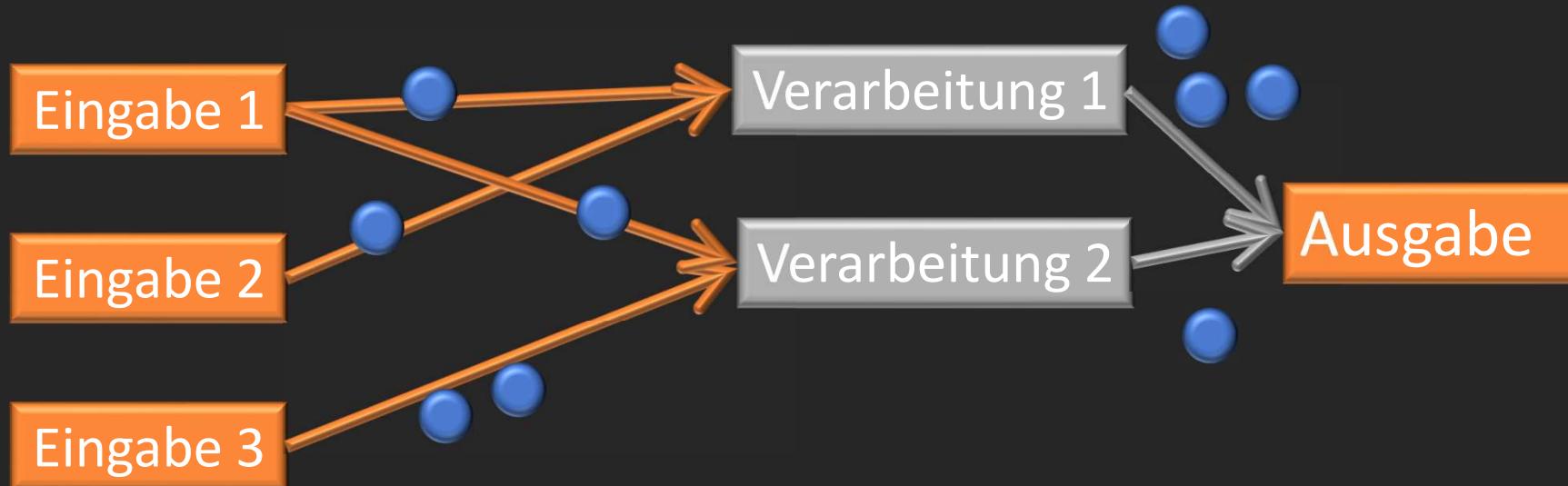


# Aufbau neuronaler Netze



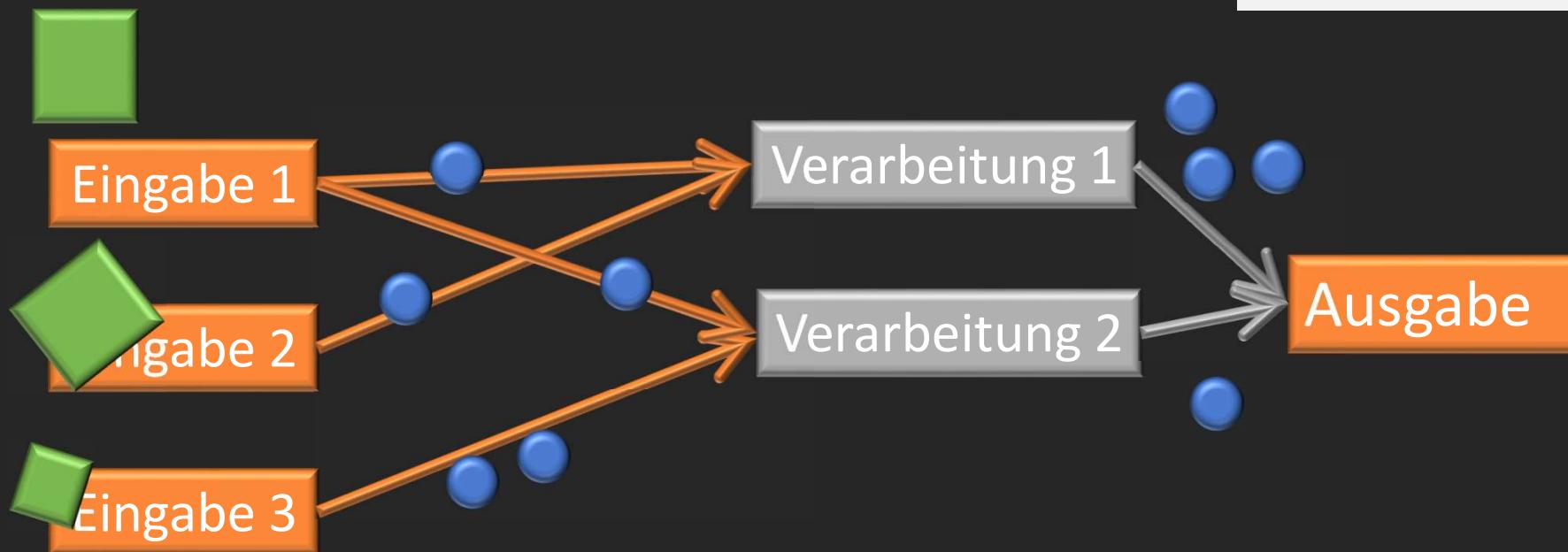
# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



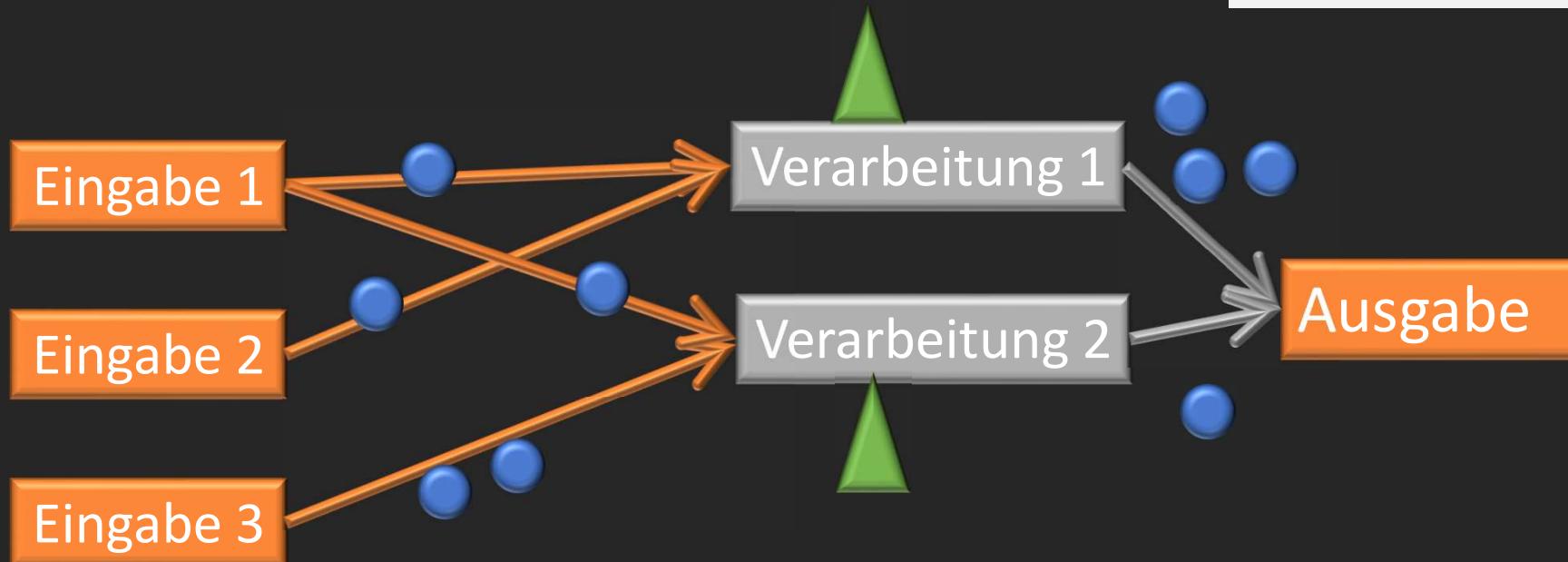
# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



