Data Science 1

Vorlesung 6 - Intro

Prof. Dr. Christian Bockermann

Hochschule Bochum

Wintersemester 2021/2022

Wiederholung

Was geschah zuletzt? Gastvortrag

- Jonas Rashedi, Firma Douglas

Wiederholung

Was geschah zuletzt? Gastvortrag

- Jonas Rashedi, Firma Douglas

Und davor?

- Einfache Klassifikation
- Entscheidungsbäume, Lernen aus Daten

Wiederholung

Was geschah zuletzt? Gastvortrag

- Jonas Rashedi, Firma Douglas

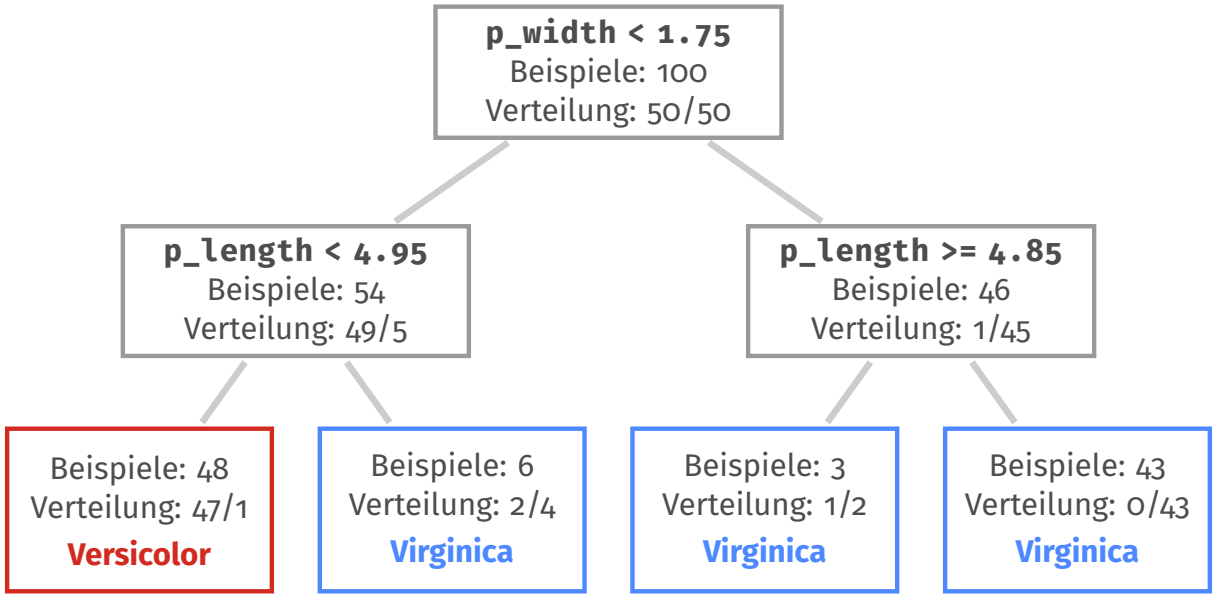
Und davor?

- Einfache Klassifikation
- Entscheidungsbäume, Lernen aus Daten

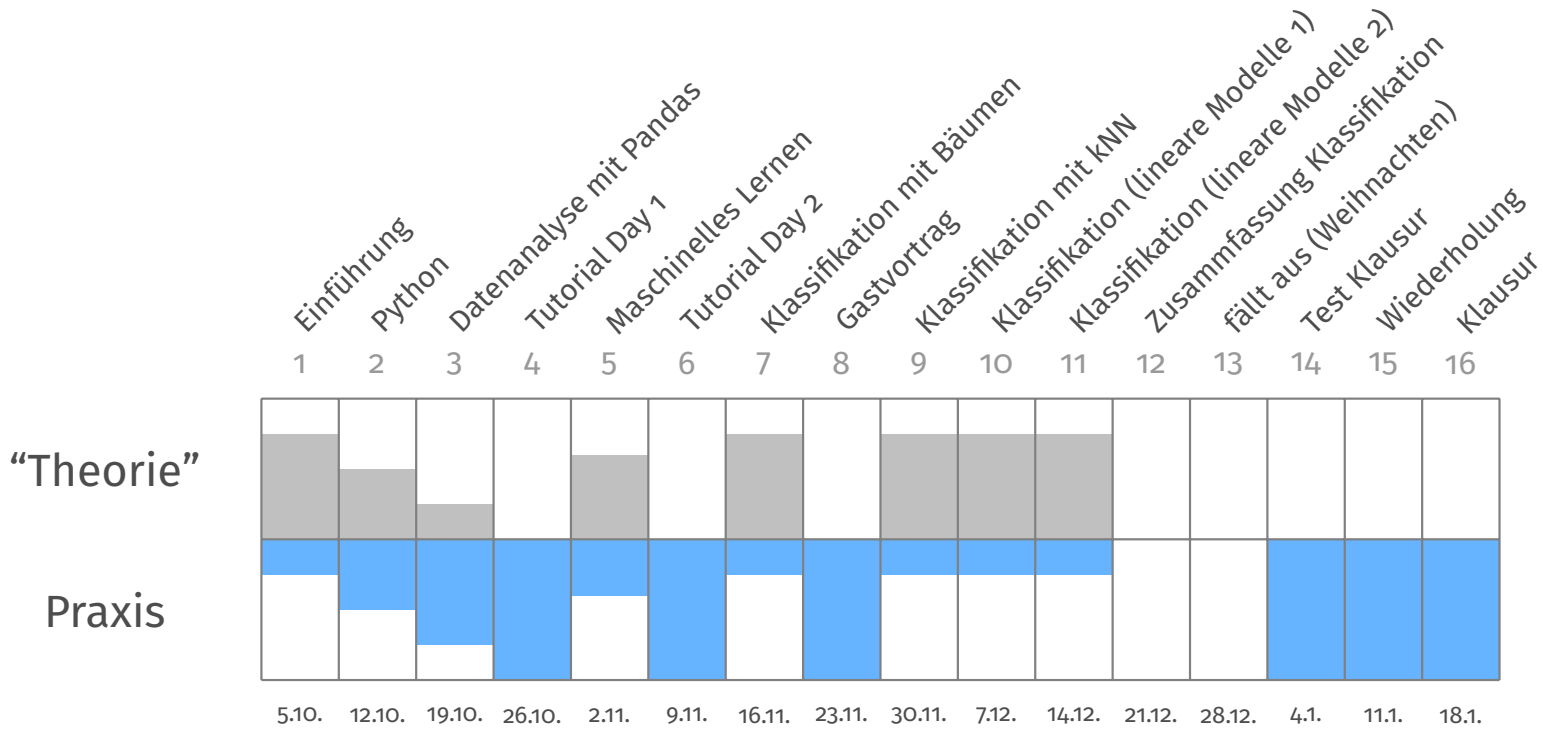
1. Model `m` (Baum) auf Trainingsdaten gelernt
2. Dann mit `m` Testdaten vorhergesagt
3. Vorhersage-Fehler bestimmt

