Data Science 1

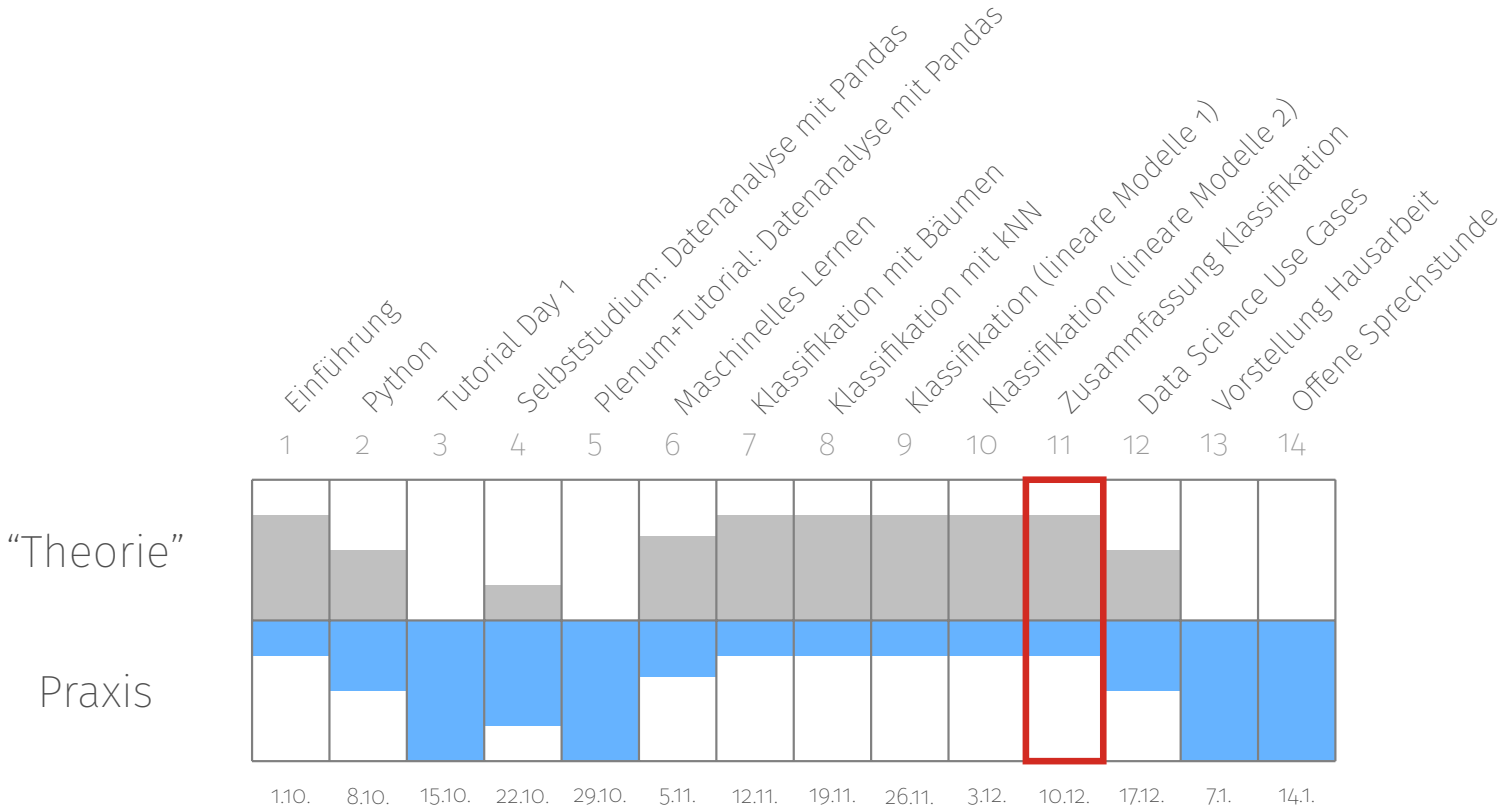
Vorlesung 9

Prof. Dr. Christian Bockermann

Hochschule Bochum

Wintersemester 2024/2025

Wo sind wir heute?



Inhalt

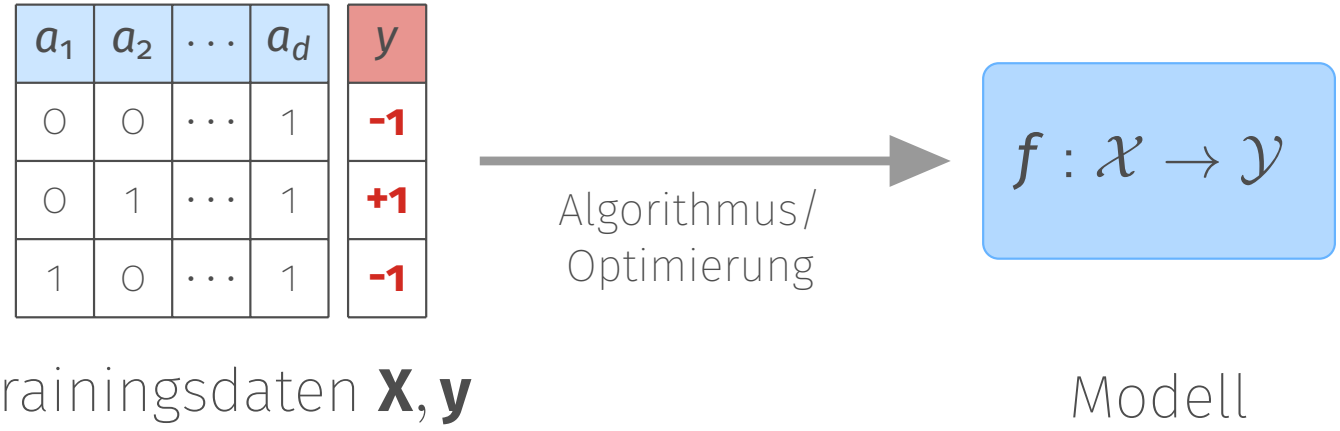
Zusammenfassung Klassifikation

Lern-Algorithmen erwarten Daten häufig in Form einer Tabelle:

	d Merkmale				
ID	a_1	a_2	$\cdots$	a_d	y
1	0	0	$\cdots$	1	**-1**
2	0	1	$\cdots$	1	**+1**
3	1	0	$\cdots$	1	**-1**

Beispiel $\mathbf{x_2} = (x_{a_1}, x_{a_2}, \ldots, x_{a_4}, y)$
$= (0, 1, \ldots, 1, \mathbf{+1})$

- Beispiele werden auch *examples* oder *instances* genannt
- Merkmale (engl. *features*) werden auch *attributes* oder *Variablen* (Statistik) bezeichnet



Trainingsdaten **X**, **y**

Modell

a_1	a_2	$\cdots$	a_d	y
0	0	$\cdots$	1	**-1**
0	1	$\cdots$	1	**+1**
1	0	$\cdots$	1	**-1**

Trainingsdaten **X**, **y**

Algorithmus/
Optimierung

$$f : \mathcal{X} \to \mathcal{Y}$$

Modell

a_1	a_2	$\cdots$	a_d	y
1	1	$\cdots$	0	**?**

Neue Daten **x**′,
y unbekannt

ML - Modell Training

a_1	a_2	$\cdots$	a_d	y
0	0	$\cdots$	1	**-1**
0	1	$\cdots$	1	**+1**
1	0	$\cdots$	1	**-1**

Trainingsdaten $\mathbf{X}, \mathbf{y}$

$f : \mathcal{X} \to \mathcal{Y}$

Modell

a_1	a_2	$\cdots$	a_d	y
1	1	$\cdots$	0	**?**

Neue Daten $\mathbf{x}'$,
y unbekannt

Vorhersage

$\hat{y} = f(\mathbf{x}')$

Klassifikation ordnet Beispielen diskreten Klassen zu

- Vorgegebene Klassen $\mathcal{Y} = \{C_1, \ldots, C_k\}$
- Gegeben Menge $\mathbf{X} \times \mathbf{y} \subset \mathcal{X} \times \mathcal{Y}$ bei der jedem Beispiel x_i die zugehörige Klasse zugeordnet ist: (x_i, y_i)
- Qualitätsfunktion $q : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$

Ziel:

- Finde Modell

$$f : \mathcal{X} \to \mathcal{Y},$$

das die Qualitätsfunktion optimiert.

Klassifikation ordnet Beispielen diskreten Klassen zu

- Vorgegebene Klassen $\mathcal{Y} = \{C_1, \ldots, C_k\}$
- Gegeben Menge $\mathbf{X} \times \mathbf{y} \subset \mathcal{X} \times \mathcal{Y}$ bei der jedem Beispiel x_i die zugehörige Klasse zugeordnet ist: (x_i, y_i)
- Qualitätsfunktion $q : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$

Ziel:

- Finde Modell

$$f : \mathcal{X} \to \mathcal{Y},$$

das die Qualitätsfunktion optimiert.

Lernen als Optimierungsproblem!

Beispiel: Klassifikation von Schwertlilien

- Klassen: $\mathcal{Y} = \{\text{setosa}, \text{versicolor}, \text{virginica}\}$
- Menge $\mathbf{X} \times \mathbf{y}$ mit 150 Beispiele mit Spalte "`species`"
- Qualitätsfunktion

$$q(\mathbf{X}\times\mathbf{y}, f) = \sum_{(x,y)\in\mathbf{X}\times\mathbf{y}} err(y, \underbrace{f(x)}_{=\hat{y}}), \quad err(y,\hat{y}) = \begin{cases} 0, & \text{falls } y = \hat{y} \\ 1, & \text{sonst.} \end{cases}$$

Beispiel: Klassifikation von Schwertlilien

- Klassen: $\mathcal{Y} = \{\text{setosa}, \text{versicolor}, \text{virginica}\}$
- Menge $\mathbf{X} \times \mathbf{y}$ mit 150 Beispiele mit Spalte "`species`"
- Qualitätsfunktion

$$q(\mathbf{X}\times\mathbf{y}, f) = \sum_{(x,y)\in\mathbf{X}\times\mathbf{y}} err(y, \underbrace{f(x)}_{=\hat{y}}), \quad err(y,\hat{y}) = \begin{cases} 0, & \text{falls } y = \hat{y} \\ 1, & \text{sonst.} \end{cases}$$

Funktion q zählt die Anzahl der Vorhersagefehler des Modells f auf der Menge X

Beispiel: Klassifikation von Schwertlilien

- Klassen: $\mathcal{Y} = \{\text{setosa}, \text{versicolor}, \text{virginica}\}$
- Menge $\mathbf{X} \times \mathbf{y}$ mit 150 Beispiele mit Spalte "`species`"
- Qualitätsfunktion

$$q(\mathbf{X}\times\mathbf{y}, f) = \sum_{(x,y)\in\mathbf{X}\times\mathbf{y}} err(y, \underbrace{f(x)}_{=\hat{y}}), \quad err(y,\hat{y}) = \begin{cases} 0, & \text{falls } y = \hat{y} \\ 1, & \text{sonst.} \end{cases}$$