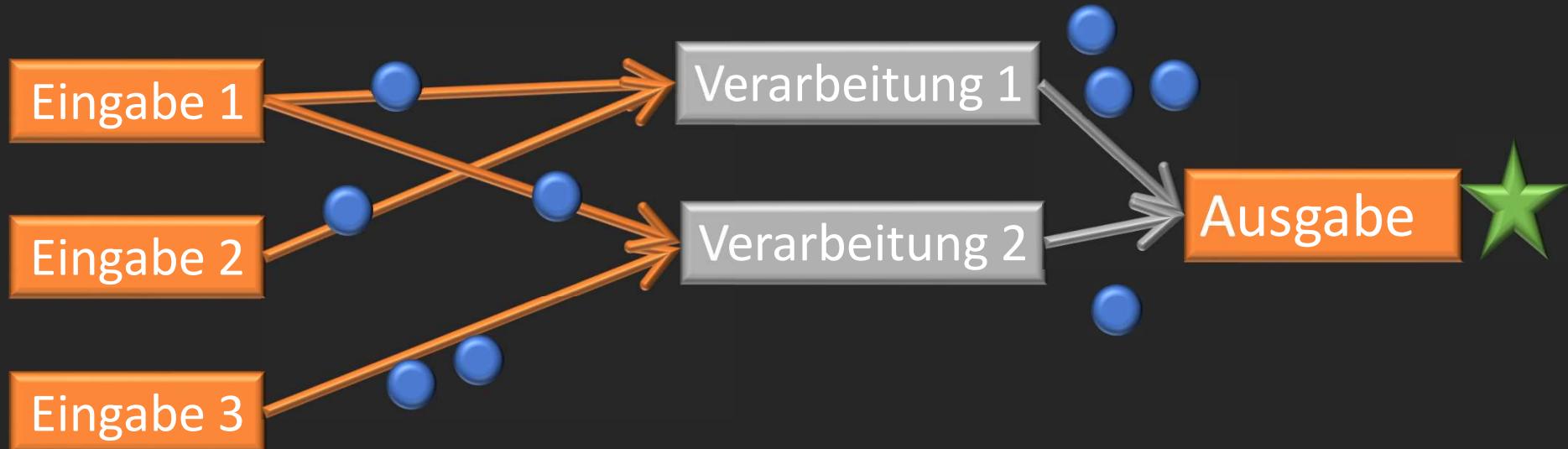
# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



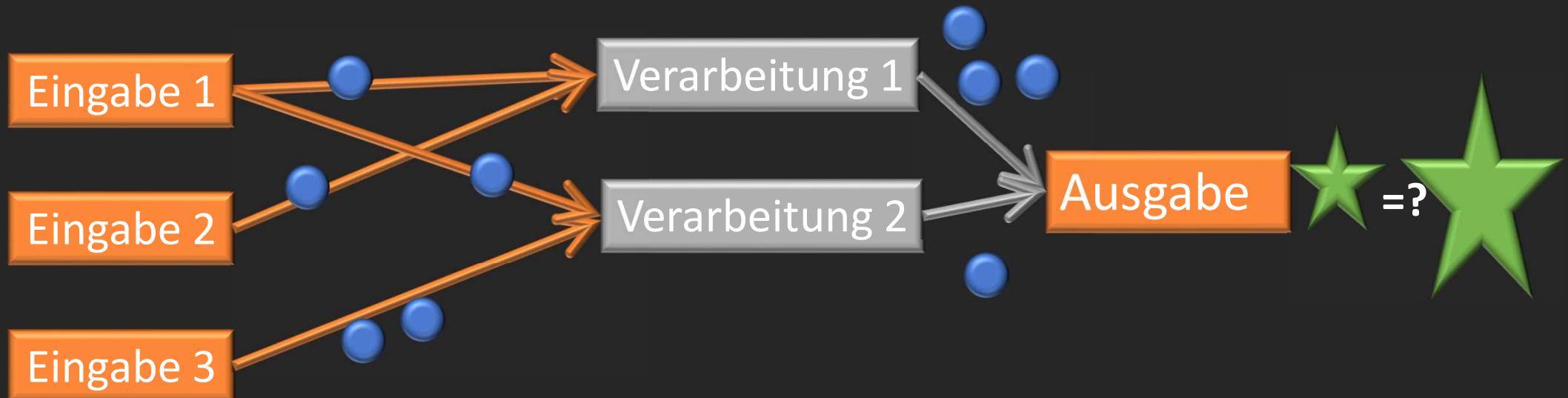
# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



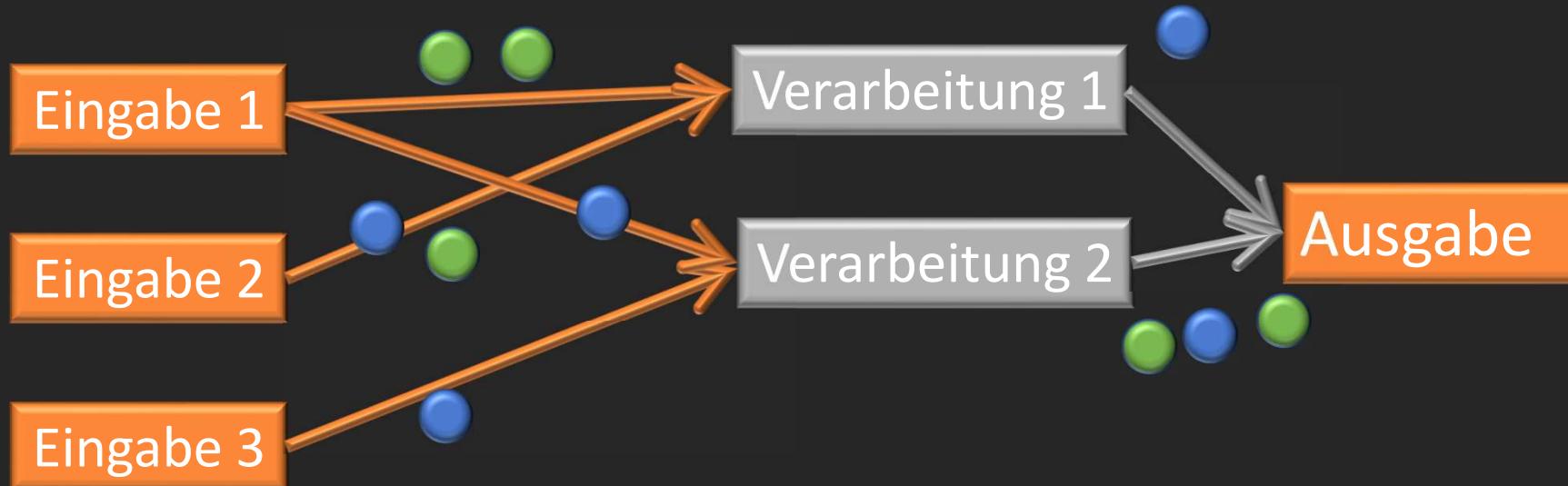
# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



# Einfaches neuronales Netz

1. Netz bekommt Daten
2. Netz verarbeitet die Daten
3. Ergebnis wird verglichen und der Fehler berechnet
4. Gewichte werden geändert



# Einführung in Machine Learning: Neuronale Netze

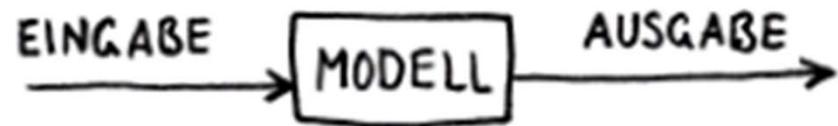


Abbildung 10.4 Ein Modell ordnet einer Eingabe eine Ausgabe zu.