Einfaches Modell: Entscheidungsbäume

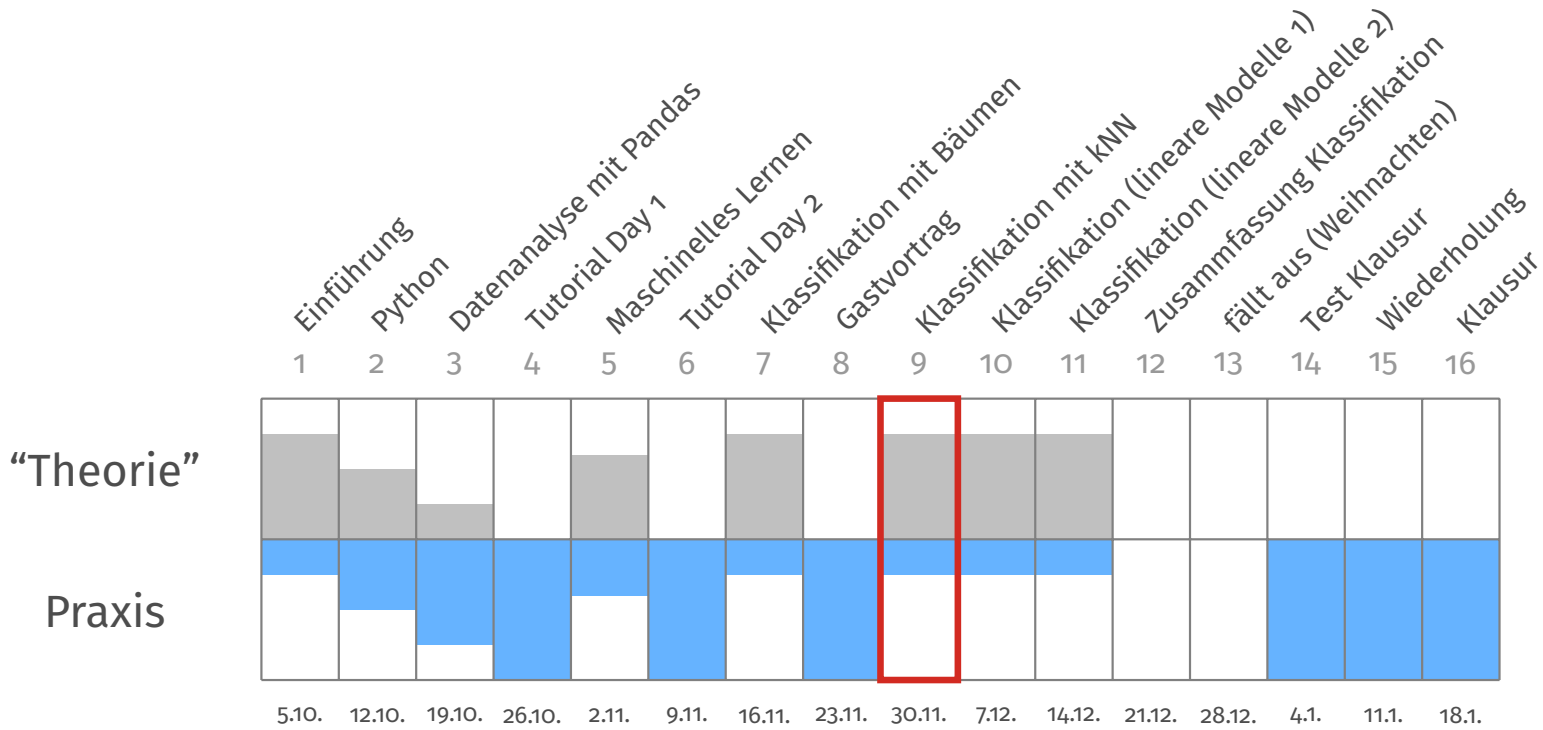
- Innere Knoten sind Entscheidungsknoten
- Blätter stellen Vorhersage dar



Wo sind wir heute (Vorlesung 6) ?

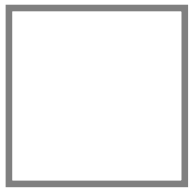


Wo sind wir heute (Vorlesung 6) ?



