Wirtschaftsinformatik – Informationsmanagement und Data Science

HOCHSCHULE HEILBRONN

› Data Science Grundlagen
Klassifikation

Prof. Dr. Carsten Lanquillon | Fakultät Wirtschaft und Verkehr | Wirtschaftsinformatik

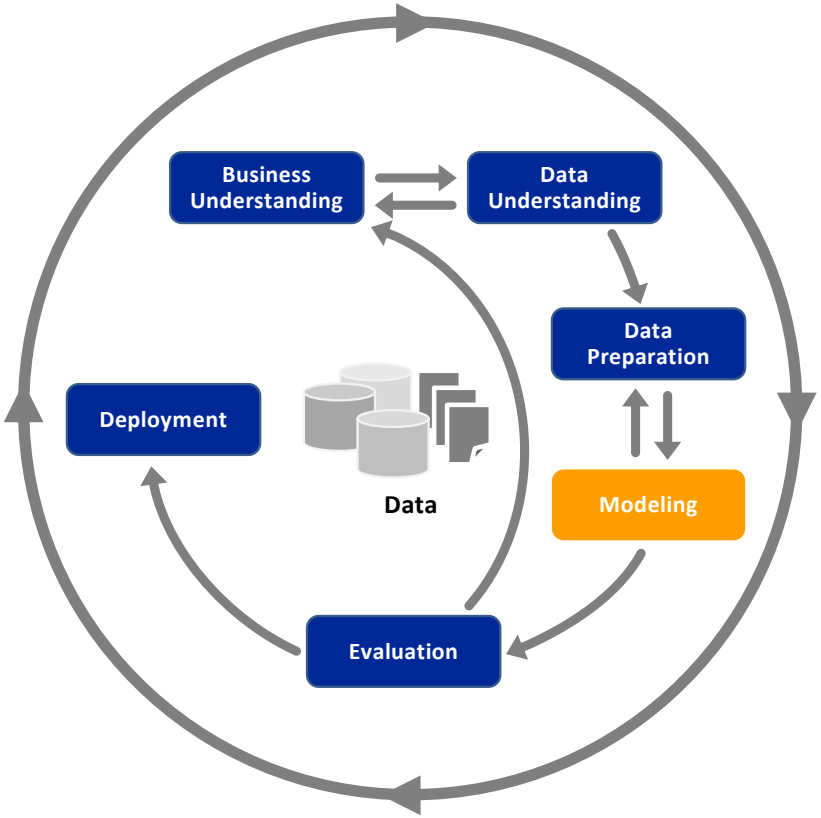
CRISP-DM Phase 4: Modeling

Ziel: Modellerstellung durch maschinelle Lernverfahren

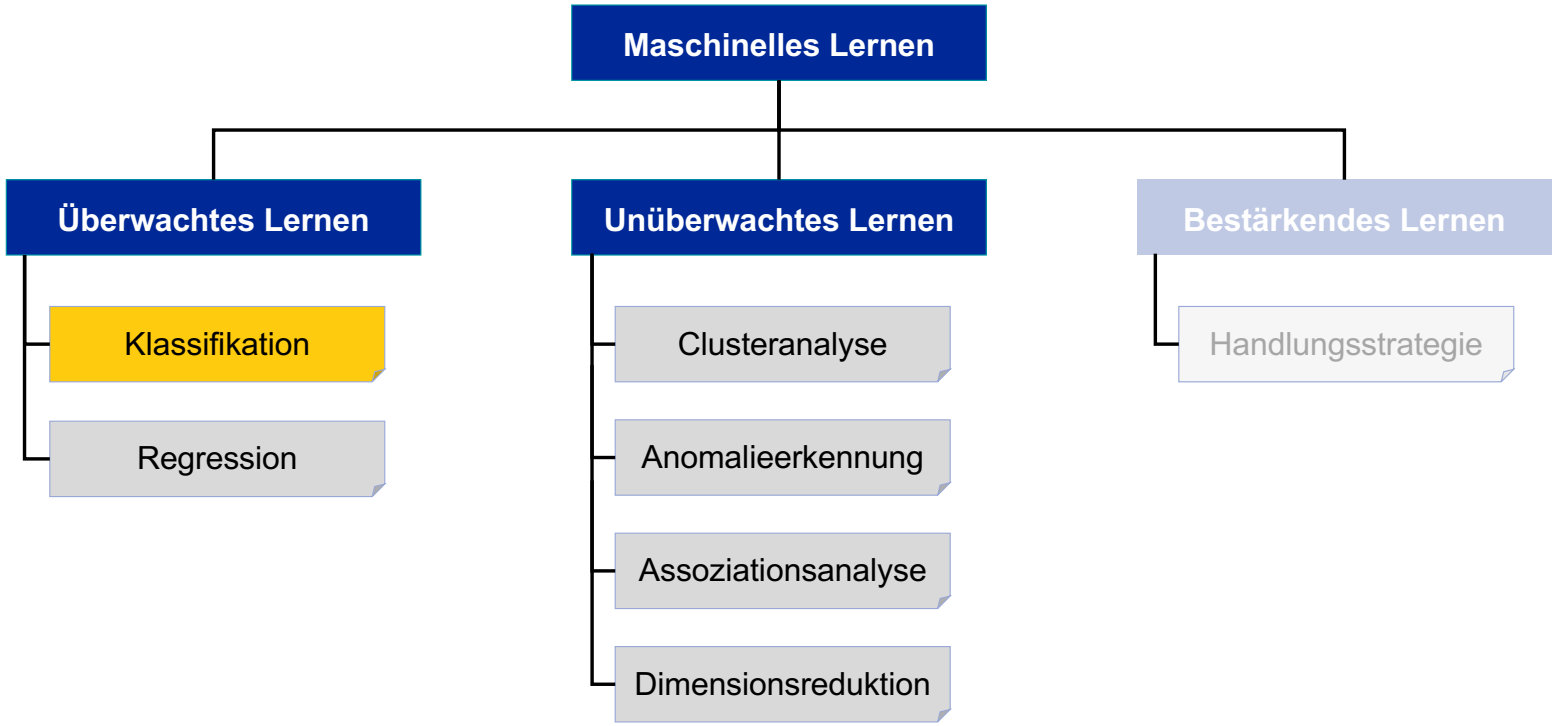
Aufgaben

- Auswahl geeigneter Lernverfahren (Model Selection)
