

# DATA SCIENCE 2

VORLESUNG - NoCode

PROF. DR. CHRISTIAN BOCKERMANN

HOCHSCHULE BOCHUM

SOMMERSEMESTER 2022

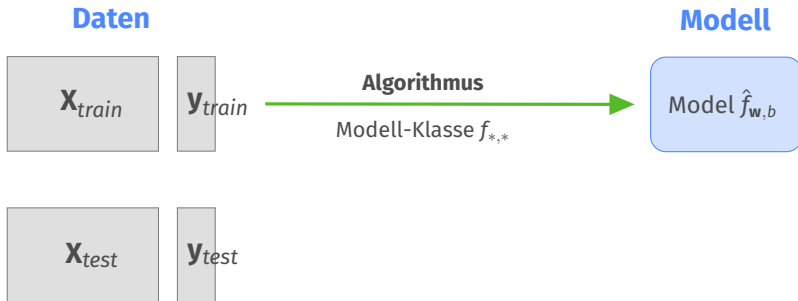
1 Datenanalyse mit Python

2 Weitere Software/Tools

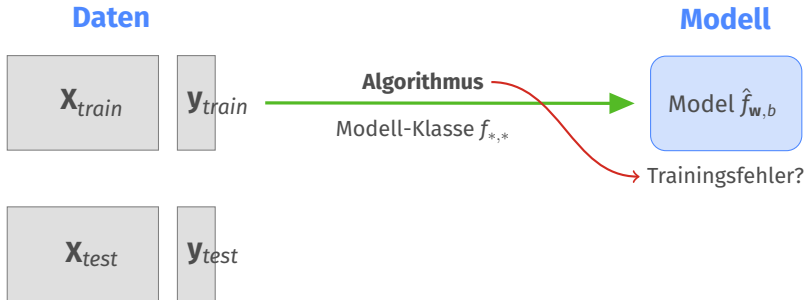
3 No-Code Ansätze

# Datenanalyse mit Python

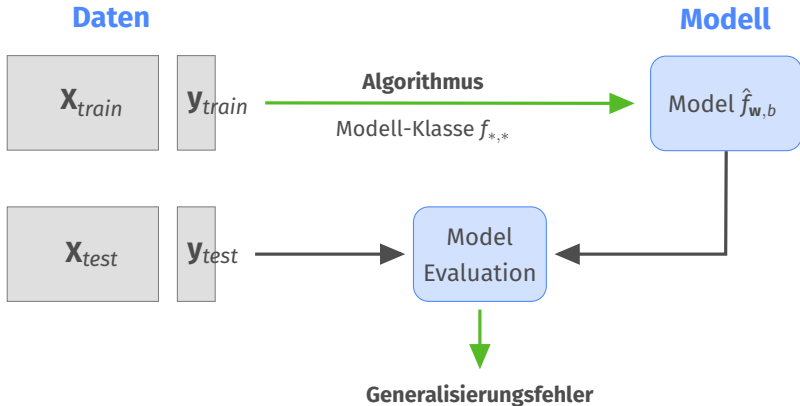
## Vorgehen beim überwachten Lernen



## 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)

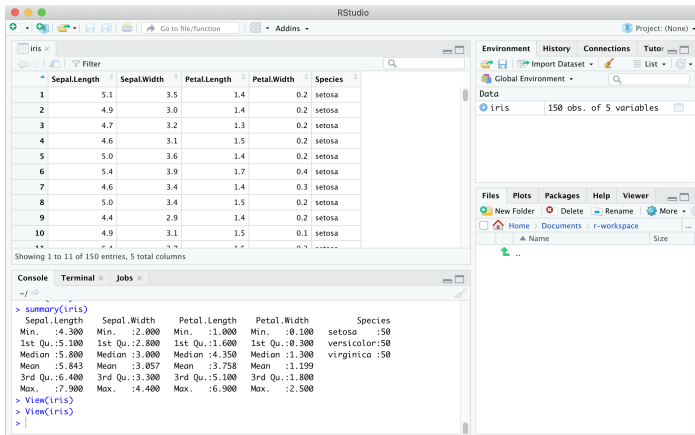
# Modell trainieren
m = DecisionTreeClassifier()
m.fit(X_tr, y_tr)
```

## 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 Table:** A preview of the first 10 rows of the 'iris' dataset.
- Console:** Shows the execution of the `summary(iris)` command and its output.

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species    |
|--------------|-------------|--------------|-------------|------------|
| 1            | 5.1         | 3.5          | 1.4         | 0.2 setosa |
| 2            | 4.9         | 3.0          | 1.4         | 0.2 setosa |
| 3            | 4.7         | 3.2          | 1.3         | 0.2 setosa |
| 4            | 4.6         | 3.1          | 1.5         | 0.2 setosa |
| 5            | 5.0         | 3.6          | 1.4         | 0.2 setosa |
| 6            | 5.4         | 3.9          | 1.7         | 0.4 setosa |
| 7            | 4.6         | 3.4          | 1.4         | 0.3 setosa |
| 8            | 5.0         | 3.4          | 1.5         | 0.2 setosa |
| 9            | 4.4         | 2.9          | 1.4         | 0.2 setosa |
| 10           | 4.9         | 3.1          | 1.5         | 0.1 setosa |

```
> 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   versicolor: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.