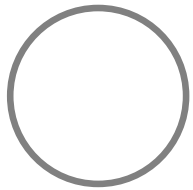
Menschliches Lernen nutzt Ähnlichkeiten aus

Zum Beispiel über Formen:



Quadrat

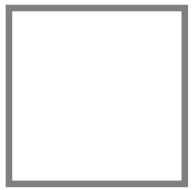
Rechteck



Kreis

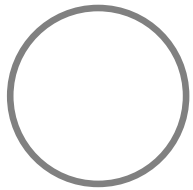
Menschliches Lernen nutzt Ähnlichkeiten aus

Zum Beispiel über Formen:



Quadrat

Rechteck



Kreis

Oder andere Eigenschaften (Größe, Gewicht):

Fussball

Handball

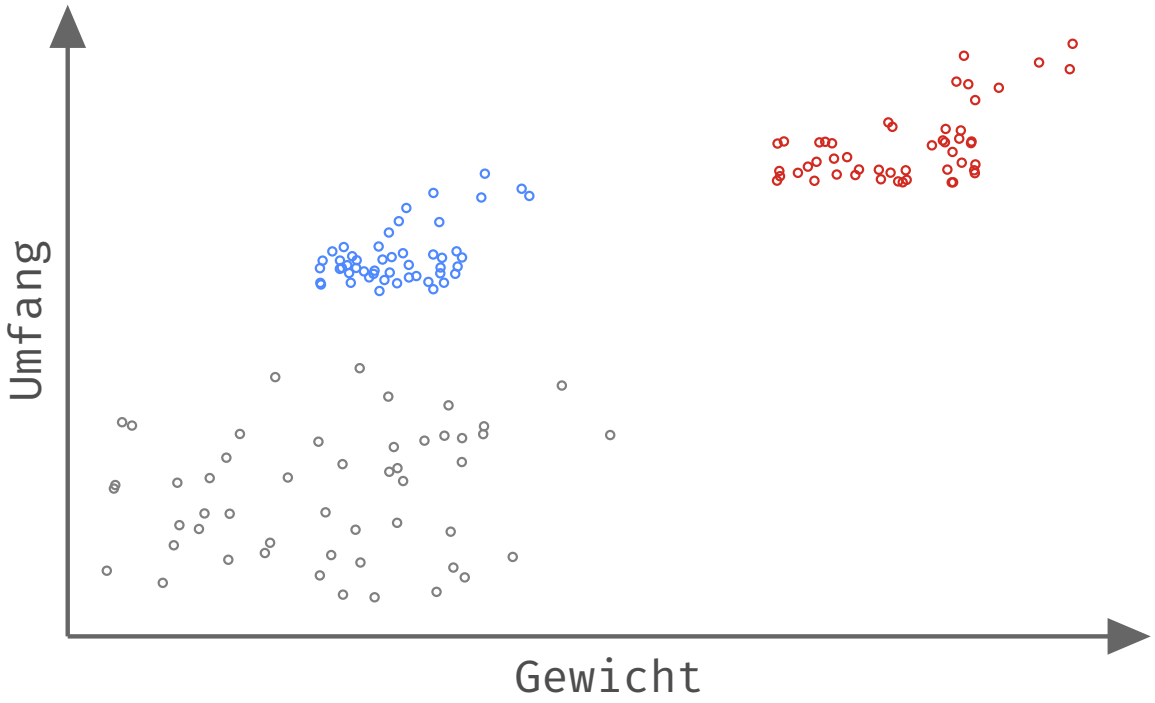
Basketball

Beispiel: Klassifikation von Bällen

Wir wollen Bälle ihrer Sportart zuordnen (Klassifikationsaufgabe)

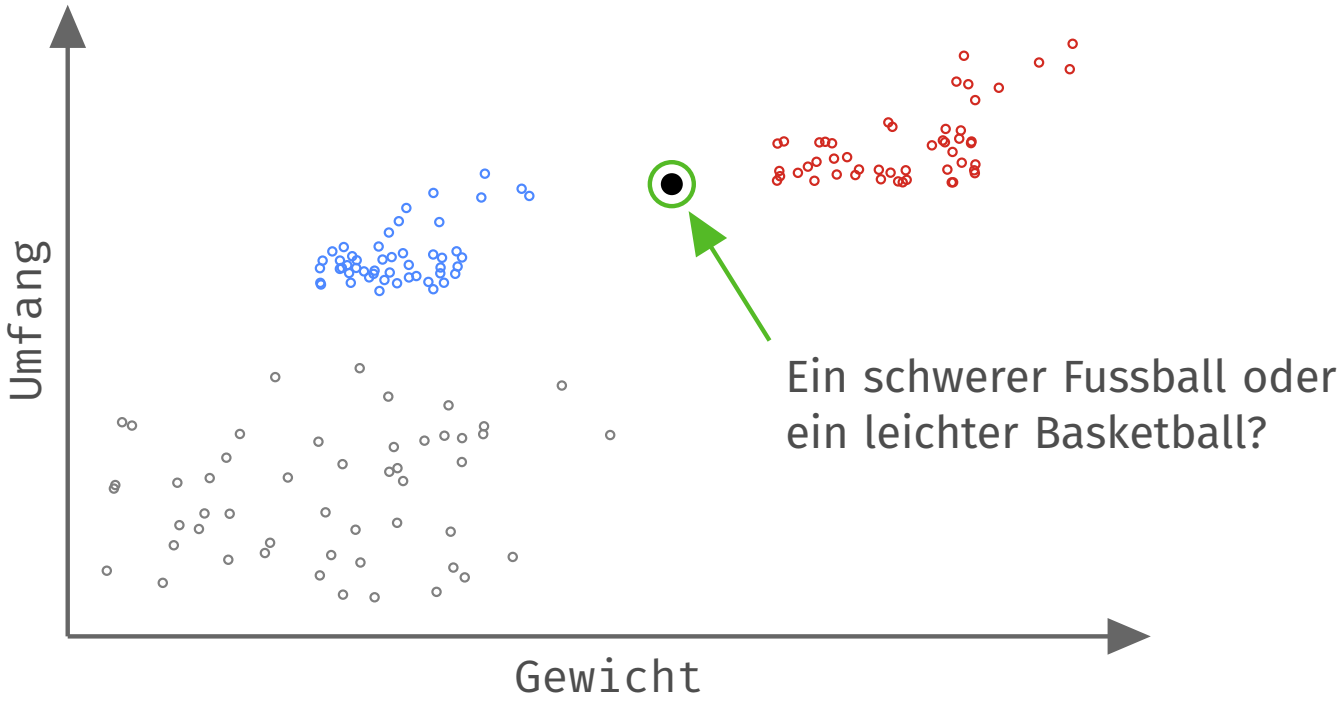
Umfang (cm)	**Gewicht** (g)	**Sportart**
70.29	444.30	Fussball
77.73	647.53	Basketball
53.34	427.07	Handball
57.09	406.12	Handball
68.28	440.96	Fussball
80.38	648.94	Basketball

