

# DATA SCIENCE 2

VORLESUNG - NoCODE

PROF. DR. CHRISTIAN BOCKERMANN

HOCHSCHULE BOCHUM

WINTERSEMESTER 2022 / 2023

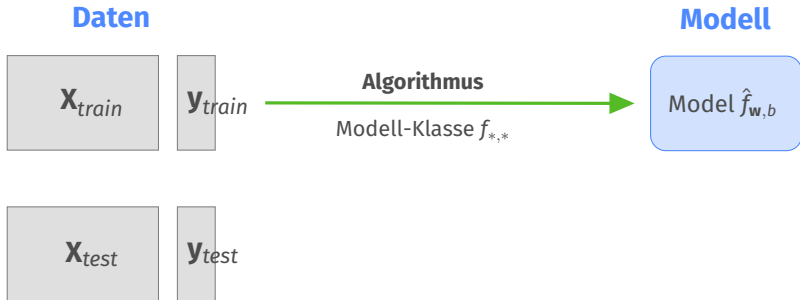
1 Datenanalyse mit Python

2 Weitere Software/Tools

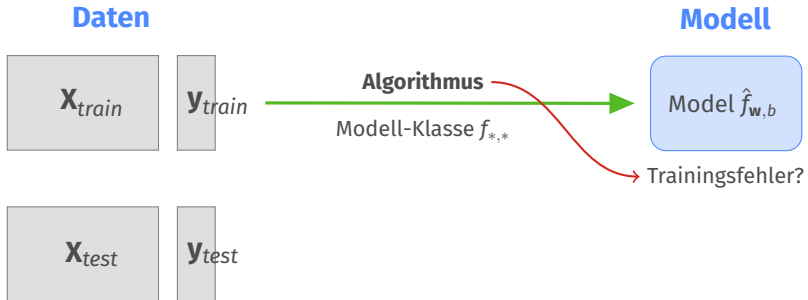
3 No-Code Ansätze

# Datenanalyse mit Python

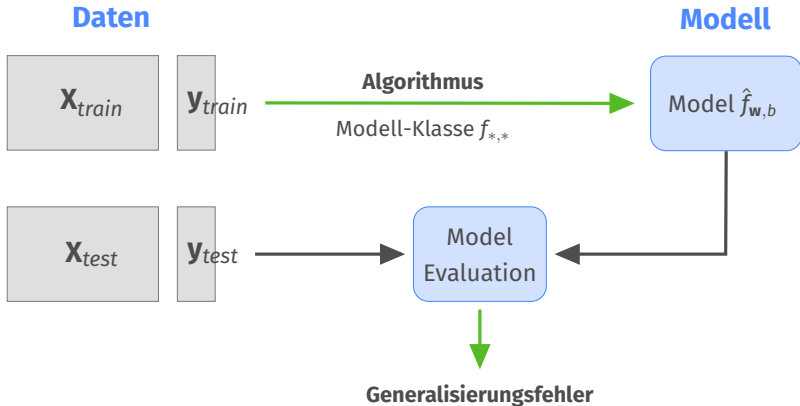
## Vorgehen beim überwachten Lernen



## Vorgehen beim überwachten Lernen



## Vorgehen beim überwachten Lernen



```
import pandas as pd

# read data from csv
df = pd.read_csv('daten.csv')
features = ['a1', 'a2', 'a3']

# Merkmale auswaehlen
X = df[features]
y = df['label']

# Daten aufteilen
X_tr, X_ts, y_tr, y_ts = train_test_split(X, y)

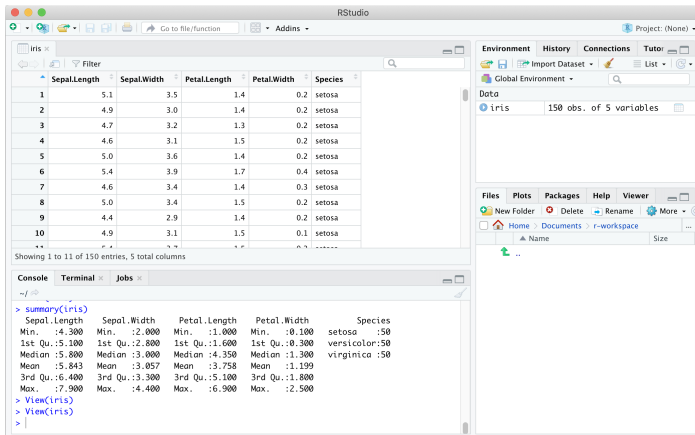
# Modell trainieren
m = DecisionTreeClassifier()
m.fit(X_tr, y_tr)
```