# Einführung in Machine Learning: Neuronale Netze

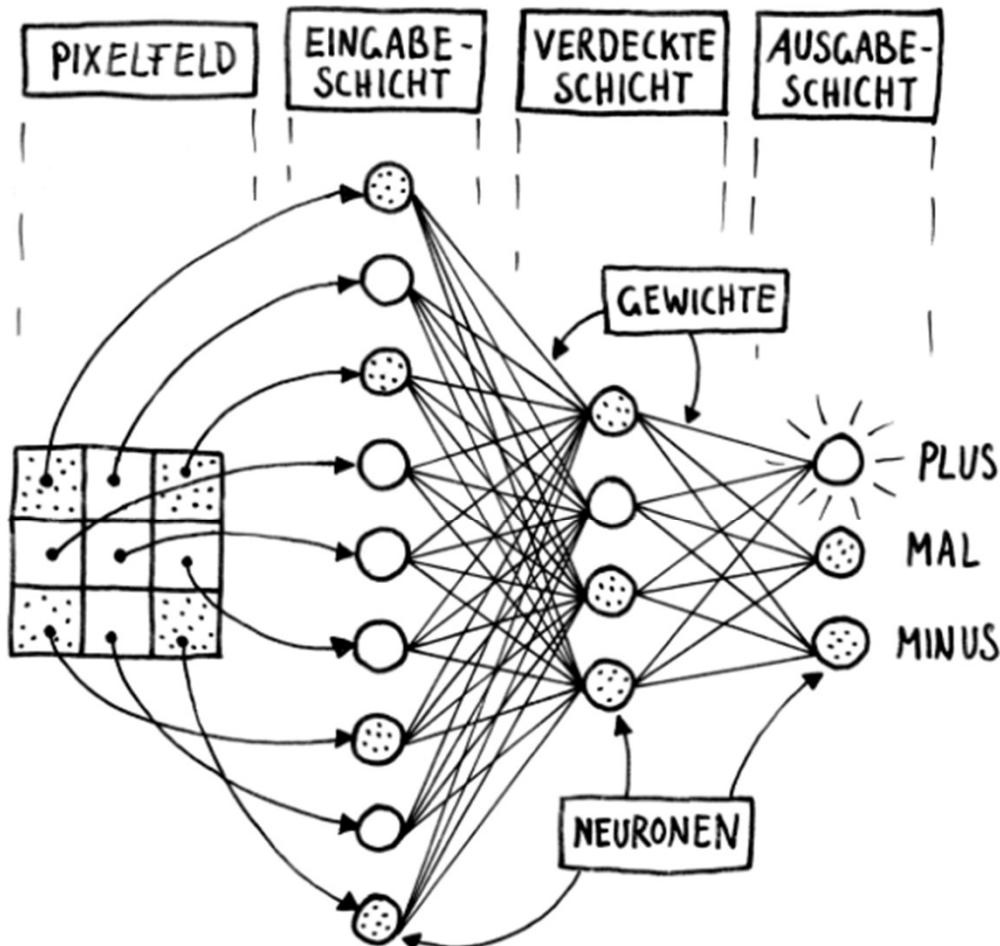
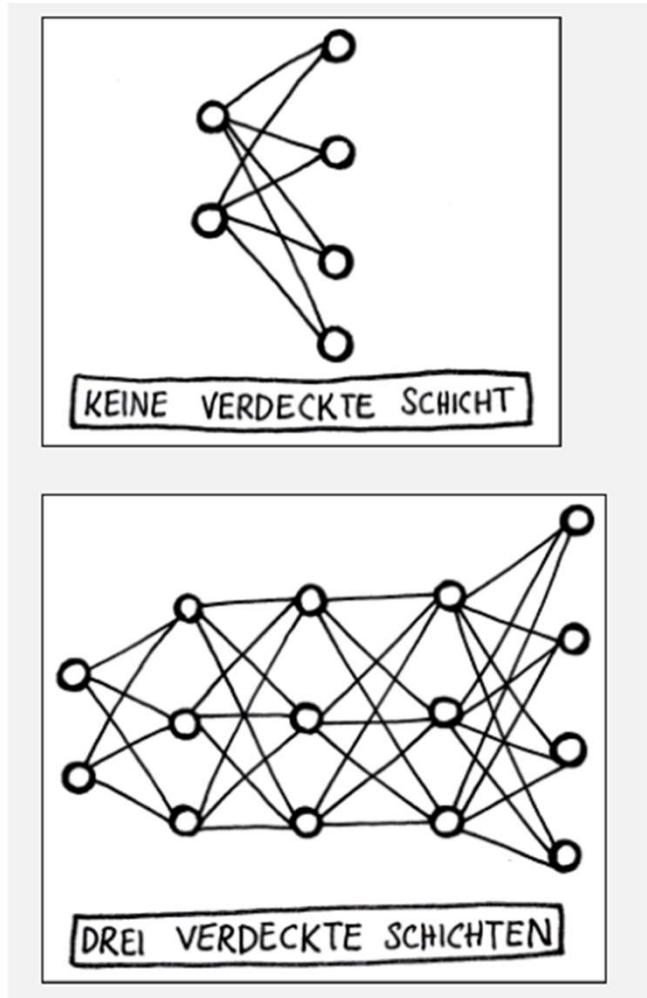