Funktion q zählt die Anzahl der Vorhersagefehler des Modells f auf der Menge X

Ziel: Finde f^* mit minimalem $q(X, f)$

Beispiel: Klassifikation von Schwertlilien

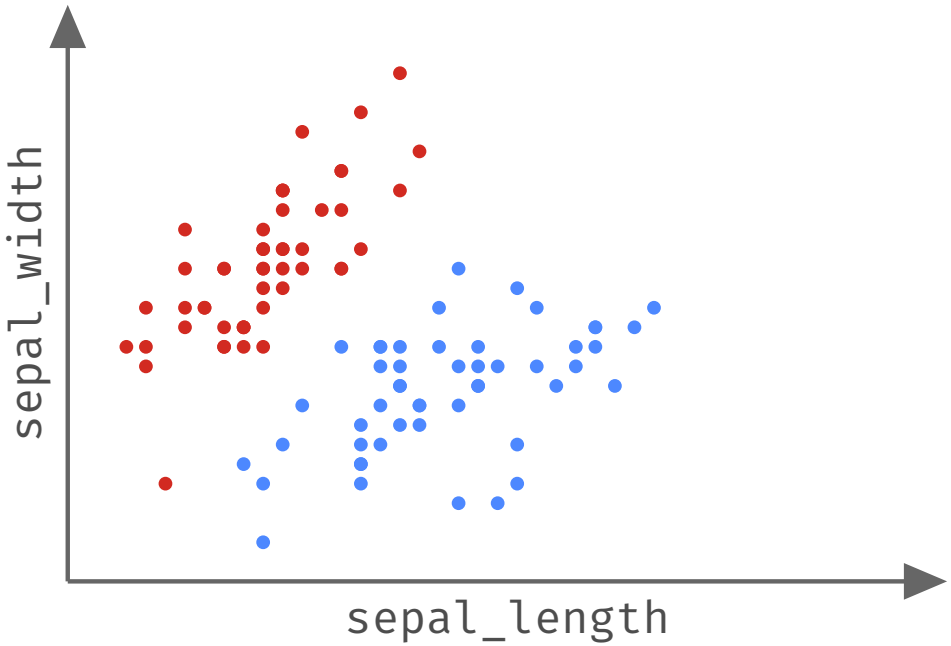
- Klassen: $\mathcal{Y} = \{\text{setosa}, \text{versicolor}, \text{virginica}\}$
- Menge $\mathbf{X} \times \mathbf{y}$ mit 150 Beispiele mit Spalte "`species`"
- Qualitätsfunktion

$$q(\mathbf{X}\times\mathbf{y}, f) = \sum_{(x,y)\in\mathbf{X}\times\mathbf{y}} err(y, \underbrace{f(x)}_{=\hat{y}}), \quad err(y,\hat{y}) = \begin{cases} 0, & \text{falls } y = \hat{y} \\ 1, & \text{sonst.} \end{cases}$$

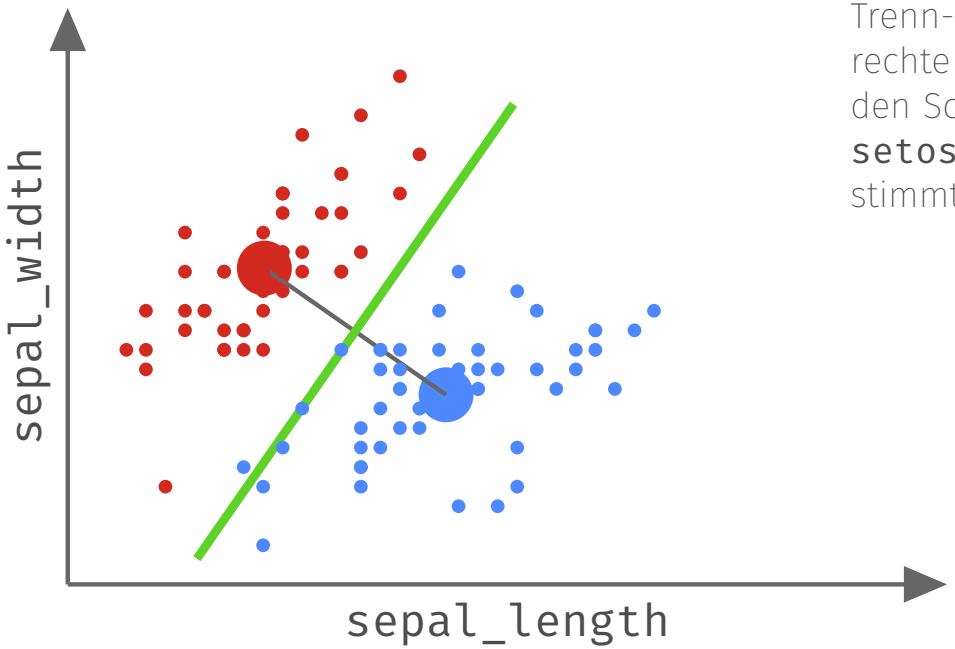
Funktion q zählt die Anzahl der Vorhersagefehler des Modells f auf der Menge X

Ziel: Finde f^* mit minimalem $q(X, f)$ → Optimierungsproblem

Beispiel: Klassifikation von Schwertlilien



Beispiel: Klassifikation von Schwertlilien

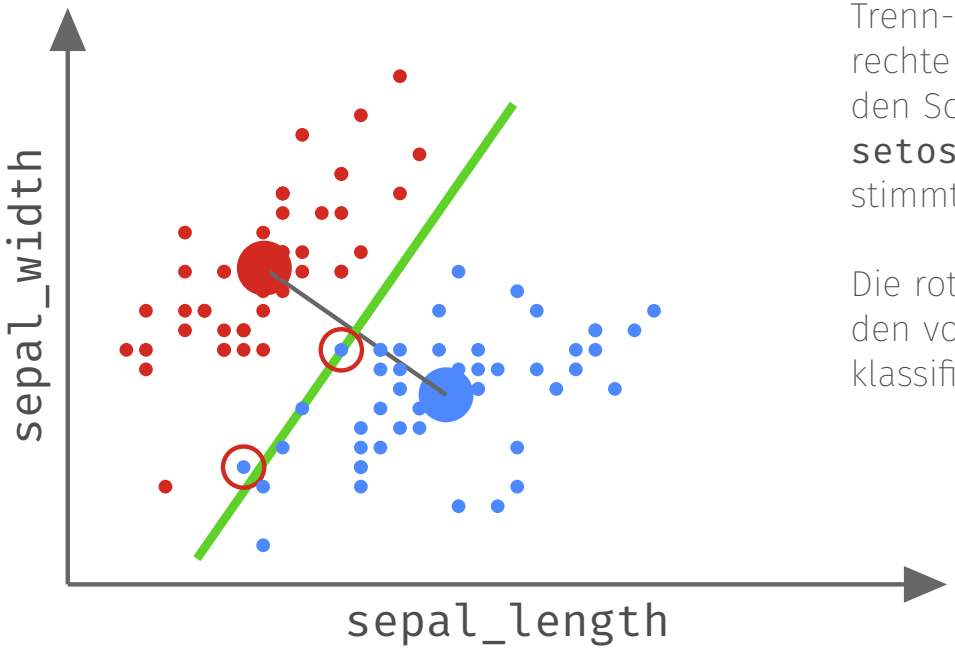


In diesem Fall wurde eine Trenn-Ebene als Mittelsenkrechte auf der Strecke zwischen den Schwerpunkten der Klasse `setosa` und `versicolor` bestimmt.

Einfacher Algorithmus:

Trenn-Ebene über die Klassenschwerpunkte der Attribute `sepal_length` und `sepal_width`

Beispiel: Klassifikation von Schwertlilien



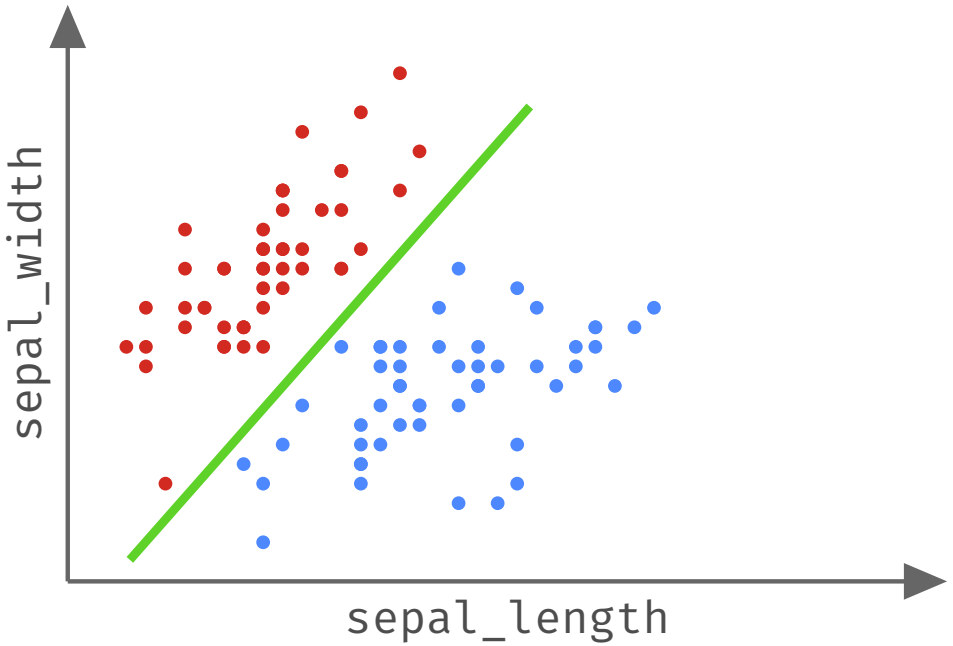
In diesem Fall wurde eine Trenn-Ebene als Mittelsenkrechte auf der Strecke zwischen den Schwerpunkten der Klasse `setosa` und `versicolor` bestimmt.

Die rot umkreisten Punkte werden von der Trenn-Ebene falsch klassifiziert.

Einfacher Algorithmus:

Trenn-Ebene über die Klassenschwerpunkte der Attribute `sepal_length` und `sepal_width`

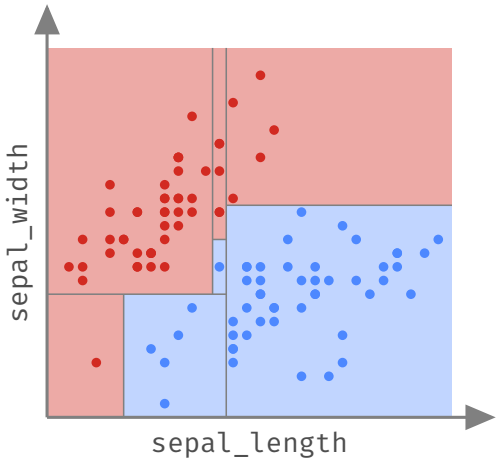
Beispiel: Klassifikation von Schwertlilien



Die Daten sind *linear separierbar* – eine andere Ebene schafft dies ohne Fehler.
Die Optimierung der Qualitätsfunktion sucht nach der besten Ebene.