Beispiel: Maße unterschiedlicher Bälle



Verschiedene Bälle nachgemessen: Fussball, Handball und Basketball

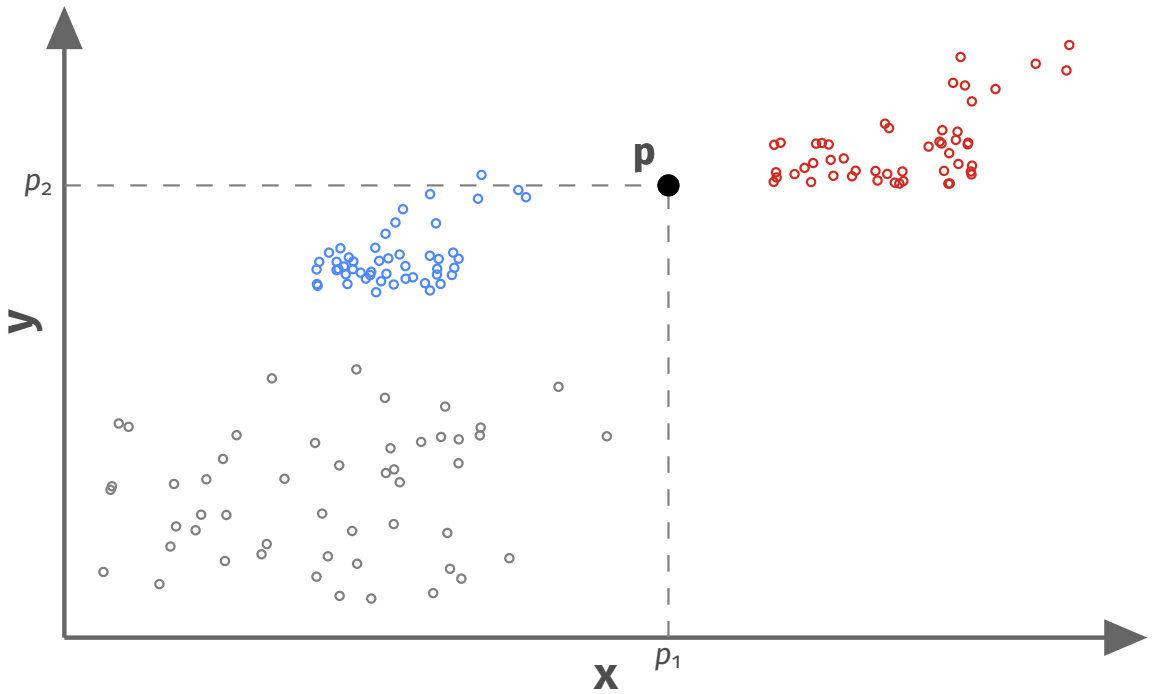
Beispiel: Maße unterschiedlicher Bälle



Verschiedene Bälle nachgemessen: Fussball, Handball und Basketball

Beispiel: Ball-Arten

Betrachte 2-dimensionalen Raum: $\mathbb{R}^2$

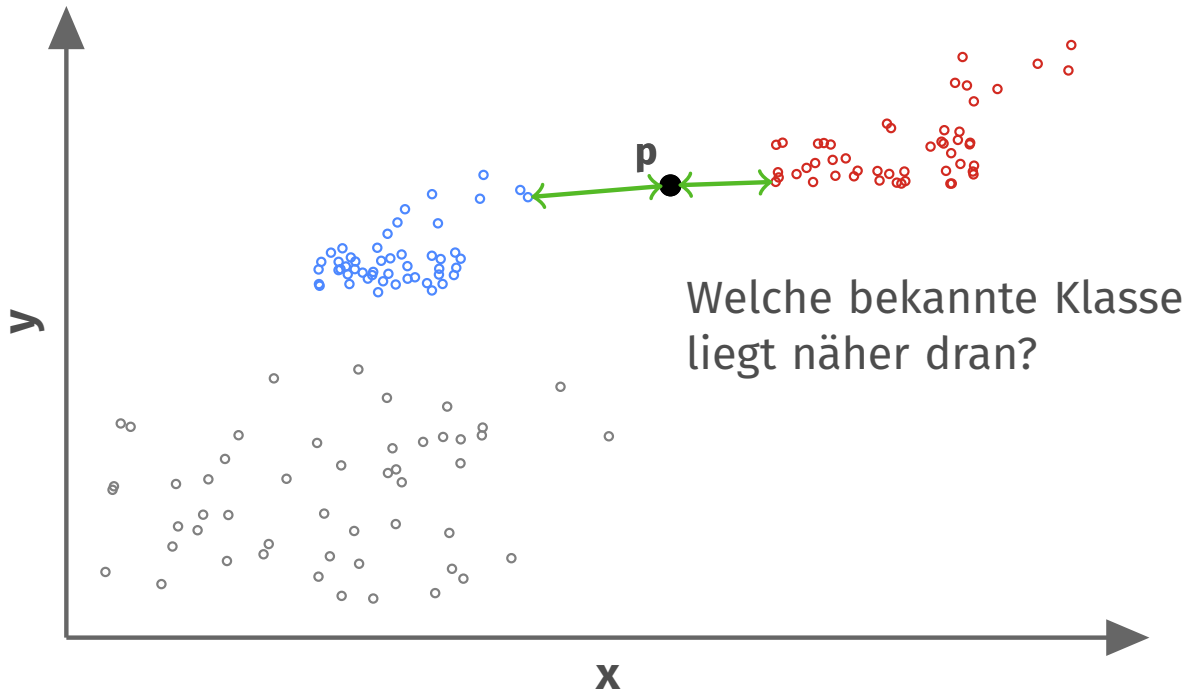


2-dimensionaler Raum: Jeder Punkt **p** besteht aus 2 Koordinaten:

$$\mathbf{p} = (p_1, p_2)$$

Beispiel: Ball-Arten

Betrachte 2-dimensionalen Raum: $\mathbb{R}^2$



Idee: Wir nutzen den Abstand als Ähnlichkeit und sagen die Klasse vorher, die am nächsten ist!