Abbildung 10.5 Ein neuronales Netz zur Erkennung von Zeichen

Quelle: Künstliche Intelligenz verstehen – Eine Spielerische Einführung, 2024

# Einführung in Machine Learning: Neuronale Netze



Quelle: Künstliche Intelligenz verstehen – Eine Spielerische Einführung, 2024

# Einführung in Machine Learning: Neuronale Netze

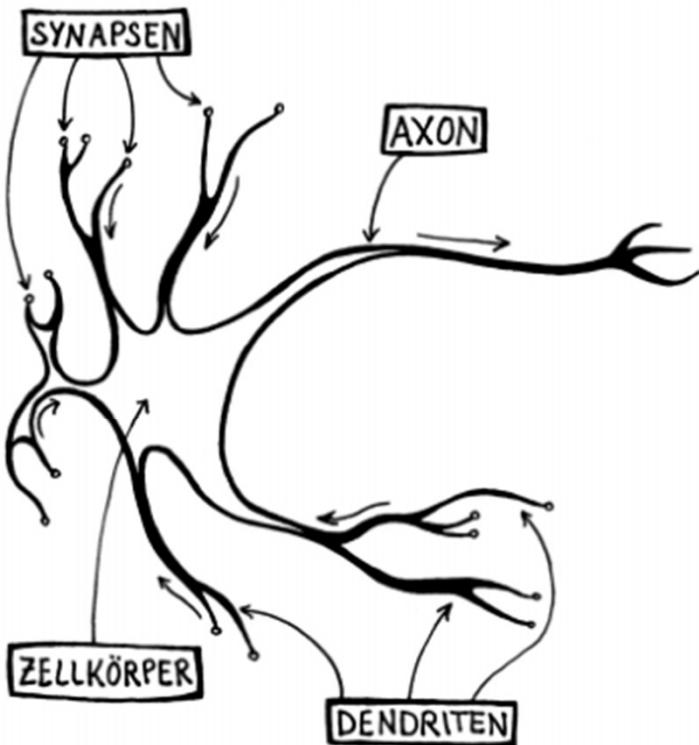
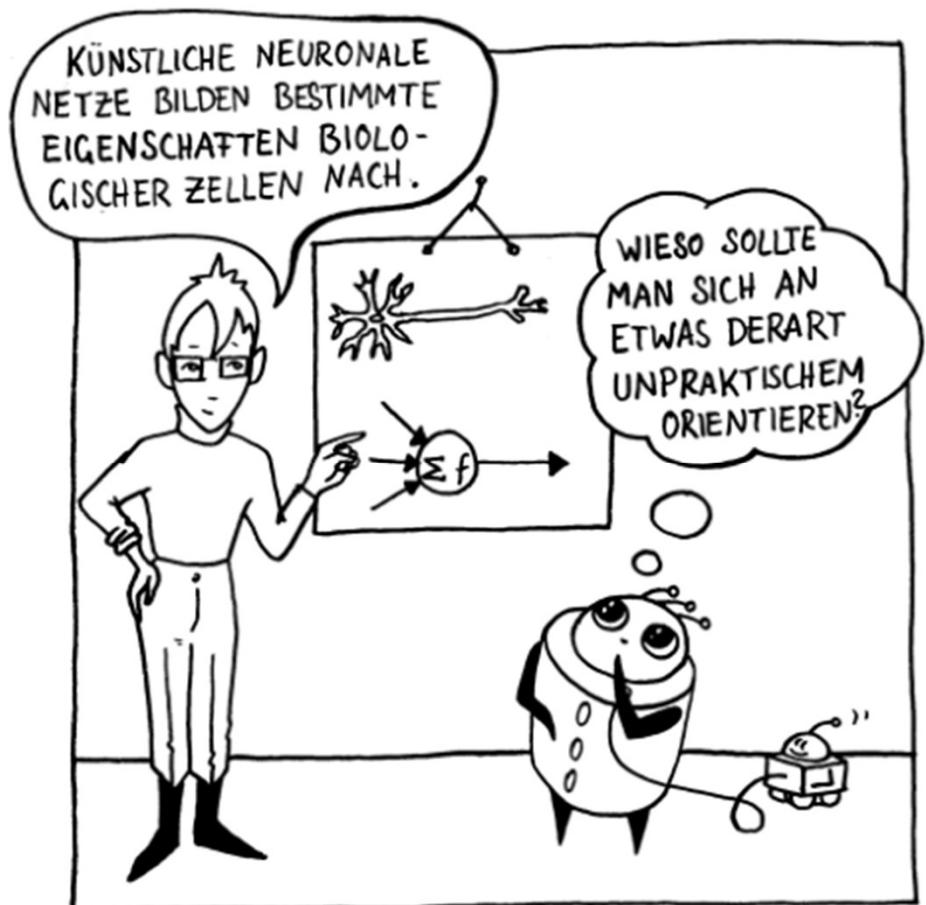


Abbildung 10.7 Das Modell einer biologischen Nervenzelle



# Einführung in Machine Learning: Neuronale Netze

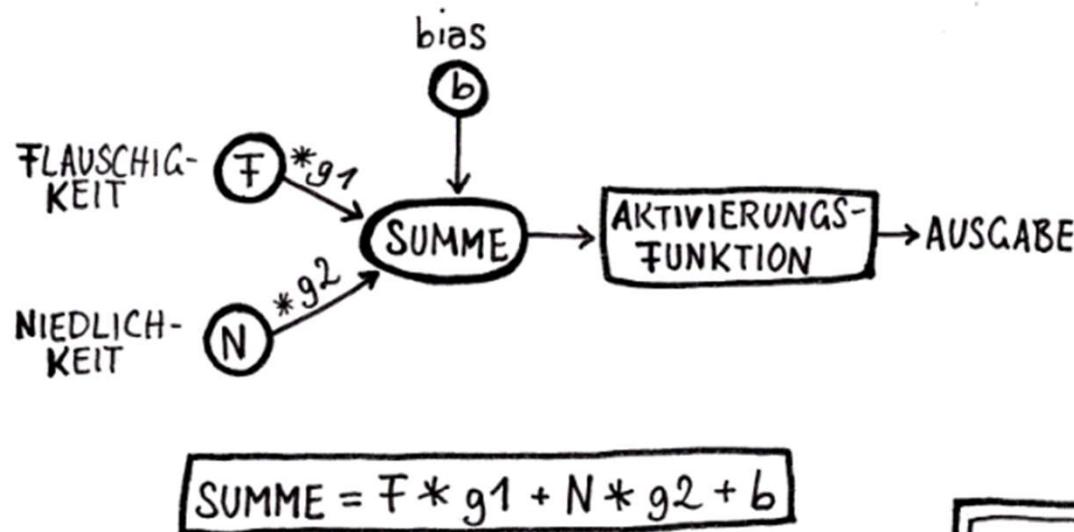
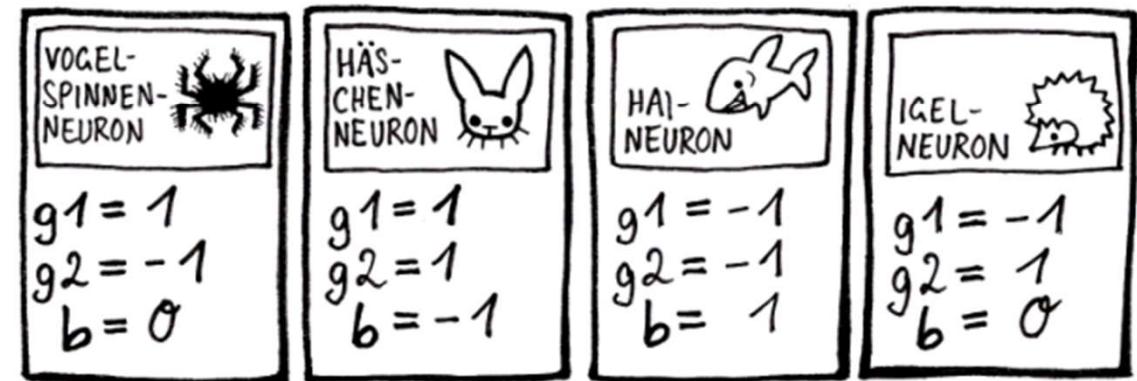


Abbildung 10.8 Ein künstliches Neuron zur Tiererkennung



# Einführung in Machine Learning: Neuronale Netze

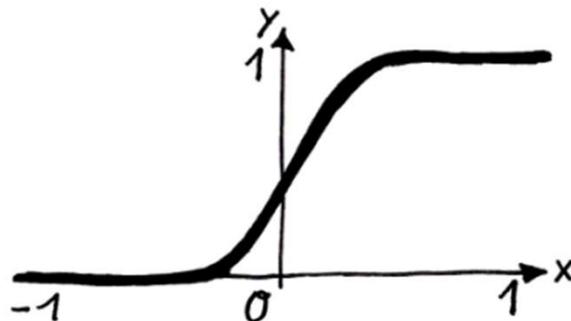


Abbildung 10.10 Die Sigmoidfunktion ist eine stetige Alternative zur Sprungfunktion.