## Programmiersprachen

- Julia, <http://julialang.org>
- Python mit Pandas, SciKit Learn  
<http://scikit-learn.org>
- R, <http://www.r-project.org>



## Programmiersprache R für Statistik Aufgaben



The screenshot displays the RStudio interface with the following components:

- Environment:** Shows the 'iris' dataset with 150 observations and 5 variables.
- Data:** A preview of the first 10 rows of the 'iris' dataset.
- Files:** Shows the current workspace directory.
- Console:** Contains the following R code and its output:

```
> summary(iris)
  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa   :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicol.:50
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica.:50
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

> View(iris)
> View(iris)
>
```

**Abbildung:** RStudio Umgebung für die Sprache R.

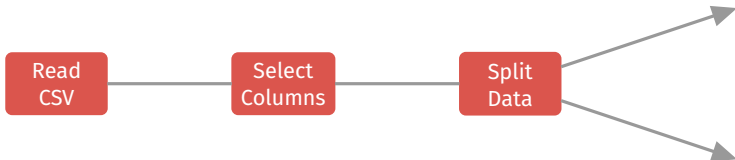
# No-Code Ansätze

## Trend: *No Code Tools*

- RapidMiner, <http://rapidminer.com>
- Knime, <http://www.knime.com>
- WEKA, MOA, <http://www.cs.waikato.ac.nz/ml/weka>
- Talend (Data Processing)

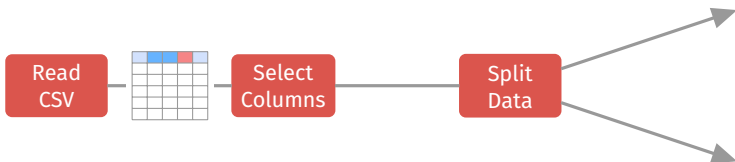
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

- Symbole für ausführbare Operationen
- Verbindungen zu Übergabe von Ergebnis-Objekten
- Einfache Start/Stopp Funktionen, Anhalten von Prozessen
- Möglichst ohne Programmierung auskommen



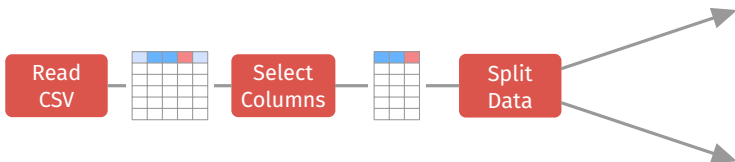
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

- Symbole für ausführbare Operationen
- Verbindungen zu Übergabe von Ergebnis-Objekten
- Einfache Start/Stopp Funktionen, Anhalten von Prozessen
- Möglichst ohne Programmierung auskommen



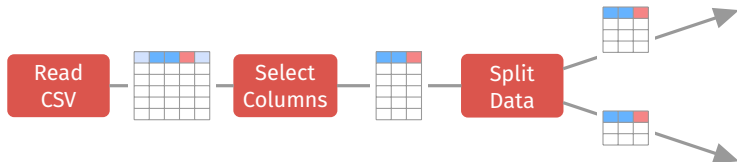
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

- Symbole für ausführbare Operationen
- Verbindungen zu Übergabe von Ergebnis-Objekten
- Einfache Start/Stopp Funktionen, Anhalten von Prozessen
- Möglichst ohne Programmierung auskommen



Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

- Symbole für ausführbare Operationen
- Verbindungen zu Übergabe von Ergebnis-Objekten
- Einfache Start/Stopp Funktionen, Anhalten von Prozessen
- Möglichst ohne Programmierung auskommen