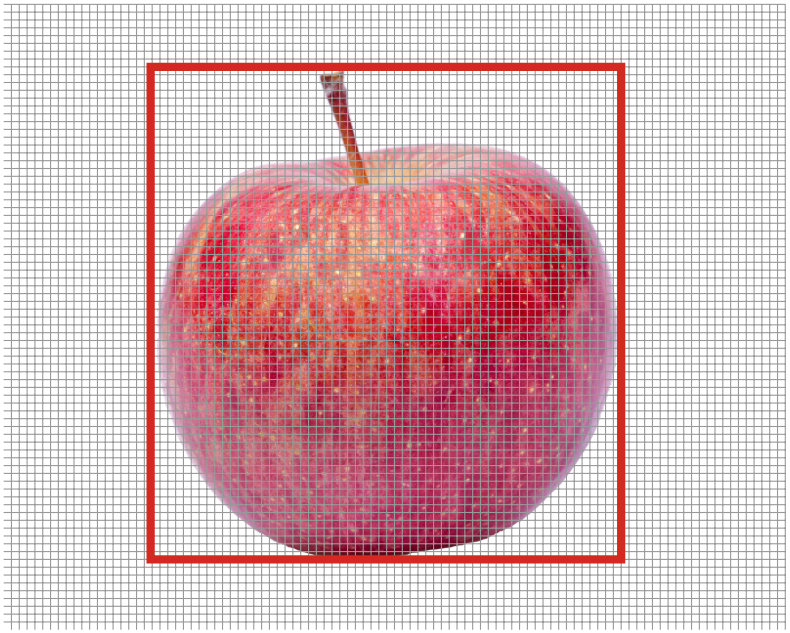
Beispiel: Obstwaage

Supermarkt-Innovation um 2008: Intelligente Waagen

- Die Waage erkennt das Obst/Gemüse per Kamera

https://www.spiegel.de/netzwelt/tech/supermarkt-technik-waage-erkennt-aufgelegtes-gemuese-a-569740.html

Wieder: Bild-Daten



- Objekt Pixel erkennen, Schwerpunkt berechnen
- Maße bestimmen, Formen ausprobieren?
- Mittleren Farb-Index berechnen

Bilder Datensatz: https://www.kaggle.com/moltean/fruits

Einfache Daten aus der Waage

Name	Gewicht	Breite	Höhe	Farbe
apple	192	8.4	7.3	0.55
apple	180	8.0	6.8	0.59
apple	176	7.4	7.2	0.6
mandarin	86	6.2	4.7	0.8
mandarin	84	6.0	4.6	0.79
mandarin	80	5.8	4.3	0.77

Einfache Daten aus der Waage

Name	Gewicht	Breite	Höhe	Farbe
apple	192	8.4	7.3	0.55
apple	180	8.0	6.8	0.59
apple	176	7.4	7.2	0.6
mandarin	86	6.2	4.7	0.8
mandarin	84	6.0	4.6	0.79
mandarin	80	5.8	4.3	0.77

Ähnlichkeit über Abstandsmaß - unterschiedliche Skalen!

Datensatz (48 Früchte): Kurse/DataScience1/data/fruits_with_colors.csv

Idee: Normalisierung der Attribute/Variablen

- Anpassung der Werte auf gleichen Wertebereich
- z.B. Skalierung jeder Spalte auf [0,1]

Min-Max-Normalisierung einer Variablen X

$$X' = \frac{X - min(X)}{max(X) - min(X)}$$

```
zaehler = df['Gewicht'] - min(df['Gewicht'])
nenner = max(df['Gewicht']) - min(df['Gewicht'])

df['Gewicht'] = zaehler / nenner
```

Alle Attribute/Variablen normalisieren:

- Normalisierung von allen numerischen Attributen
- Jedes Attribute für sich normalisieren

```
# Liste der numerischen Spalten:
#
spalten = ['Gewicht', 'Breite', 'Hoehe', 'Farbe']

for spalte in spalten:
  zaehler = df[spalte] - min(df[spalte])
  nenner = max(df[spalte]) - min(df[spalte])

  df[spalte] = zaehler / nenner
```

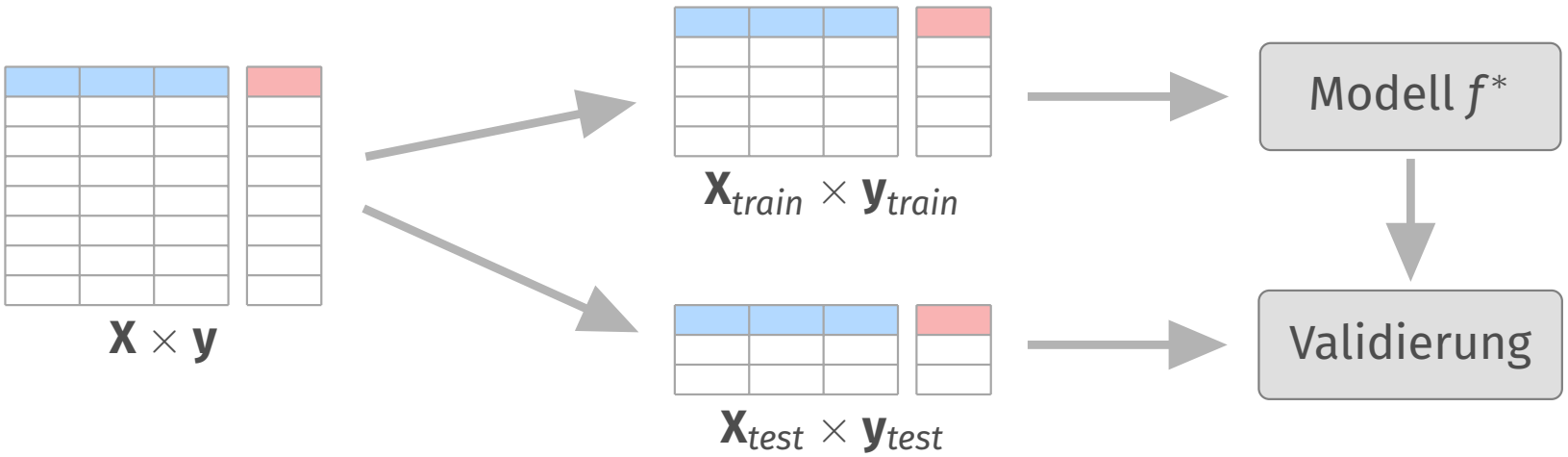
Diskussion: Wie genau müssen wir sein?