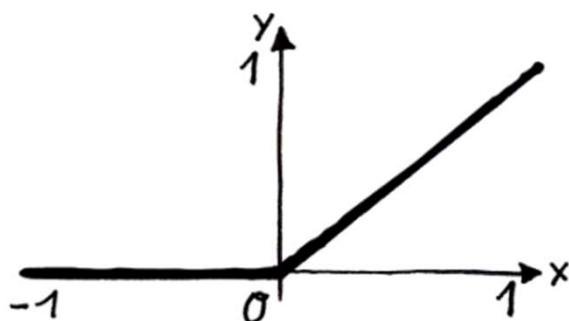


Abbildung 10.11 Die reLU-Funktion reicht alle Werte größer 0 unverändert durch; Werte kleiner 0 werden auf 0 gesetzt.

# Neuronale Netze: Grundkonzepte und Anwendungsbeispiele

## Zusammenfassung

- Maschinelles Lernen vs Deep Learning
- Neuronale Netze – Intuition
- Neuronale Netze – Funktionsweise
- Neuronale Netze – Backpropagation

**Mittwoch**  
**Maschinelles Lernen & Vorhersagemodelle**

Grundlagen von  
Machine Learning:  
Algorithmen und deren  
Anwendungen

Neuronale Netze:  
Grundkonzepte und  
Anwendungsbeispiele

Übungen

# Neuronale Netze: Übungen

## Zusammenfassung

- Tiere erkennen
- Gradienten
- Backpropagation
- Gängige Missverständnisse

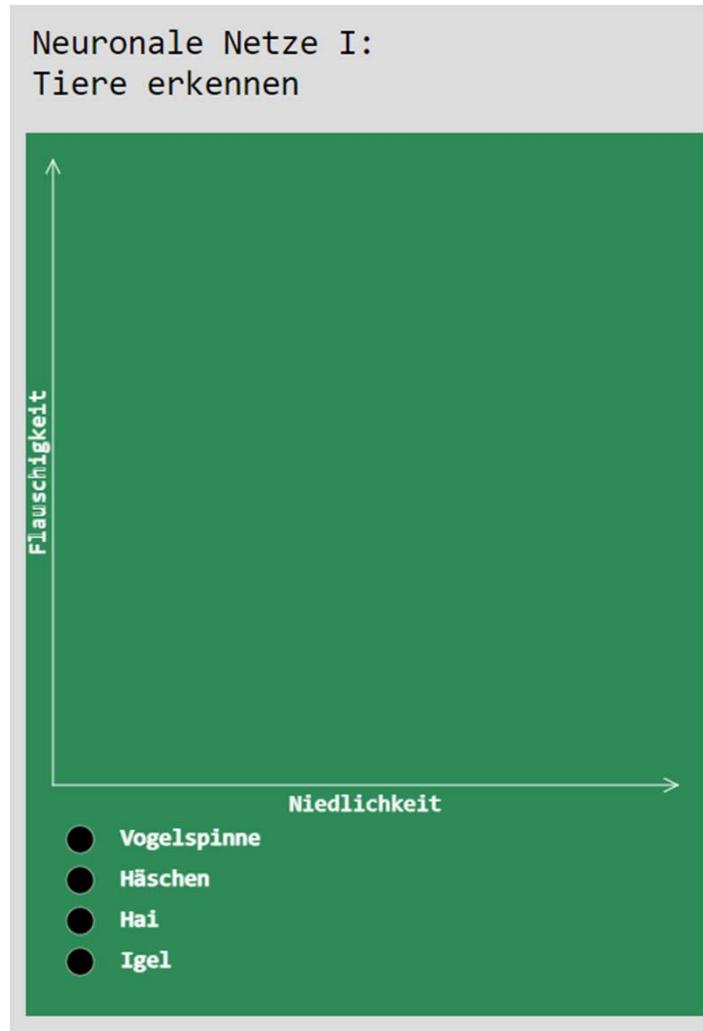
Mittwoch  
**Maschinelles Lernen & Vorhersagemodelle**

Grundlagen von  
Machine Learning:  
Algorithmen und deren  
Anwendungen

Neuronale Netze:  
Grundkonzepte und  
Anwendungsbeispiele

Übungen

# Einführung in Machine Learning: Neuronale Netze



# Einführung in Machine Learning: Neuronale Netze

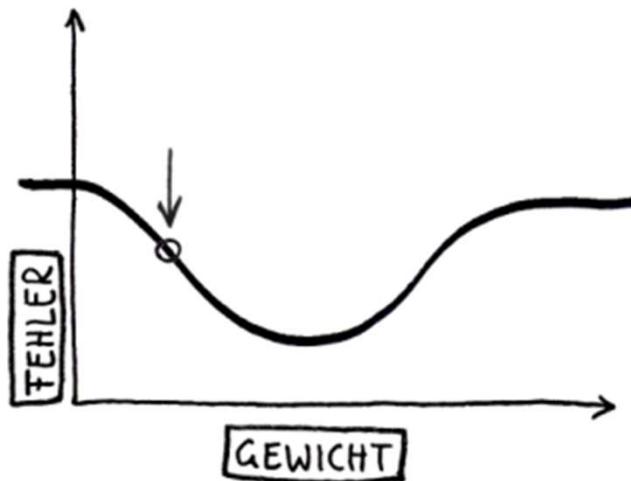
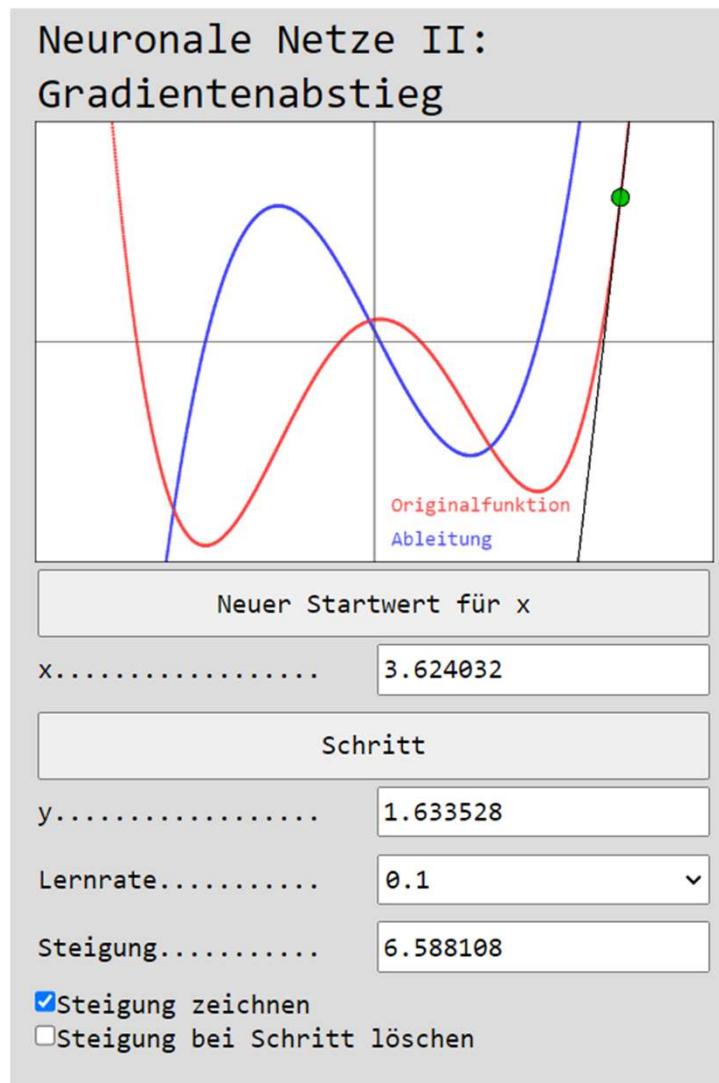