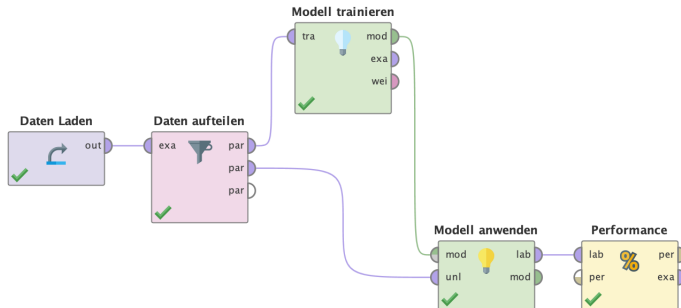


The screenshot displays the RapidMiner Studio Free 9.7.002 interface. The main workspace shows a workflow process with the following nodes: 'Daten Laden', 'Daten aufteilen', 'Modell trainieren', 'Modell anwenden', and 'Performance'. The 'Modell trainieren' node is selected, and its parameters are shown on the right: criterion (gain\_ratio), maximal depth (10), apply pruning (checked), confidence (0.1), apply prepruning (checked), minimal gain (0.01), and minimal leaf size (2). The bottom right shows a 'Help' section for the 'Decision Tree' operator, including a synopsis: 'This Operator generates a decision tree...'. The interface also includes a 'Repository' on the left, an 'Operators' list, and a 'Wisdom of Crowds' notification at the bottom.

**Abbildung:** Die graphische Schnittstelle von RapidMiner.



Prozesse werden als Graph mit vordefinierten Operator-Bausteinen gebaut



**Abbildung:** Ein Prozeß als Graph in RapidMiner.

RapidMiner wurde als OpenSource Tool am Lehrstuhl für künstliche Intelligenz der TU Dortmund entwickelt

- Prozess-Definition für ETL, Modellierung und Auswertung
- Einfaches Inspizieren / Exploration von Daten
- Enterprise Version für Unternehmen verfügbar
- Marktplatz mit Vielzahl von Erweiterungen
- *Wisdom of the crowds* Ansatz für schnellen Start

# KNIME ist ebenfalls ein graphisches Tool für Prozess-Design

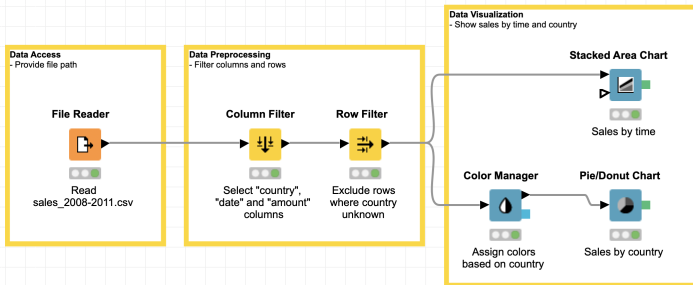
The screenshot displays the KNIME Analytics Platform interface. The main workspace shows a workflow titled "Visual Analysis of Sales Data". The workflow consists of the following nodes:

- Data Access:** "Provide file path" (File Reader) - Reads "sales\_2008-2011.csv".
- Data Preprocessing:** "Filter columns and/or rows" (Column Filter and Row Filter).
  - Column Filter:** Select "country", "sales" and "amount" columns.
  - Row Filter:** Exclude rows where country is unknown.
- Data Visualization:** "Show sales by time and country".
  - Color Manager:** Assign colors based on country.
  - Stacked Area Chart:** Sales by time.
  - Pie/Donut Chart:** Sales by country.

The interface also includes a left sidebar with "KNIME Explorer" and "Node Repository", and a bottom panel with "Outline" and "KNIME Console". The console shows the following output:

```
*****
*** Welcome to KNIME Analytics Platform v4.2.2.v202009250800 ***
*** Copyright by KNIME AG, Zurich, Switzerland ***
*****
Log file is located at: /Users/chris/.knime-workspace/.metadata/knime/knime.log
WARN Color Manager 3:2 Column "income" has no nominal values set
WARN Decision Tree Predictor 3:4 DataColumnSpec already contains a colo
WARN Decision Tree Predictor 3:4 DataColumnSpec already contains a colo
WARN Decision Tree Predictor 3:4 DataColumnSpec already contains a colo
WARN Decision Tree Predictor 3:4 DataColumnSpec already contains a colo
```

**Abbildung:** Die graphische Schnittstelle von KNIME.



**Abbildung:** Ein Prozess zur Visualisierung mit KNIME.

## Demo Rapidminer