Diskussion: Wie genau müssen wir sein?

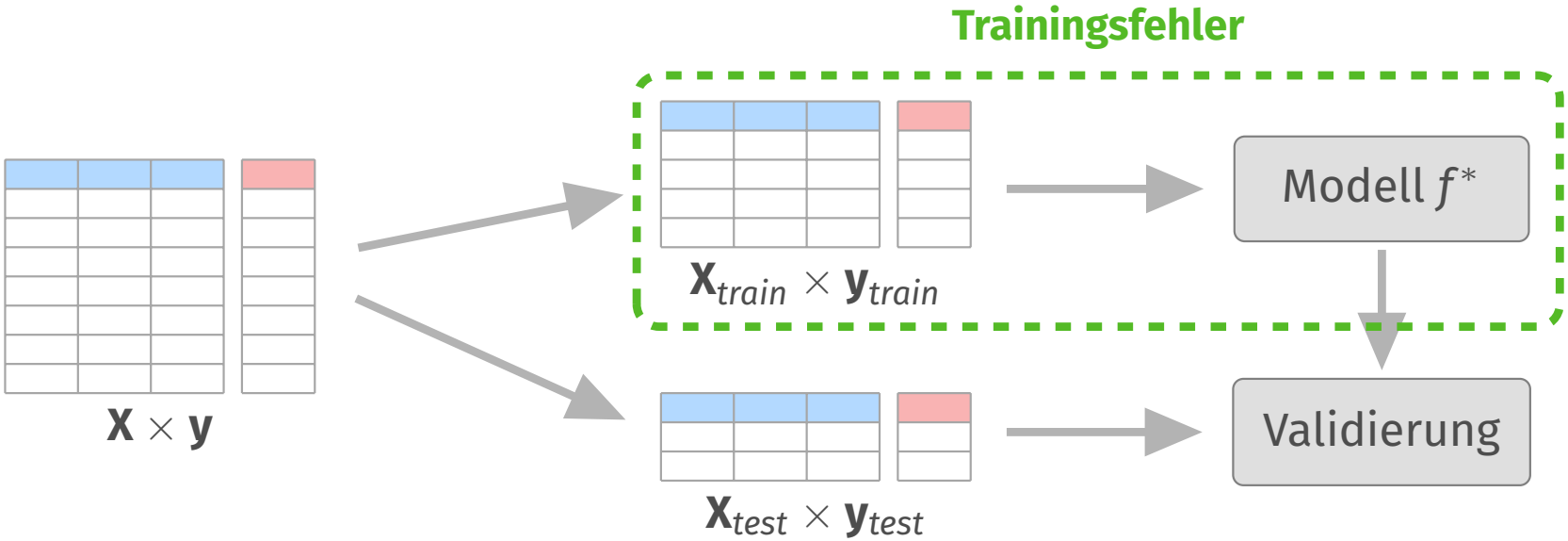
Diskussion: Wie genau müssen wir sein?

Diskussion: Wie genau müssen wir sein?

Diskussion: Wie genau müssen wir sein?