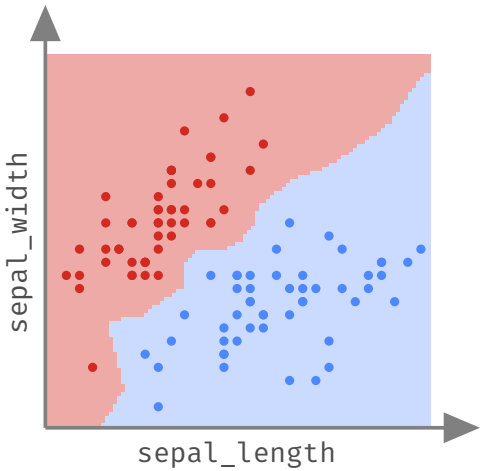
Klassifikationsverfahren

Entscheidungsbäume, nächste Nachbarn und lineare Modelle



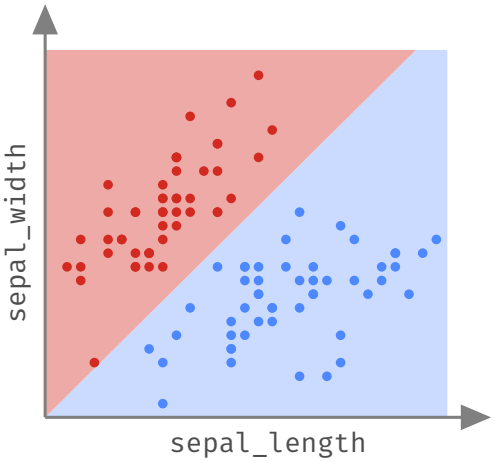
Entscheidungsbaum

Trennung nach einzelnen Attributen, achsenparallel



k-nächste Nachbarn

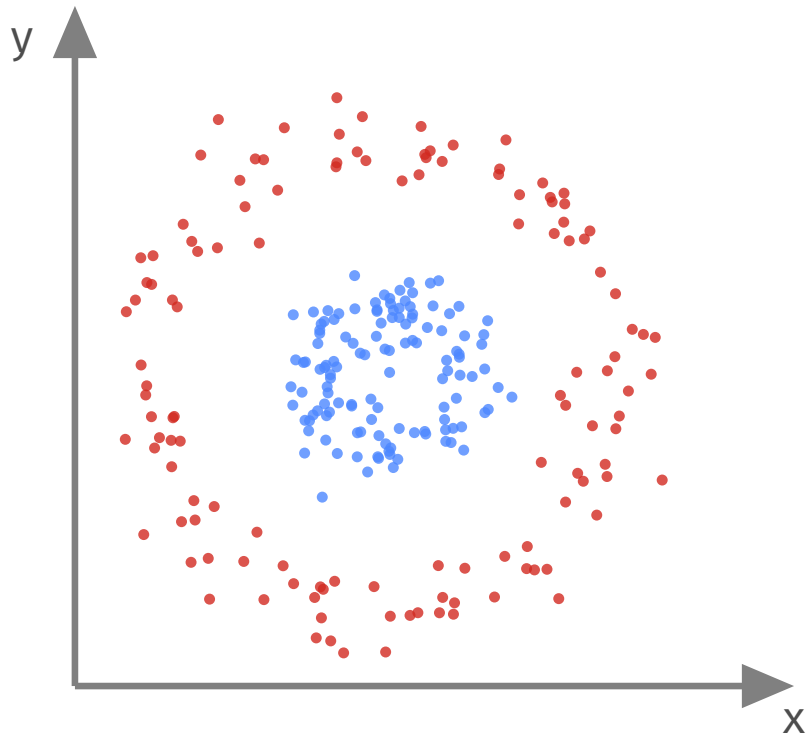
Trennung in Regionen, nach Distanz (Berechnung über alle Attribute)



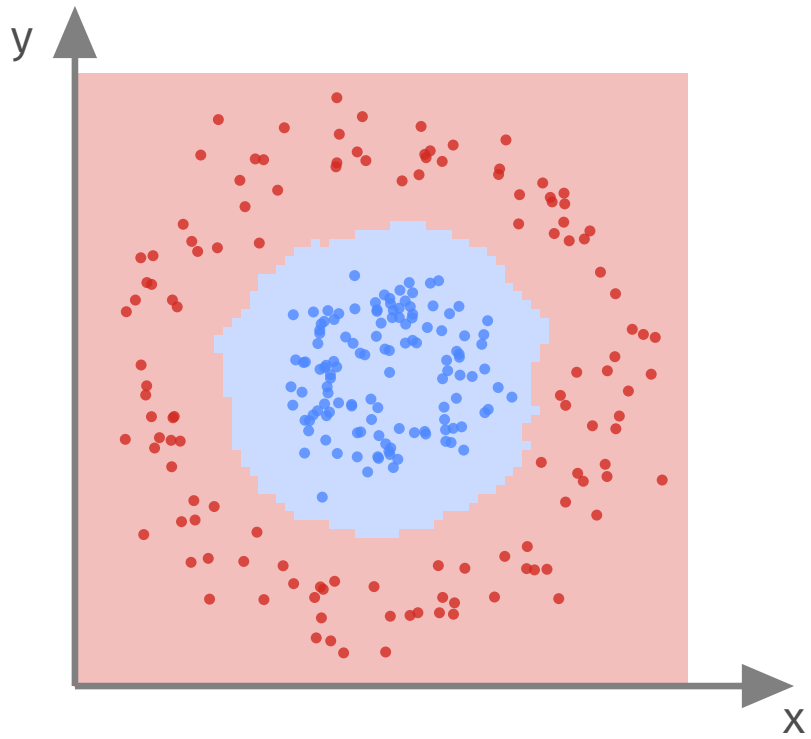
Lineare Modelle

Trennung mit linearer Funktion über alle Attribute

Betrachten wir einen anderen Datensatz:



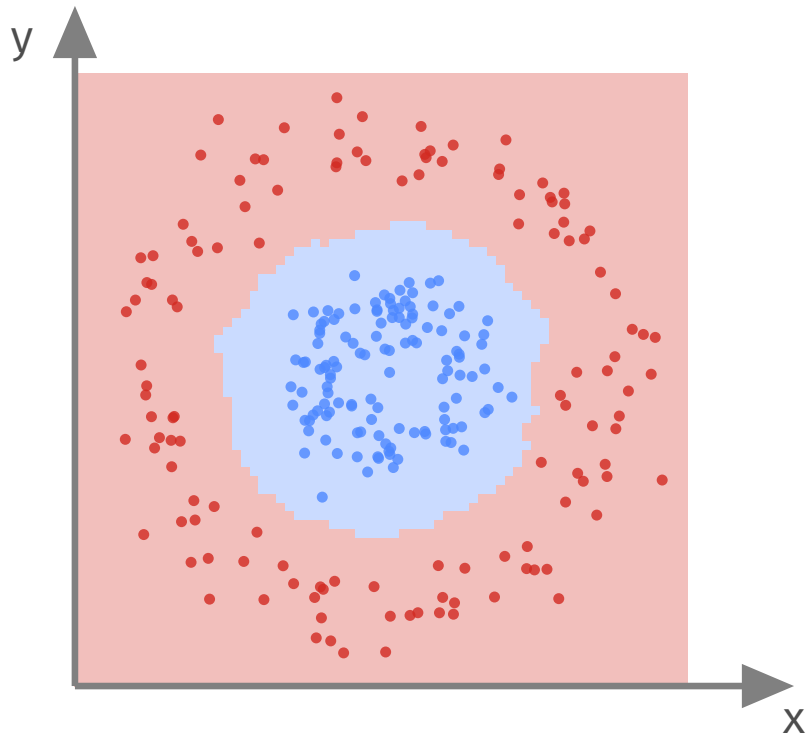
Betrachten wir einen anderen Datensatz:



Mit k-NN Modell kein Problem.

k-NN ist aber sehr langsam bei der Vorhersage! : (

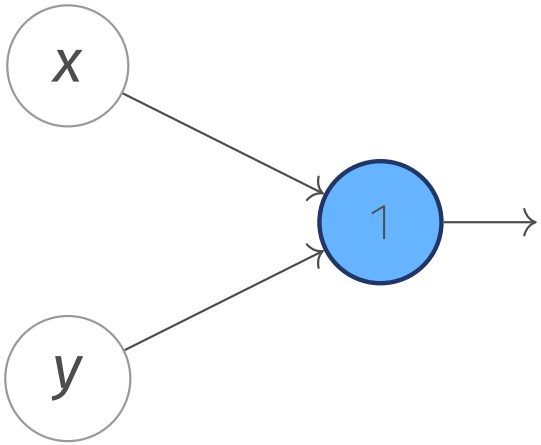
Betrachten wir einen anderen Datensatz:



Mit linearem Modell nicht lösbar?

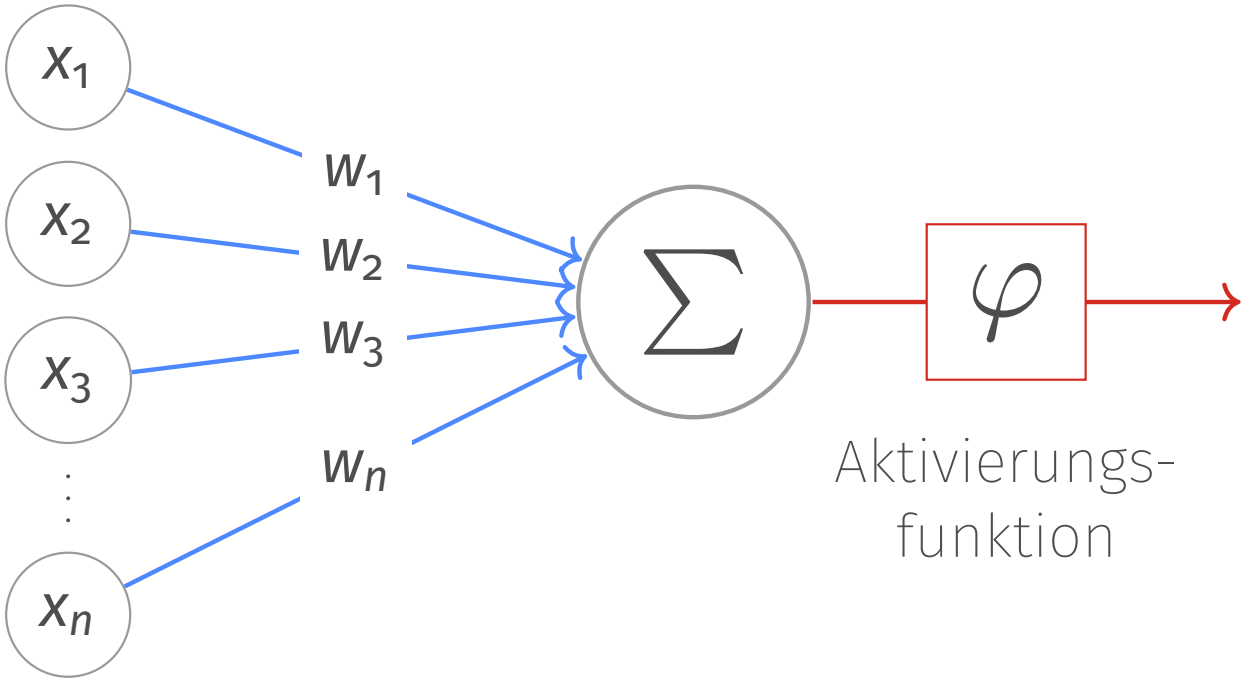
Neuronale Netze (und ChatGPT)