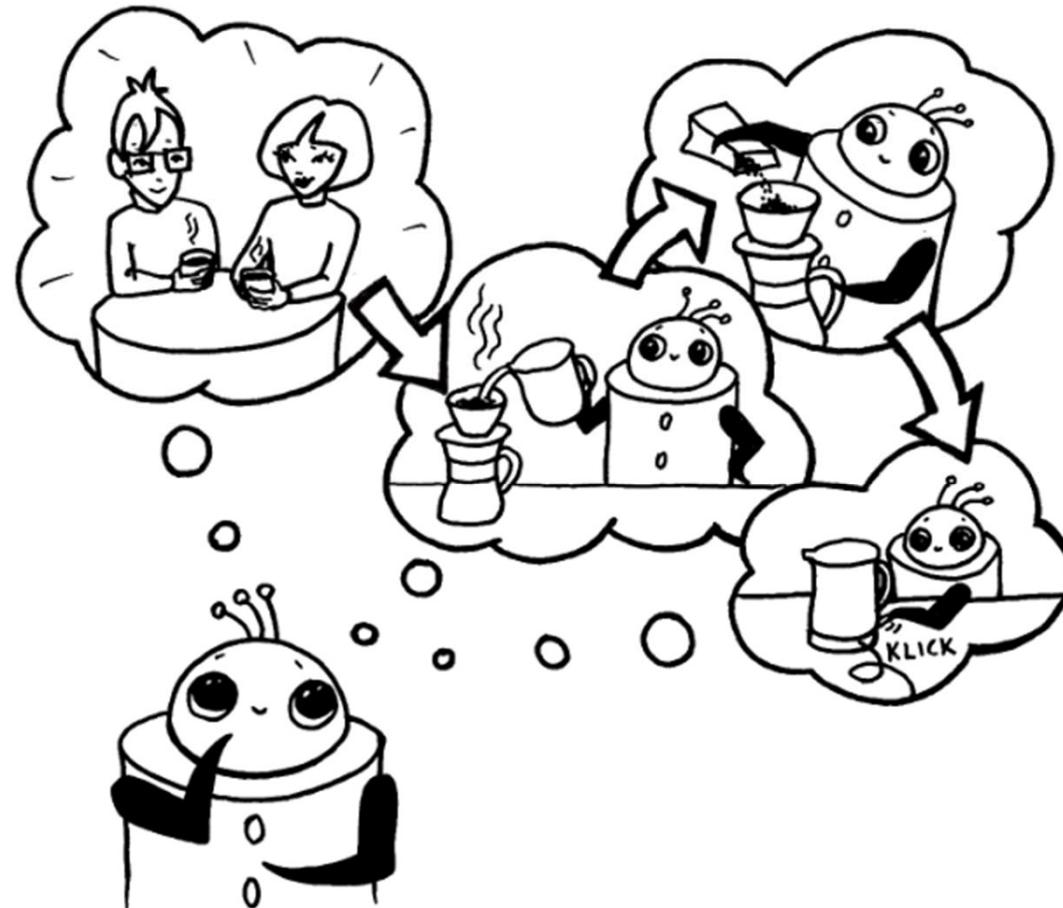
Abbildung 11.4 Eine mathematische Funktion, die ein Gewicht auf den Fehler abbildet; der Pfeil zeigt die aktuelle Position des Gewichts.



# Einführung in Machine Learning: Neuronale Netze



# Neuronale Netze: Backpropagation (dt. Fehlerrückführung)



# Neuronale Netze: Backpropagation (dt. Fehlerrückführung)

## **Backpropagation:**

- Weitergabe des Fehlers von der Ausgabeschicht zurück durch das Netzwerk
- Schrittweise Anpassung der Gewichte

## **Fehlerberechnung:**

- Unterschied zwischen erwarteter und tatsächlicher Ausgabe
- Berechnung des Einflusses eines Gewichts auf den Fehler durch eine Funktion
- Minimierung des Fehlers mittels Gradientenabstieg

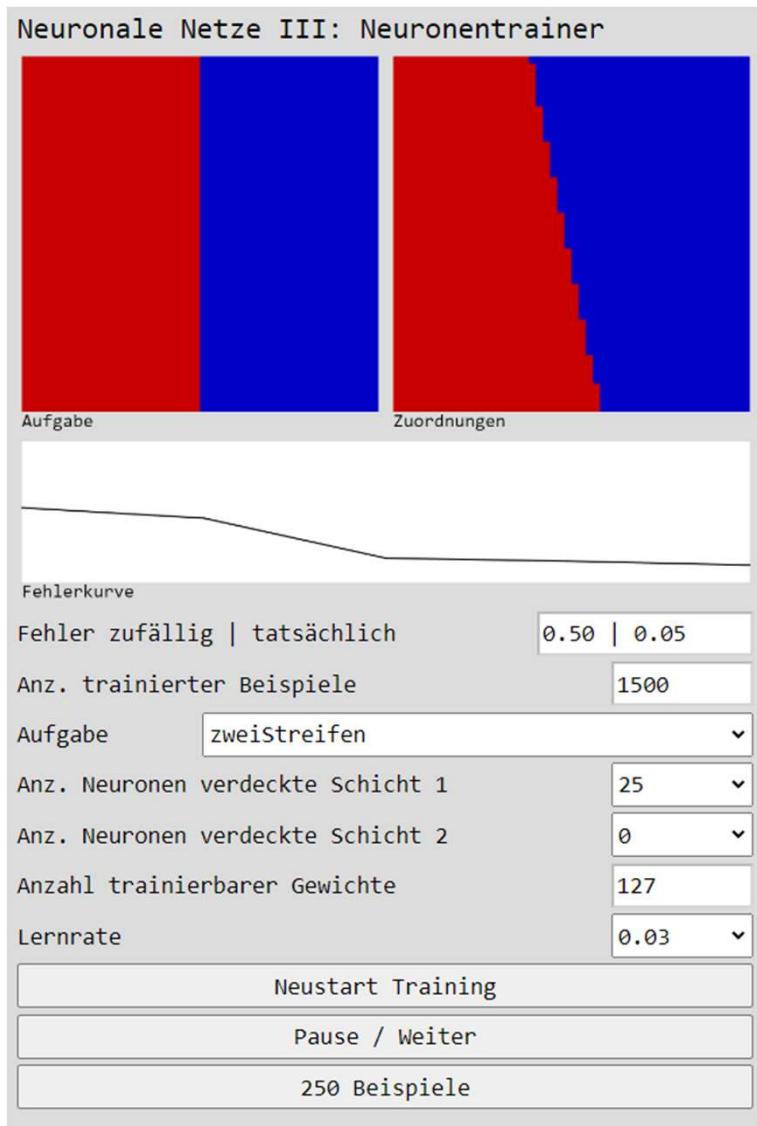
## **Herausforderungen bei mehrschichtigen Netzen:**