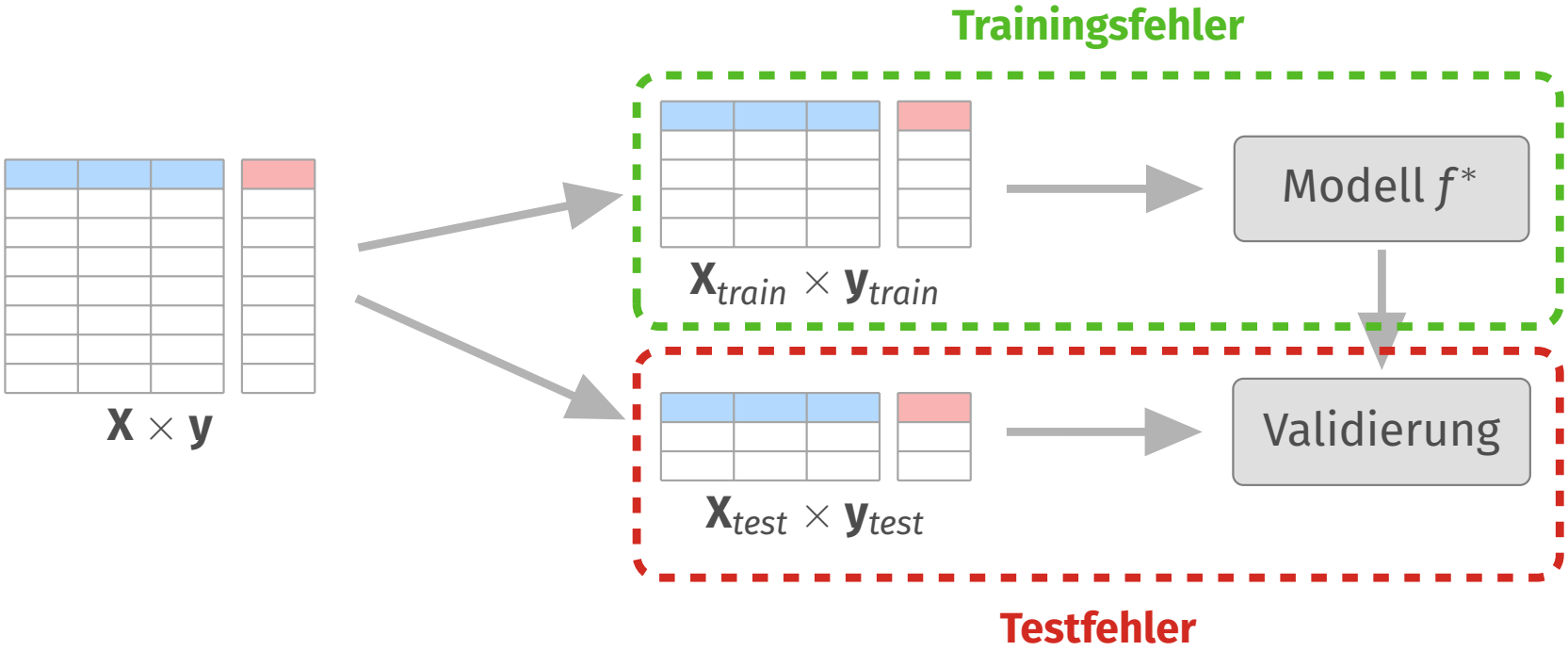


Diskussion: Wie genau müssen wir sein?



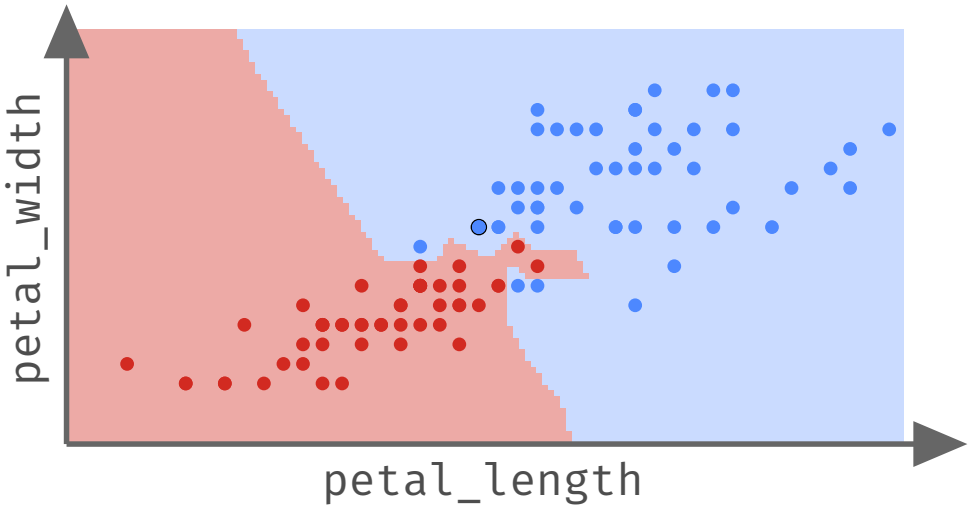
- Trainingsfehler: Modell an Trainingsdaten anpassen

Diskussion: Wie genau müssen wir sein?

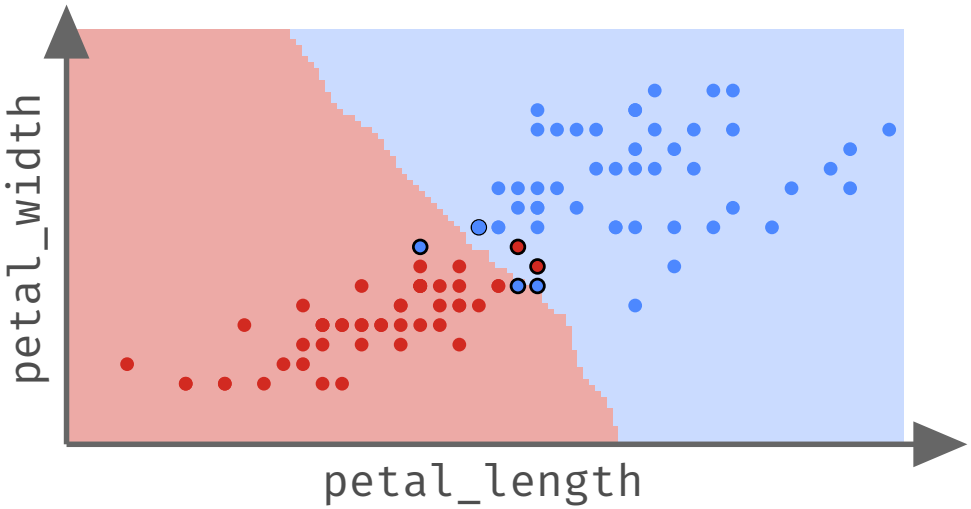


- Trainingsfehler: Modell an Trainingsdaten anpassen
- Testfehler: Modell auf unbekannten Daten (auch: Generalisierungsfehler)