- Modellierung nach Neuronen (Gehirn)
- kleine Zellen die Input zu Output verarbeiten
- Basiert auf der Idee des Perzeptrons (Rosenblatt, 1957)



$$f(x,y) = \begin{cases} 1 & \text{, falls } x+y > 0 \\ 0 & \text{, sonst} \end{cases}$$

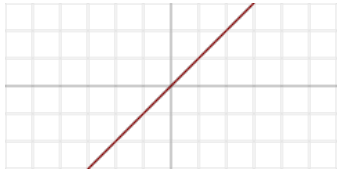
Neuron in einem Neuronalen Netz



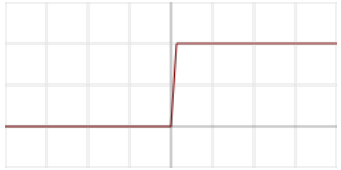
Aktivierungsfunktion

Berechnet Ausgabe aus gewichteter Eingabe-Summe.

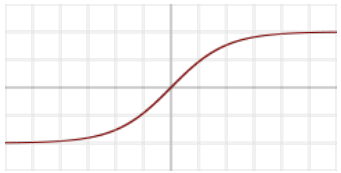
Identität $\mathrm{id}(x) = x$



Binäry Step $\varphi(x) = \begin{cases} 0, \text{ falls } x < 0 \\ 1, \text{ sonst} \end{cases}$



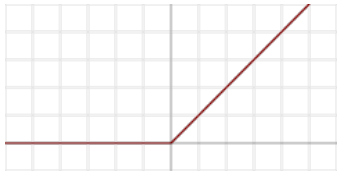
Tangens Hyperbolicus $\tanh(x) = \dfrac{e^{2x} - 1}{e^{2x} + 1}$



Weitere Aktivierungsfunktionen

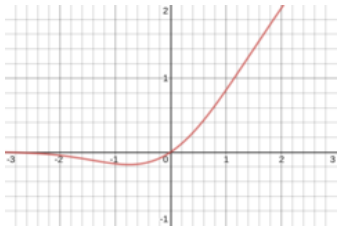
ReLU[1]

$$(x)^+ = \begin{cases} 0, \text{ falls } x \leq 0 \\ x, \; x > 0 \end{cases}$$



GELU[2]

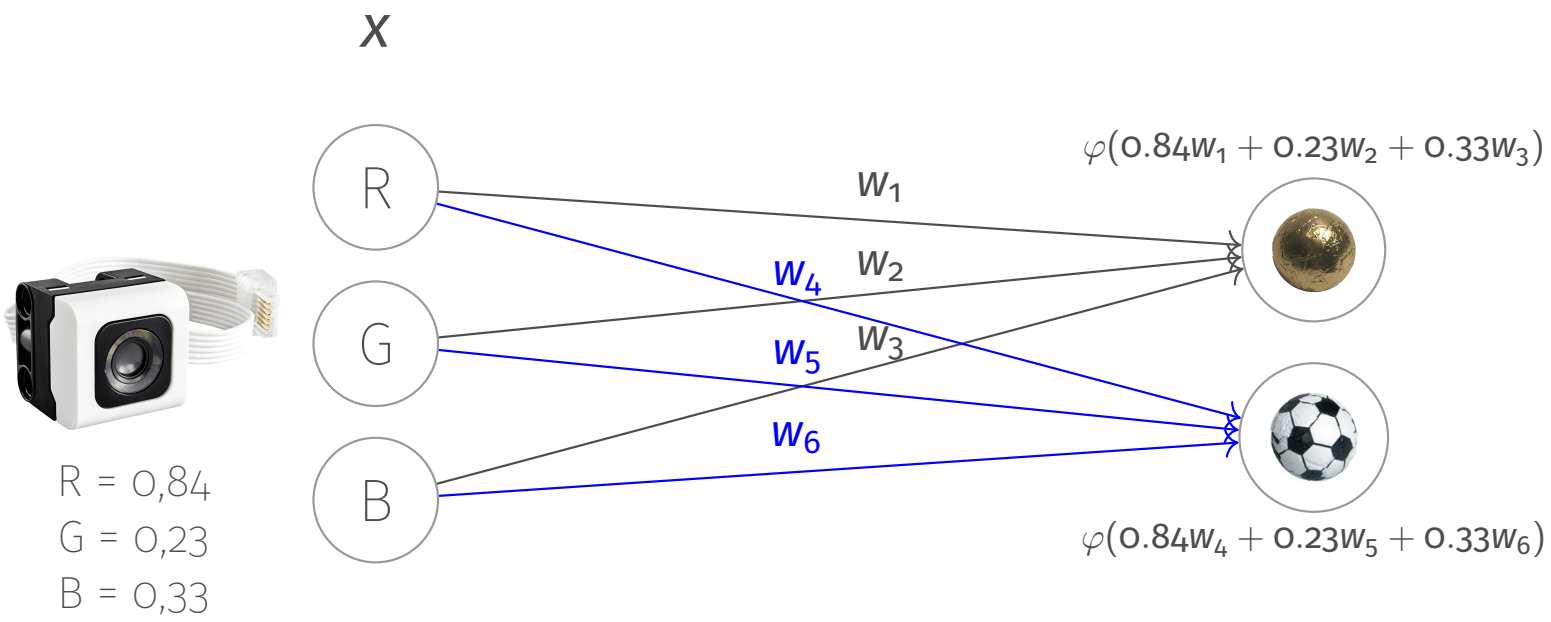
$$g(x) = x\tfrac{1}{2}\left[1 + erf(x/\sqrt{2})\right]$$



[1] *Rectified linear units improve restricted boltzmann machines*, Nair, Vinod and Hinton, Geoffrey E., Proceedings of the 27th International Conference on International Conference on Machine Learning, 2010

[2] *Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units*, Dan Hendrycks and Kevin Gimpel, Computing Research Repository (CoRR), 2016

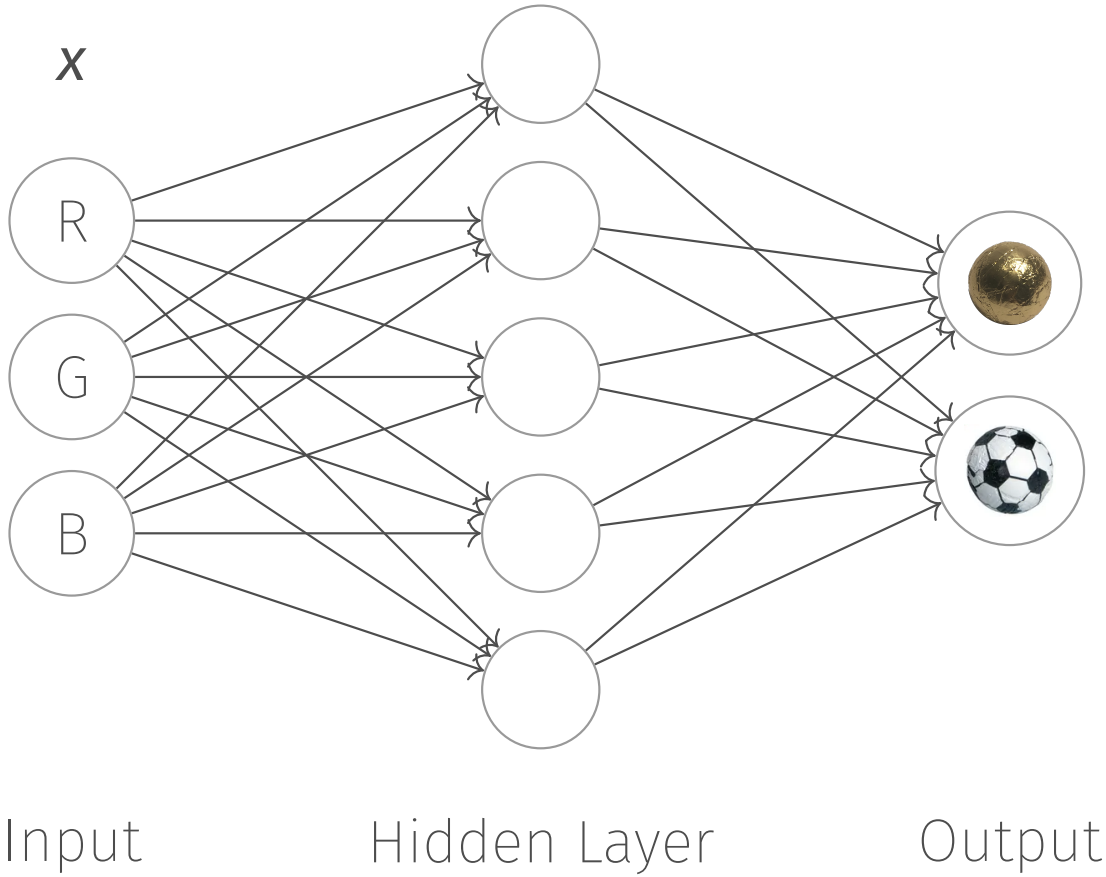
Funktionen mit neuronalen Netzen



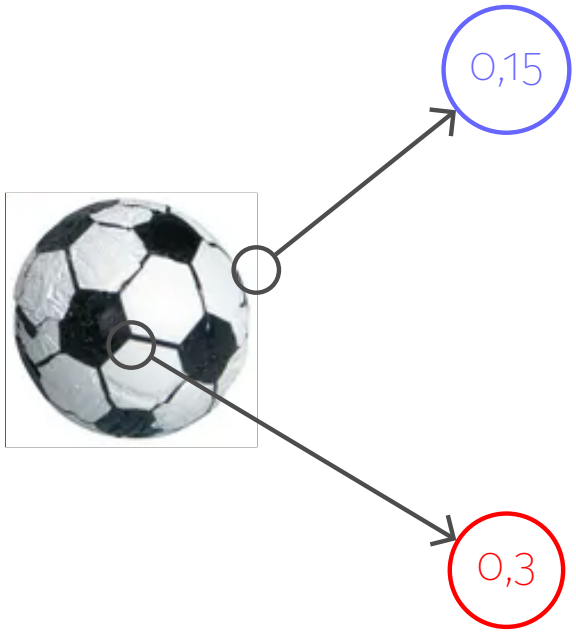
Ziel: Finde die Gewichte w_i, die zur genauesten Vorhersage führen!