- Berechnungen bei vielen Neuronen aufwendig
- Zusammenhang zwischen Gewichten/Bias der Neuronen und dem Fehler in der Ausgabeschicht einfacher
- Berechnungen in tieferen Schichten komplexer (Ableitungen von Ableitungen)

## **Vereinfachungstrick:**

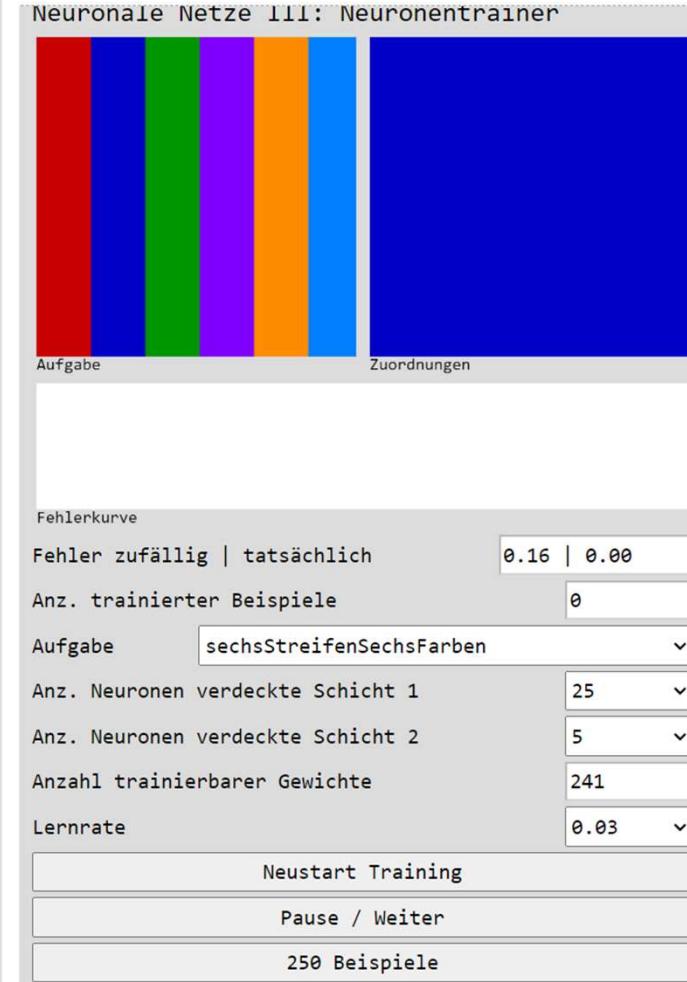
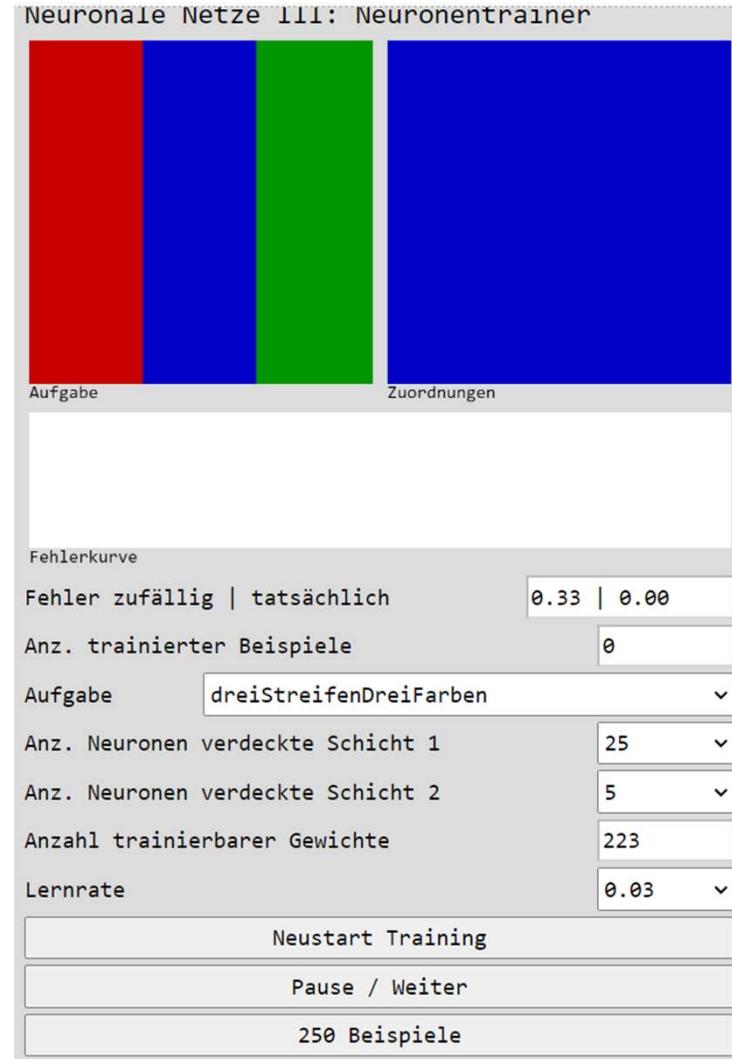
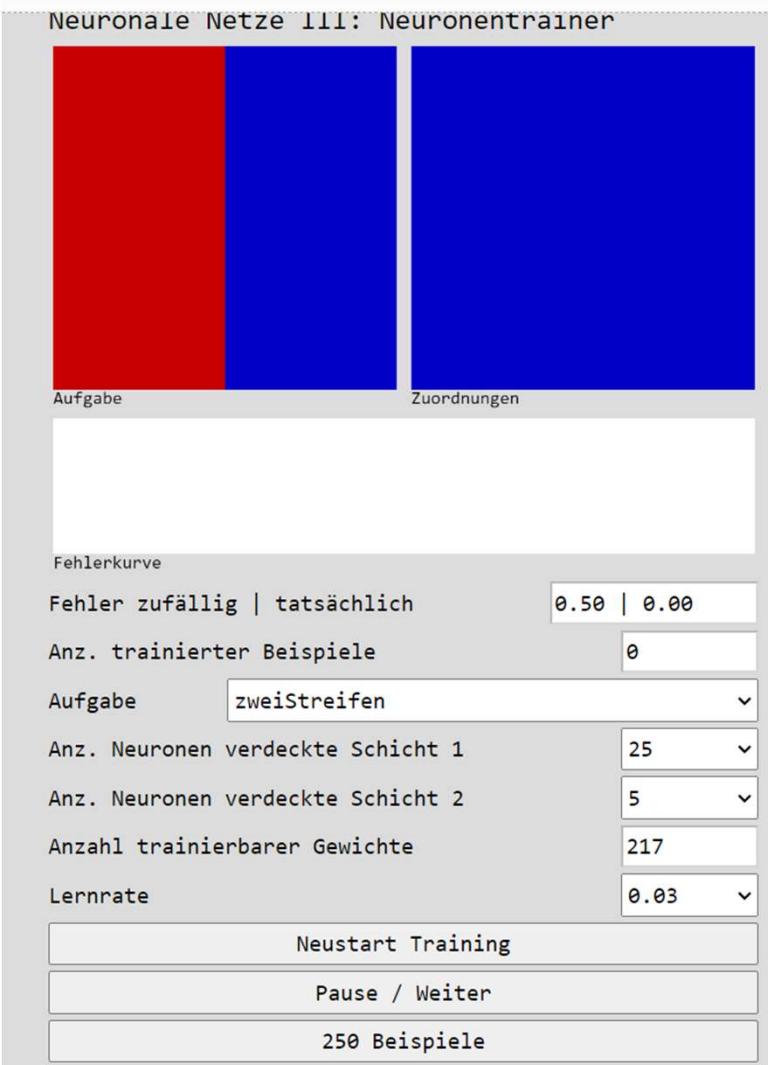
- Berechnung eines Gradienten für jedes Gewicht, bezogen auf die Ausgabe des jeweiligen Neurons
- Schrittweise Berechnung und Kombination der Gradienten entgegen der Flussrichtung
- Ergebnis: Gradienten, die sich tatsächlich auf den Fehler beziehen