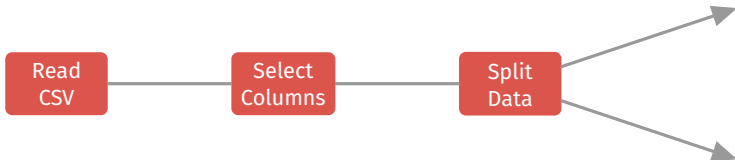
# No-Code Ansätze

## 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)

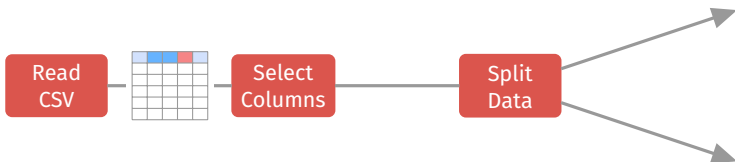
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

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



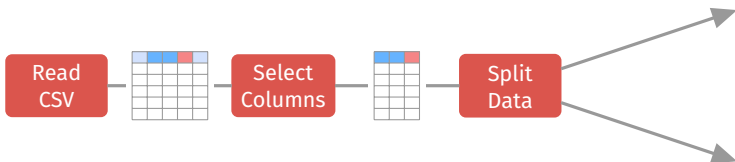
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

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



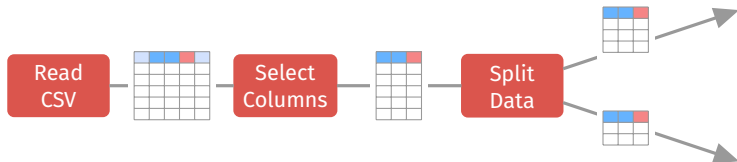
Werkzeuge um Prozesse mit graphischen Elementen zu entwerfen:

- 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

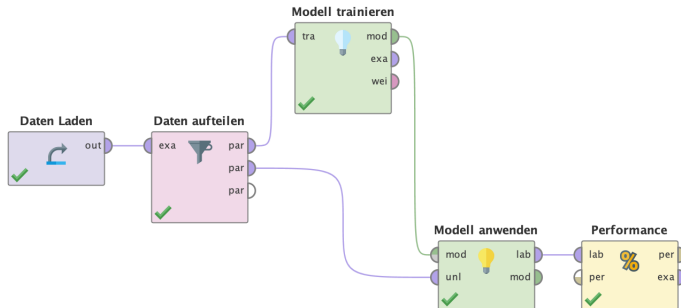


The screenshot displays the RapidMiner Studio Free 9.7.002 interface. The main workspace shows a workflow process with the following nodes: 'Daten Laden' (Data Load), 'Daten aufteilen' (Data Split), 'Modell trainieren' (Model Train), 'Modell anwenden' (Model Apply), and 'Performance'. The 'Modell trainieren' node is highlighted with an orange border. The 'Parameters' panel on the right shows settings for 'Modell trainieren (Decision Tree)', including '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 'Operators' panel on the left shows a search for 'performance' with results under 'Modeling (3)', 'Validation (20)', and 'Utility (1)'. The 'Help' panel at the bottom right provides information about the 'Decision Tree' operator, including its category (Supervised Classification) and a synopsis.

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



Prozesse werden als Graph mit vordefinierten Operator-Bausteinen gebaut

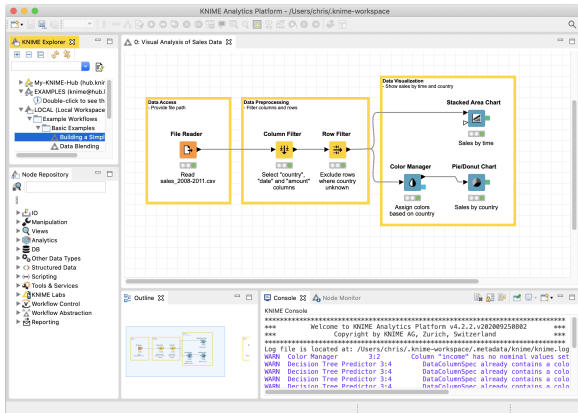


**Abbildung:** Ein Prozeß als Graph in RapidMiner.

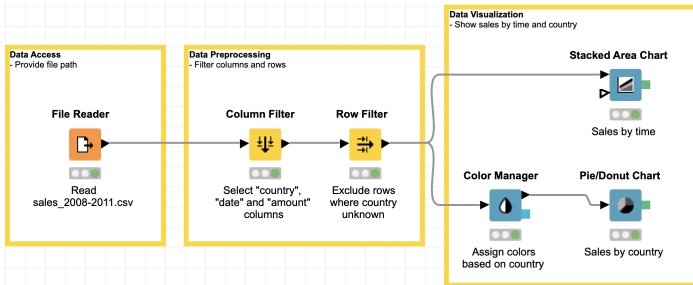
RapidMiner wurde als OpenSource Tool am Lehrstuhl für künstliche Intelligenz der TU Dortmund entwickelt

- 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



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



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

## Demo Rapidminer