Komplexe Funktionen mit neuronalen Netzen

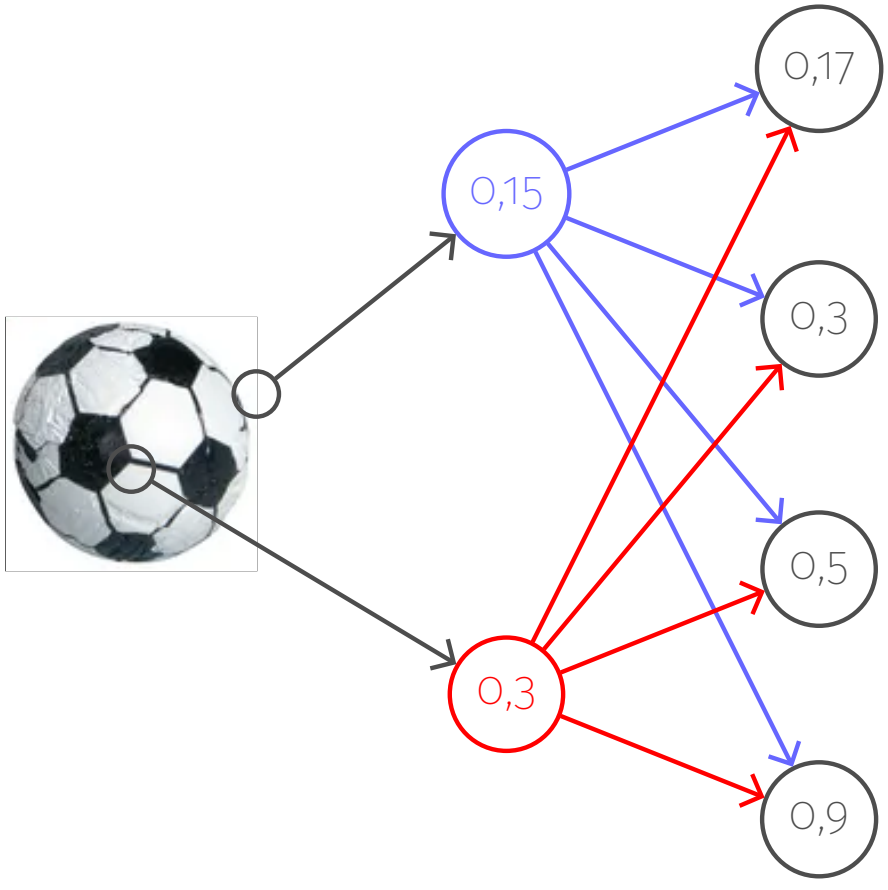
R = 84%
G = 23%
B = 33%



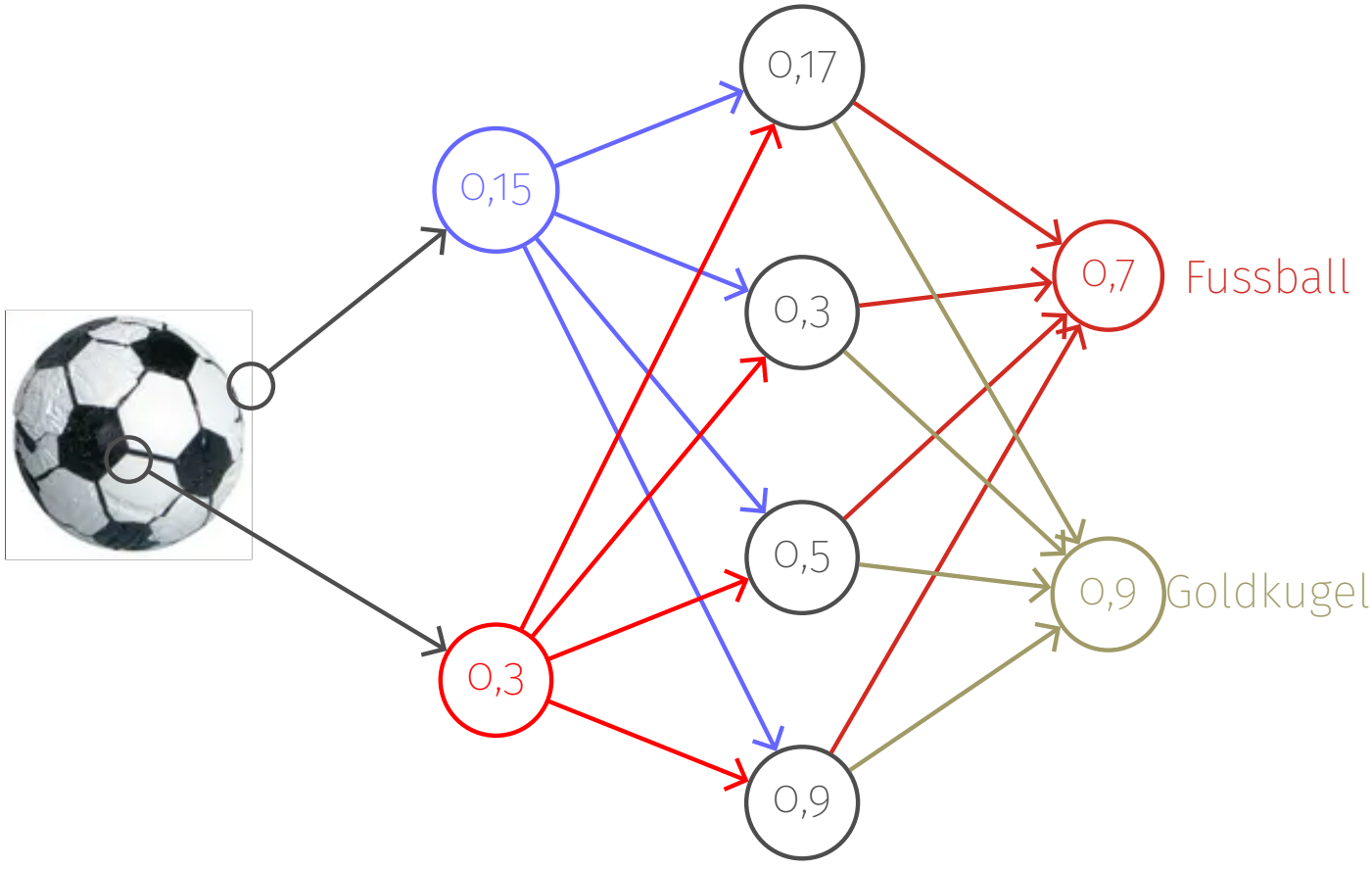
Einfache Bilderkennung mit NN

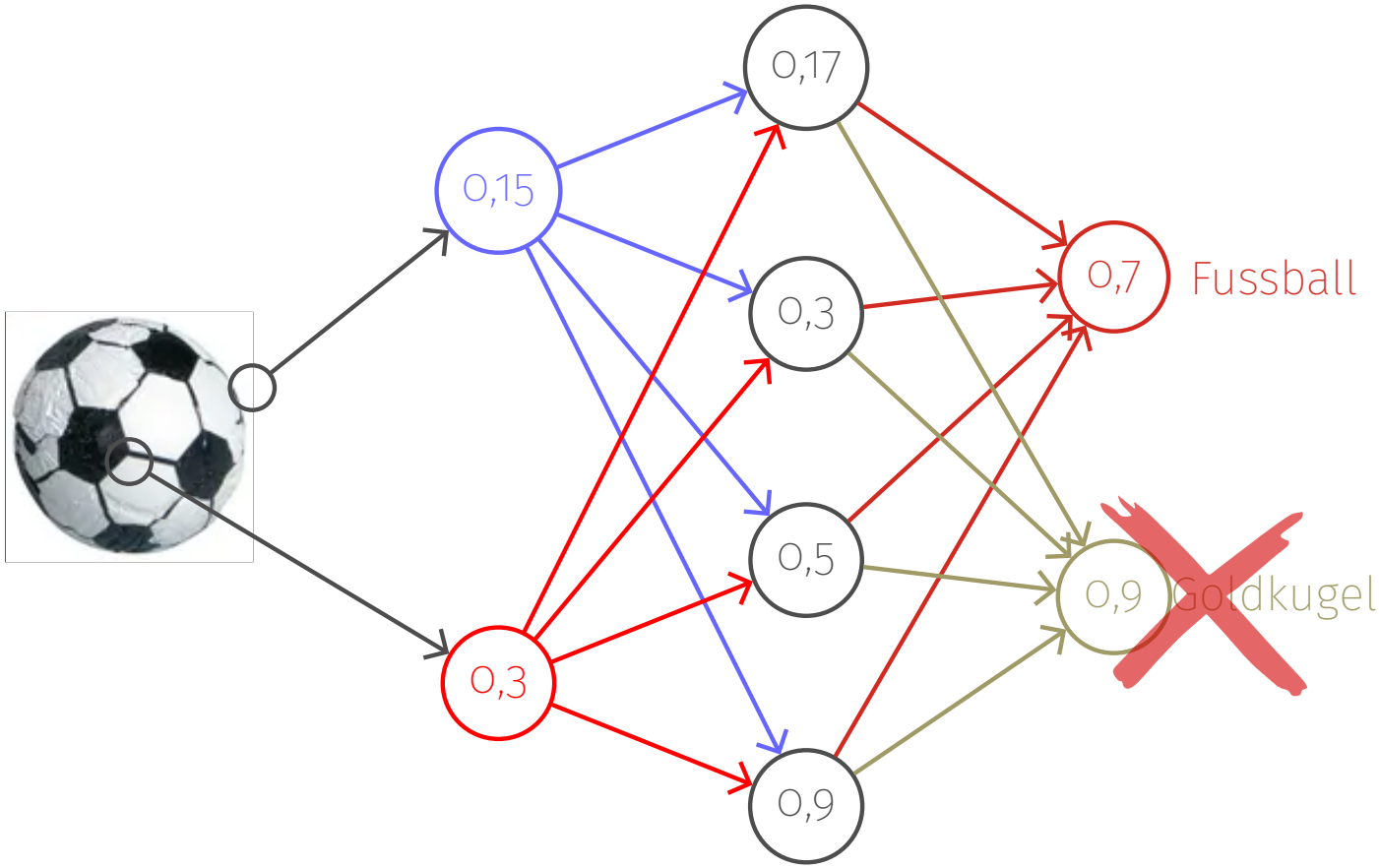


Einfache Bilderkennung mit NN



Einfache Bilderkennung mit NN




0,17
0,15
0,7
Fussball
0,3
0,5
0,9
Goldkugel
0,3
0,9

Korrektur der Gewichte

0,15

0,3

0,5

0,3

0,9

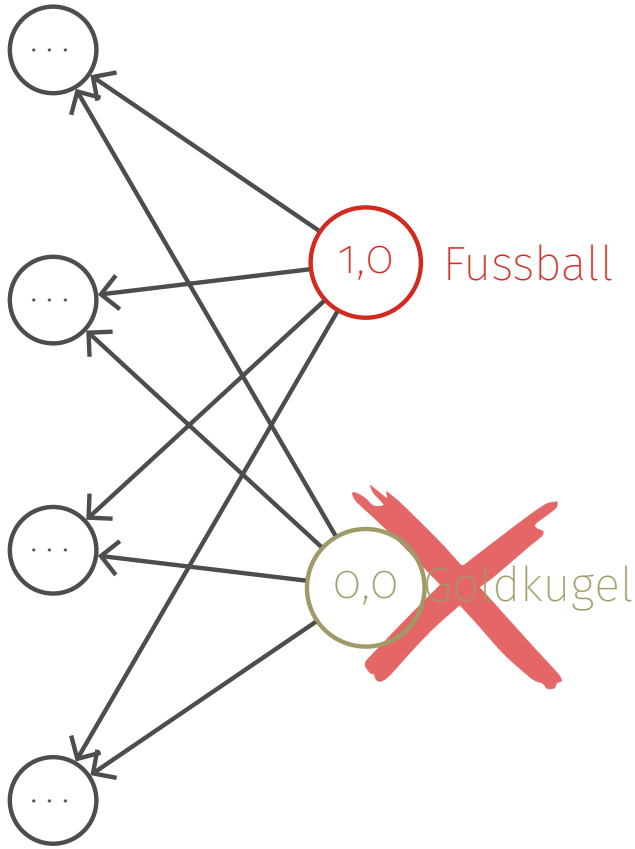
0,17

1,0 Fussball

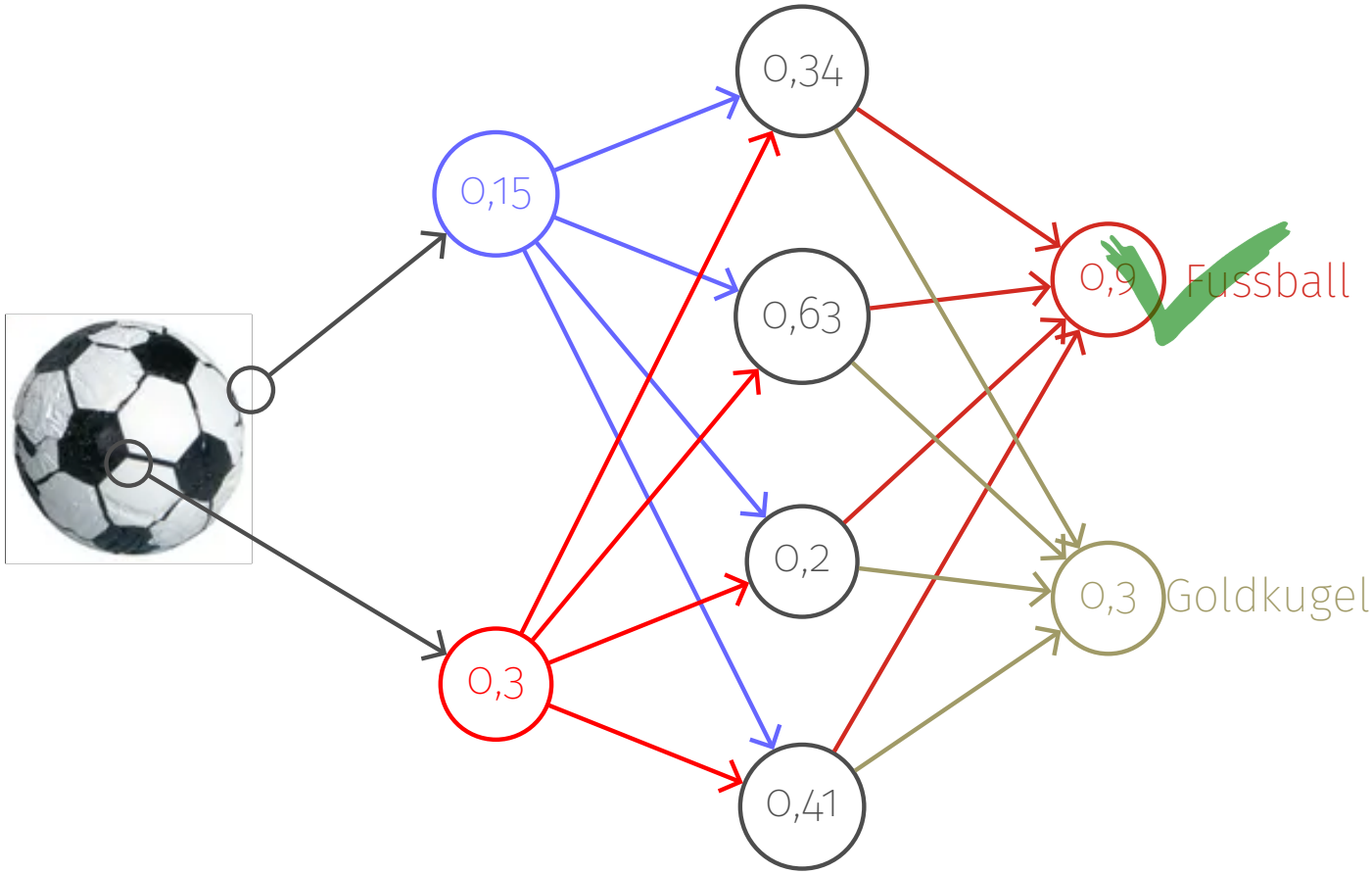
0,0 Goldkugel

0,15

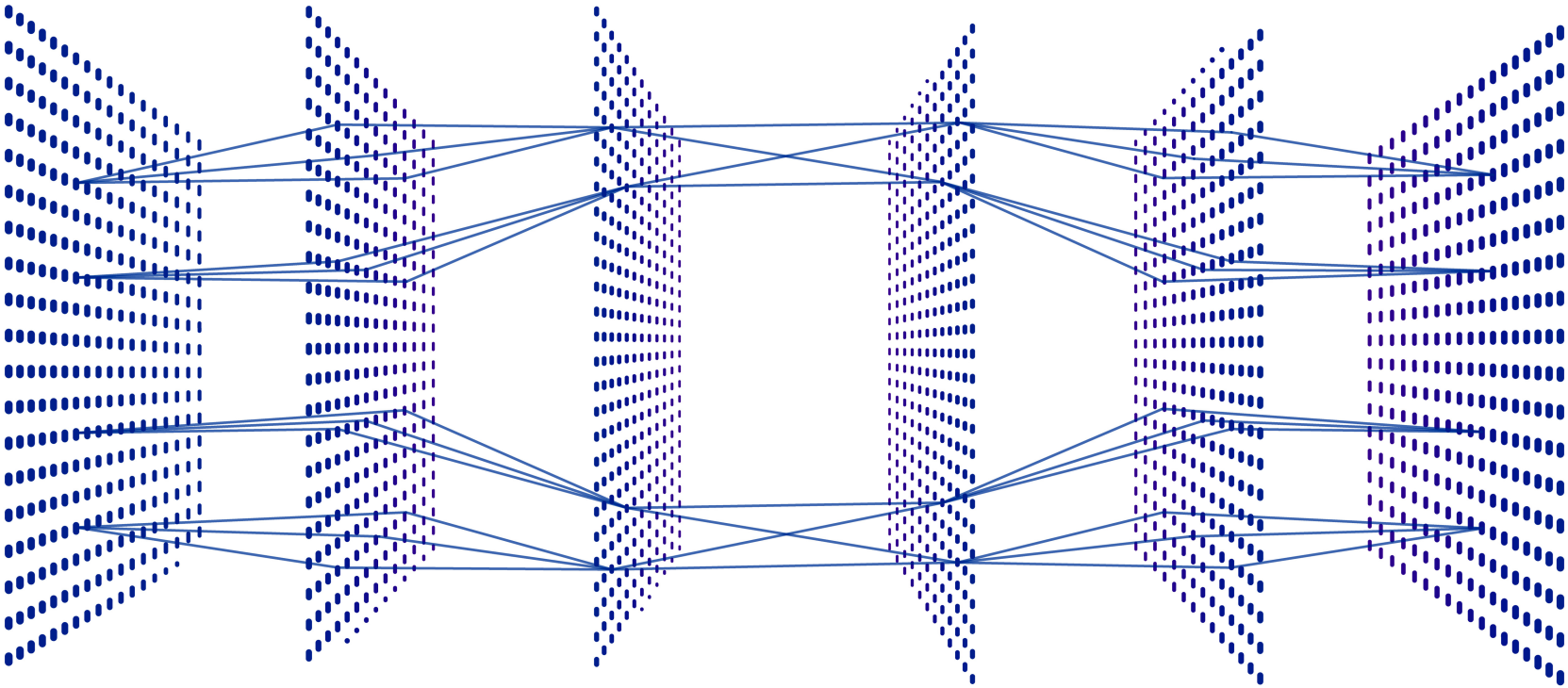
0,3



Neue Gewichte



Deep Learning



ChatGPT

- Generative AI, entwickelt von OpenAI
- Tiefes, Neuronales Netz (viele Layer/Ebenene)
- Verschiedene Ebenen, die unterschiedliche Aufgaben ausführen
- 175 Milliarden Parameter

ChatGPT

- Generative AI, entwickelt von OpenAI
- Tiefes, Neuronales Netz (viele Layer/Ebenene)
- Verschiedene Ebenen, die unterschiedliche Aufgaben ausführen
- 175 Milliarden Parameter
- Klassifikation mit Textbeispielen zum Lernen

Beispiel: "Das Konzert war großartig."
Klasse: "positive Stimmung"

ChatGPT

- Generative AI, entwickelt von OpenAI
- Tiefes, Neuronales Netz (viele Layer/Ebenene)
- Verschiedene Ebenen, die unterschiedliche Aufgaben ausführen
- 175 Milliarden Parameter
- Klassifikation mit Textbeispielen zum Lernen

  Beispiel: "Das Konzert war großartig."
  Klasse: "positive Stimmung"
- Bestärkendes Lernen aus Interaktion

  Benutzer: "Deine vorherige Antwort war hilfreich."
  Verstärkung: Positives Signal, um die Modellgewichtungen anzupassen.