# Einführung in Machine Learning: Neuronale Netze



Quelle: Künstliche Intelligenz verstehen – Eine Spielerische Einführung, 2024

# Einführung in Machine Learning: Neuronale Netze



### Data Input

[Generate](#) Import Draw

Data points 10000

Sample Circle

Data range

```
a ∈ [-5, 5]
b ∈ [-5, 5]
```

```
Function (a, b) {
// js
let isIncircle = a**2 + b**2 < 9;
return isIncircle ? 1 : -1;
}
```

### Training the model

Epoch : 0000

[Randomize](#)

1 Hidden layers [-](#) [+](#)

TANH [-](#) [+](#)

B 0.19 TANH

<https://akarzazi.github.io/neural-network-playground/>

### Data visualization

5  
b  
-5

-5 a 5

5  
b  
-5

-5 a 5

1  
Label  
-1

### Prediction visualization

5  
b  
-5

-5 a 5

5  
b  
-5

-5 a 5

### Loss view

Test loss 0.78  
Train loss no value

no value

View Range Full Epochs

-0.55 0.78

## Data Input

Generate Import Draw

Data points 10000

Sample Circle

Data range

a ∈ [-5, 5]  
b ∈ [-5, 5]

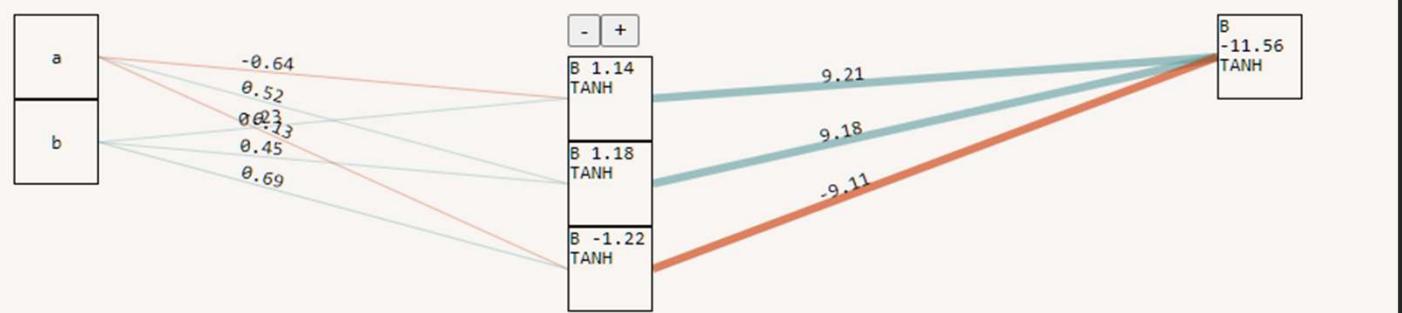
```
Function (a, b) {  
// js  
let isIncircle = a**2 + b**2 < 9;  
return isIncircle ? 1 : -1;  
}
```

## Training the model

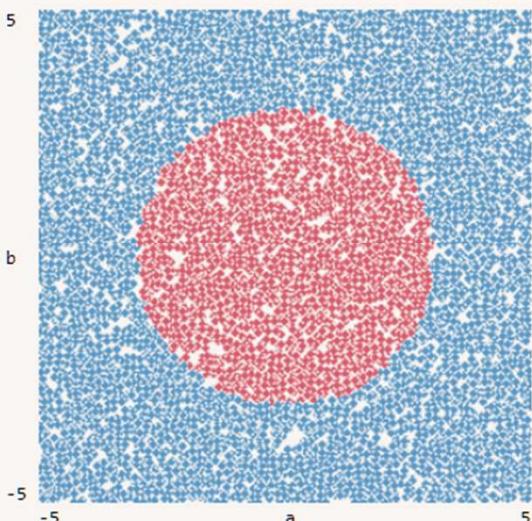


Epoch : 1051

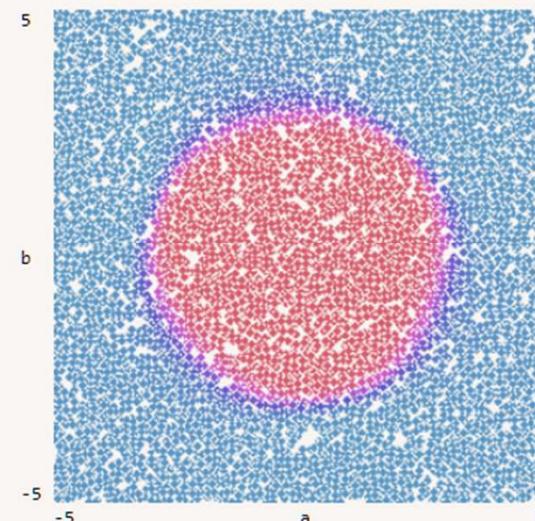
Randomize



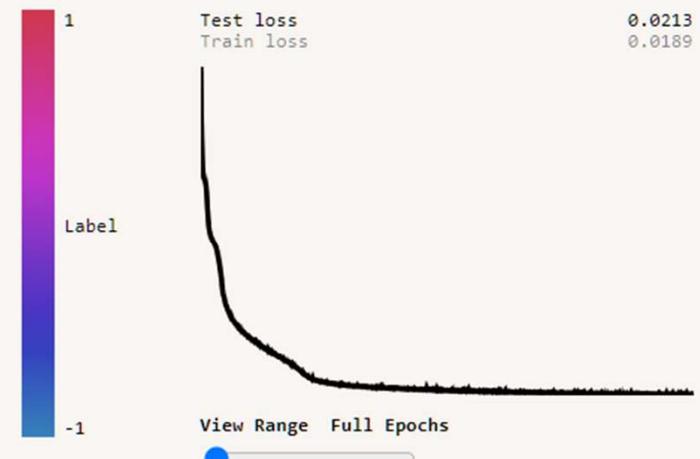
## Data visualization



## Prediction visualization



## Loss view



# Gängige Missverständnisse künstlicher neuronaler Netzwerke?

1

Künstliche neuronale Netzwerke arbeiten wie das menschliche Gehirn?

2

Künstliche Neuronale Netze können jede Art von Problem lösen?

3

Künstliche Neuronale Netze sind objektiv und unvoreingenommen?

4

Künstliche Neuronale Netze sind vollständig selbstlernend?

5

Einmal trainiert, funktionieren Künstliche neuronale Netze fehlerfrei?

# Neuronale Netze: Übungen

## Zusammenfassung

- Tiere erkennen
- Gradienten
- Backpropagation
- Gängige Missverständnisse

**Mittwoch  
Maschinelles Lernen &  
Vorhersagemodelle**

Grundlagen von  
Machine Learning:  
Algorithmen und deren  
Anwendungen

Neuronale Netze:  
Grundkonzepte und  
Anwendungsbeispiele

Übungen