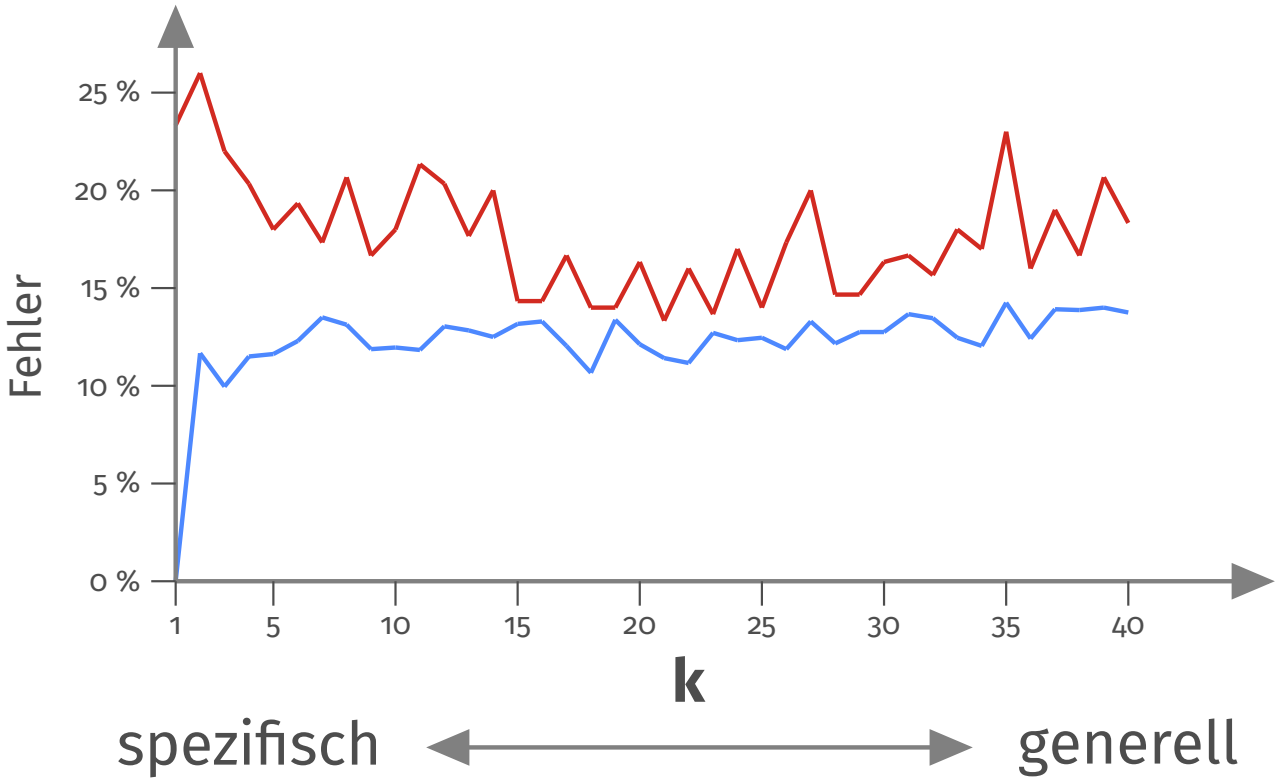
k=1



k=16



Training und Test-Fehler auf generiertem Datensatz (k-NN)



Overfitting

“Das Modell passt nur zu den Trainingsdaten.”

Overfitting

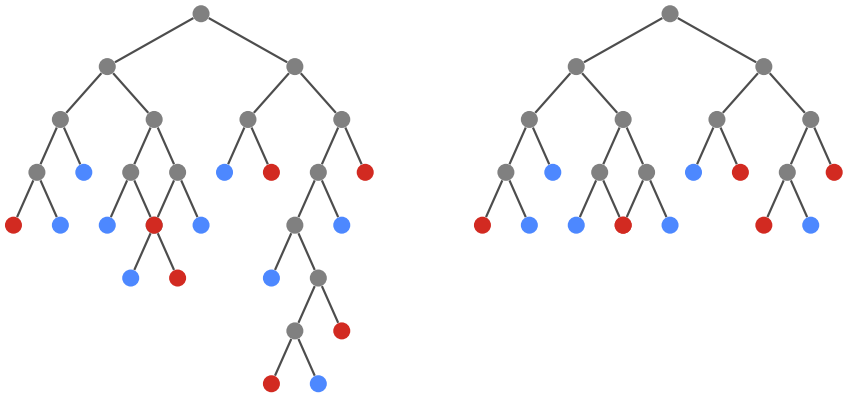
"Das Modell passt nur zu den Trainingsdaten."

	Trainingsfehler **klein**	Trainingsfehler **groß**
Testfehler **klein**	Das sieht gut aus!	
Testfehler **groß**	**Overfitting!**	Das Modell lernt nicht!?

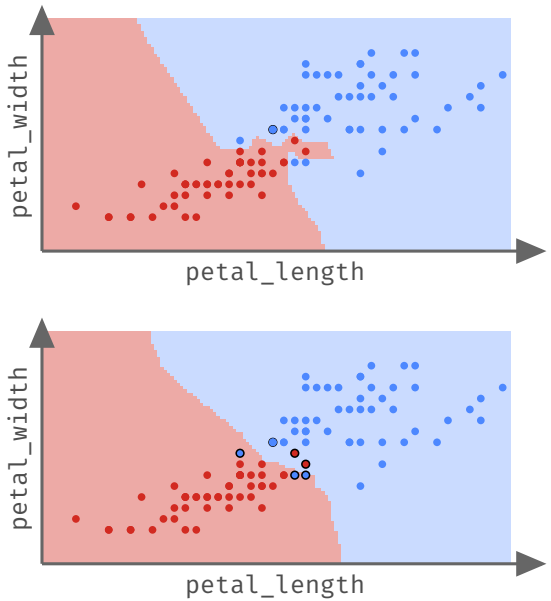
Overfitting - zu spezifisches Modell

- Modell zu sehr an die Trainingsdaten angepasst
- Vorhersage auf unbekannten Daten schlechter
- Modellkomplexität begrenzen (generelleres Modell)

Tiefe bei Bäumen beschränken



k bei k-NN erhöhen



A capella Song zum Thema Overfitting

https://youtu.be/DQWI1kvmwRg

Weitere Klassifikationsverfahren

- Klassifikation mit linearen Funktionen
- Vertiefung der Python-Kenntnisse

Wie geht's weiter?

Weitere Klassifikationsverfahren

- Klassifikation mit linearen Funktionen
- Vertiefung der Python-Kenntnisse

Tutorial Sessions

- Zusätzliche Übungsblätter mit einfachen Aufgaben
- Basics (Python) + andere Dinge (nach Bedarf)
- Kurze Slots (1 Std) im BBB für Unterstützung, z.B.
  - Mittwochs 11-12 Uhr
  - Donnerstags 11-12 Uhr