Model Selection

Entscheidungsbäume

+ leicht zu interpretieren
+ schnell in der Vorhersage
- Nur jeweils ein Attribut zum verzweigen
- neigt zu Overfitting
- nur Achsen-parallele Entscheidungslinien

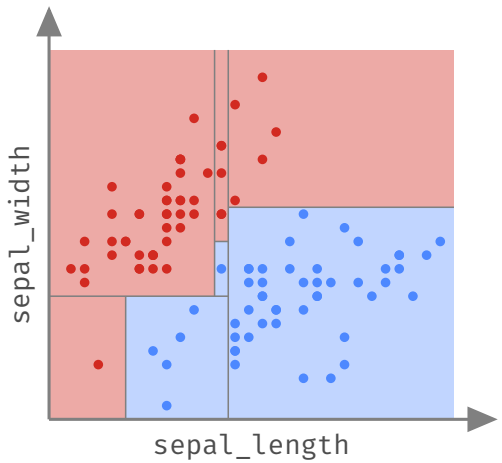
k-Nearest-Neighbors

+ leicht verständlich
+ robustes Verfahren, wenig Overfitting (für größeres ***k***)
- u.U. sehr langsam in der Vorhersage
- Distanzfunktion kann schwierig werden (Welche? Problemspezifisch?)
- Problematisch in hohen Dimensionen (spärliche Datenbereiche)

SVM (linear/nicht-linear)

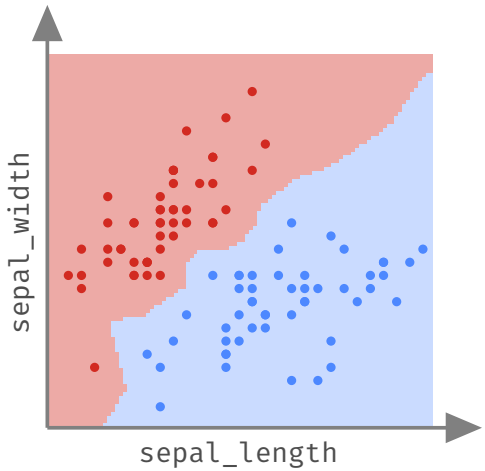
+ schnell in der Vorhersage
+ Funktioniert auch in hohen Dimensionen gut (Texte)
- Interpretierbarkeit?
- Auswahl der Kern-Funktion?
- Einstellungen für Parameter?

Entscheidungsbäume, nächste Nachbarn und lineare Modelle



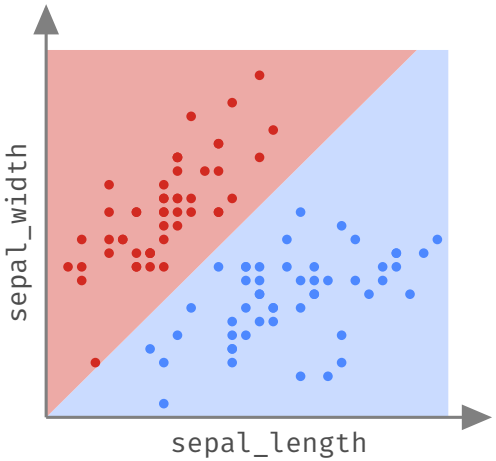
Entscheidungsbaum

Trennung nach einzelnen Attributen, achsenparallel



k-nächste Nachbarn

Trennung in Regionen, nach Distanz (Berechnung über alle Attribute)

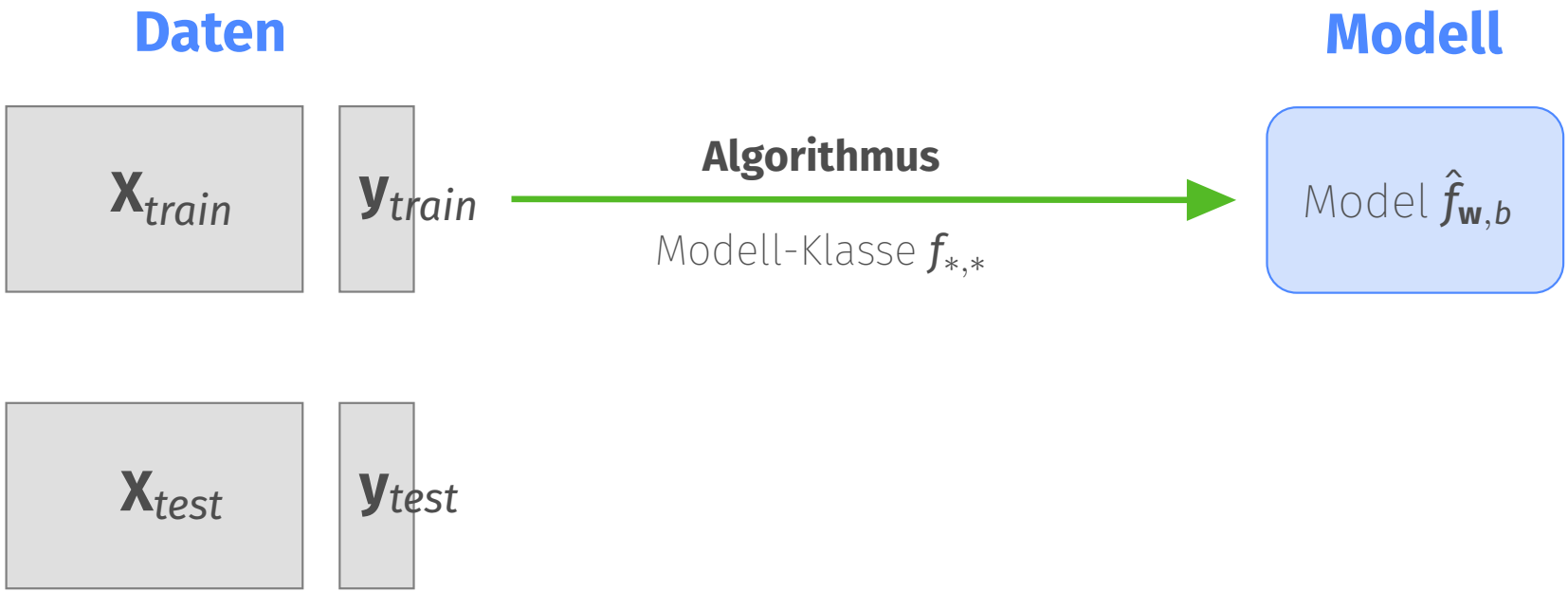


Lineare Modelle

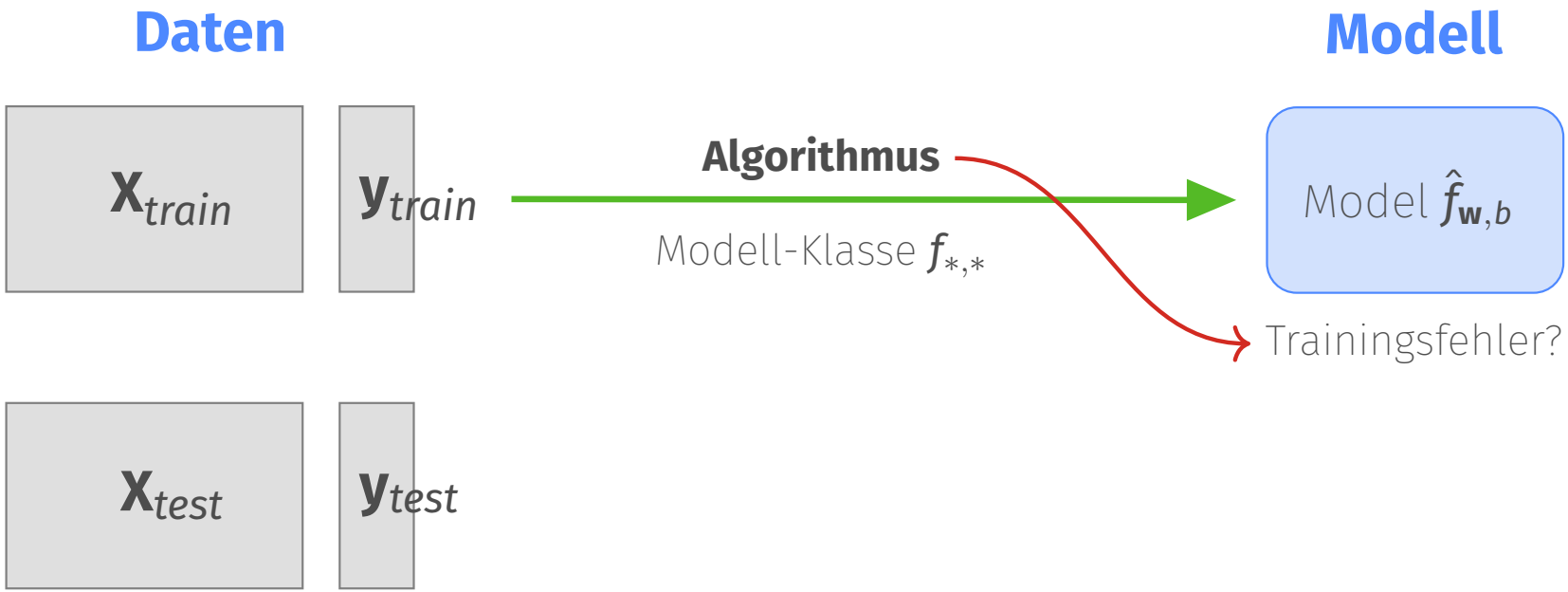
Trennung mit linearer Funktion über alle Attribute

Welches Verfahren für meine Daten?

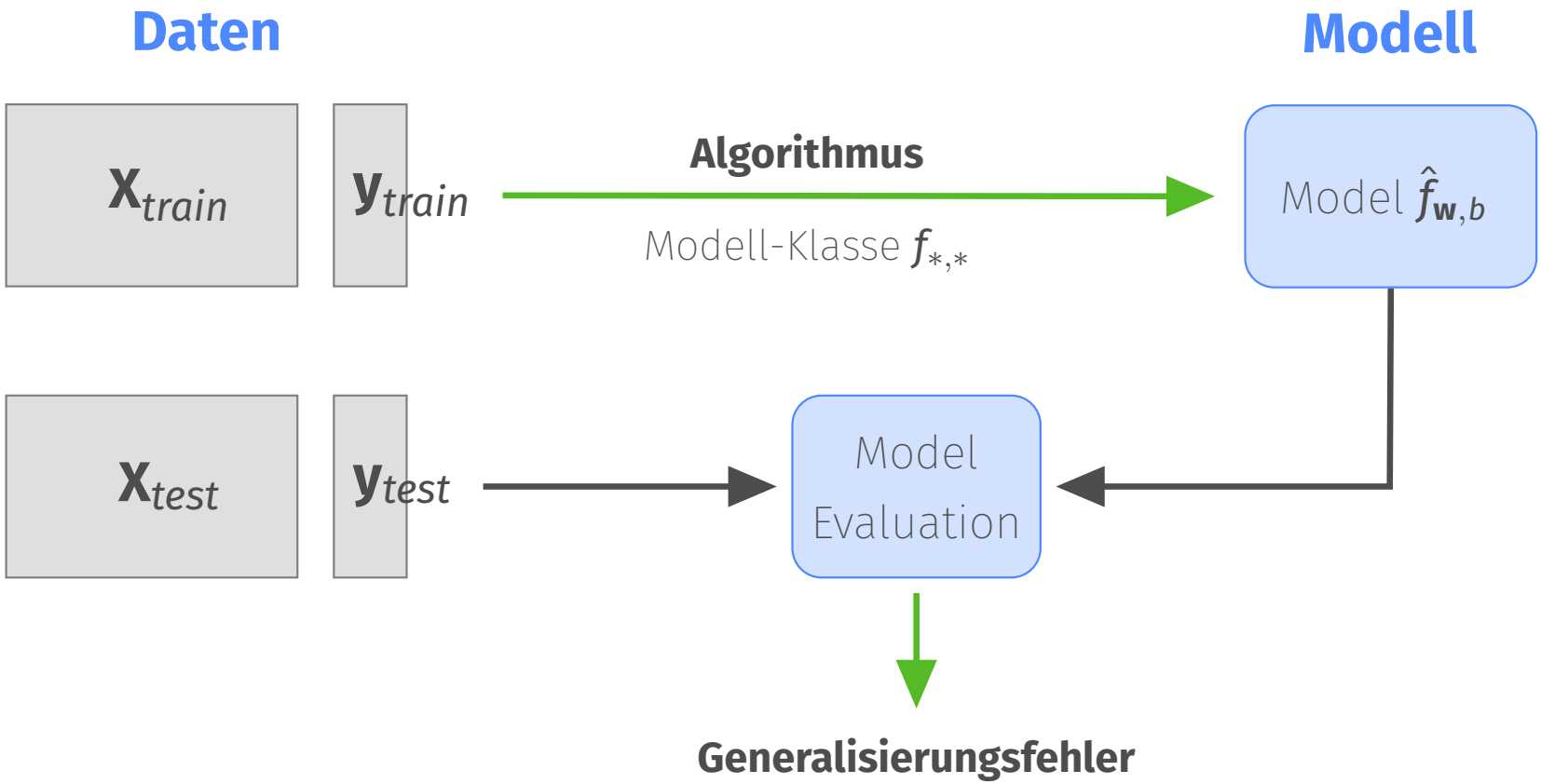
Vorgehen beim überwachten Lernen



Vorgehen beim überwachten Lernen



Vorgehen beim überwachten Lernen



Algorithmus-Auswahl:

- Wählt den Lern-Algorithmus, der das Modell auswählen soll
- Jeder Algorithmus ist auf Modell-Klasse beschränkt

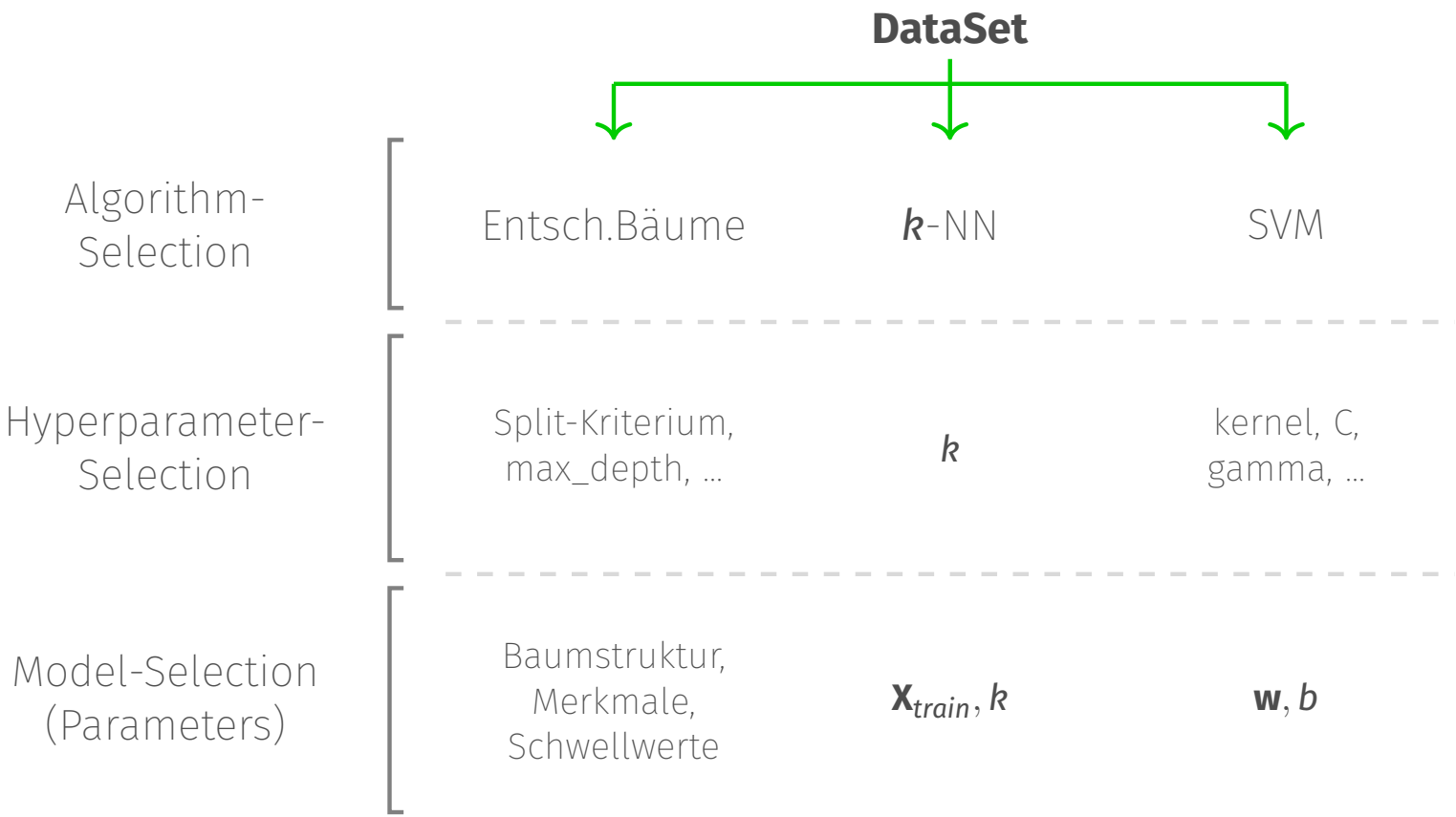
Hyperparameter-Auswahl:

- Wählt die Parameter für den Lern-Algorithmus
- Parameter bestimmen, wie das beste Modell gesucht wird

Parameter-Auswahl:

- Auswahl der Modell-Parameter durch den Algorithmus
- Modell-Parameter legen *ein* Modell aus Modell-Klasse fest

Suche nach dem besten Modell



Organisatorisches / Wie geht's weiter?

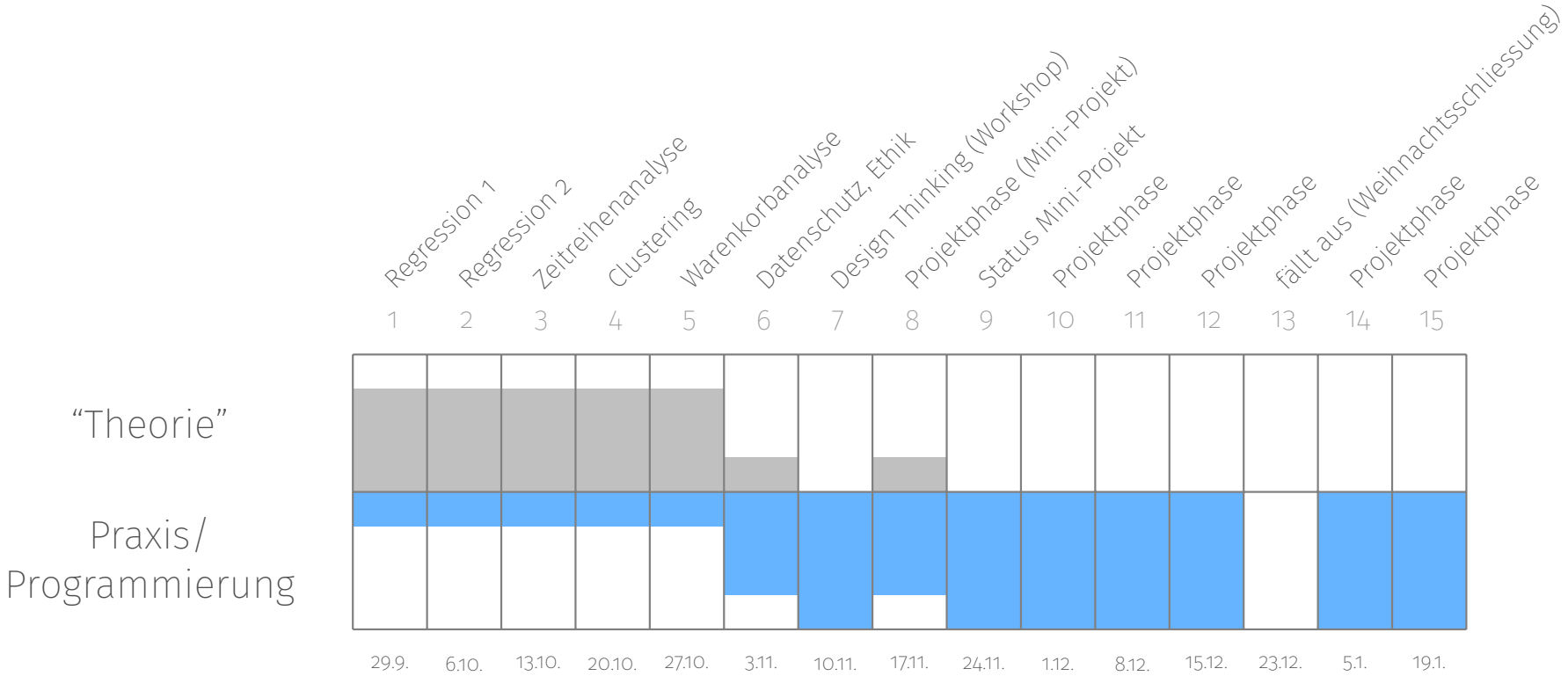
Prüfungsleistung - Hausarbeit

- Prüfungsleistung ist Hausarbeit
- Aufgabenstellung wird am 7.1.2025 vorgestellt
- Bearbeitung bis 9.2.2025 um 23:59 Uhr
  (Abgabe: PDF des Notebooks in Moodle hochladen)

- Gruppenarbeit mit max. 3 Personen möglich
- Selbstorganisation der Gruppen, bitte bis 14.1.2025 Gruppeneinteilung verbindlich mitteilen (Mail)

Was erwartet Sie in Data Science 2?

Aufbau der Vorlesung



Ziel des Kurses

- Datengetriebenes Denken fördern
- Lern-Probleme in Anwendungen identifizieren
- Ideen für Data Science Lösungen entwickeln
- Exploration+Prototyping von Daten/Modellen

Projektphase

- Zusammenhängenden Anwendungsfall bearbeiten
- Unterschiedliche Aufgaben in gleichem Fallbeispiel
- Gruppenarbeit, gemischte Studiengänge (!?)