- Nicht auf einen Lösungsansatz (Lernverfahren) festlegen („*no free lunch*“)
- Festlegen von Modellparametern
- Aufbau verschiedener Modelle (Lernen aus Daten)
- Auswahl eines Modells für die Aufgabenstellung

Fokus: Klassifikation

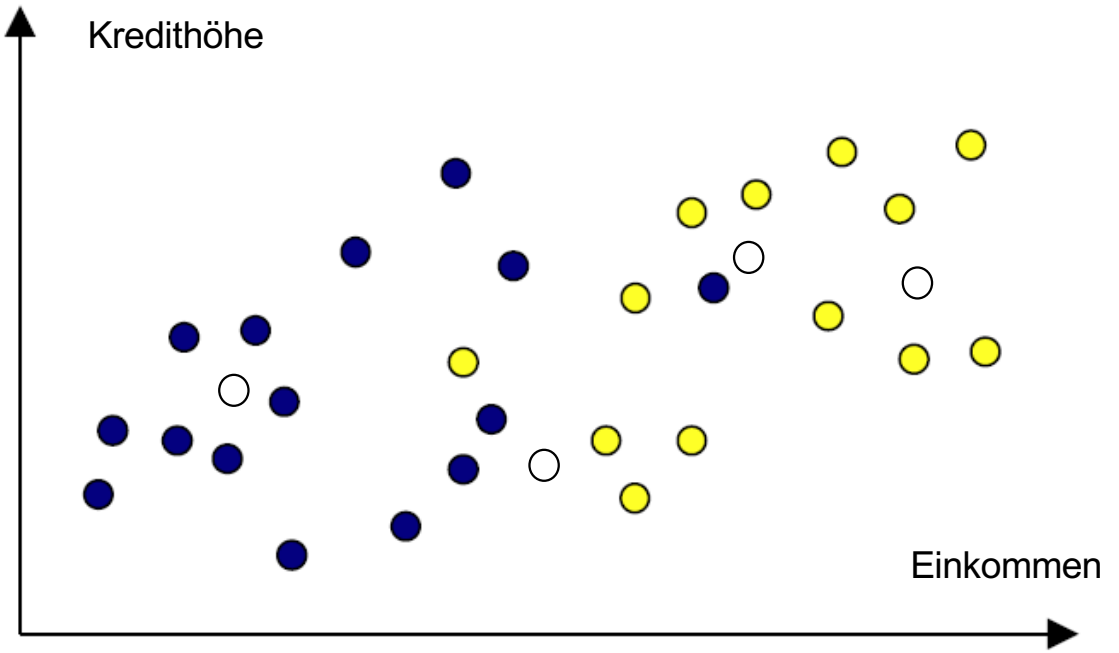


Lernformen und Data-Mining-Aufgaben



Klassifikation als Lernaufgabe

Ziel: Zuordnung von Objekten zu bekannten Klassen anhand von beobachtbaren Merkmalen

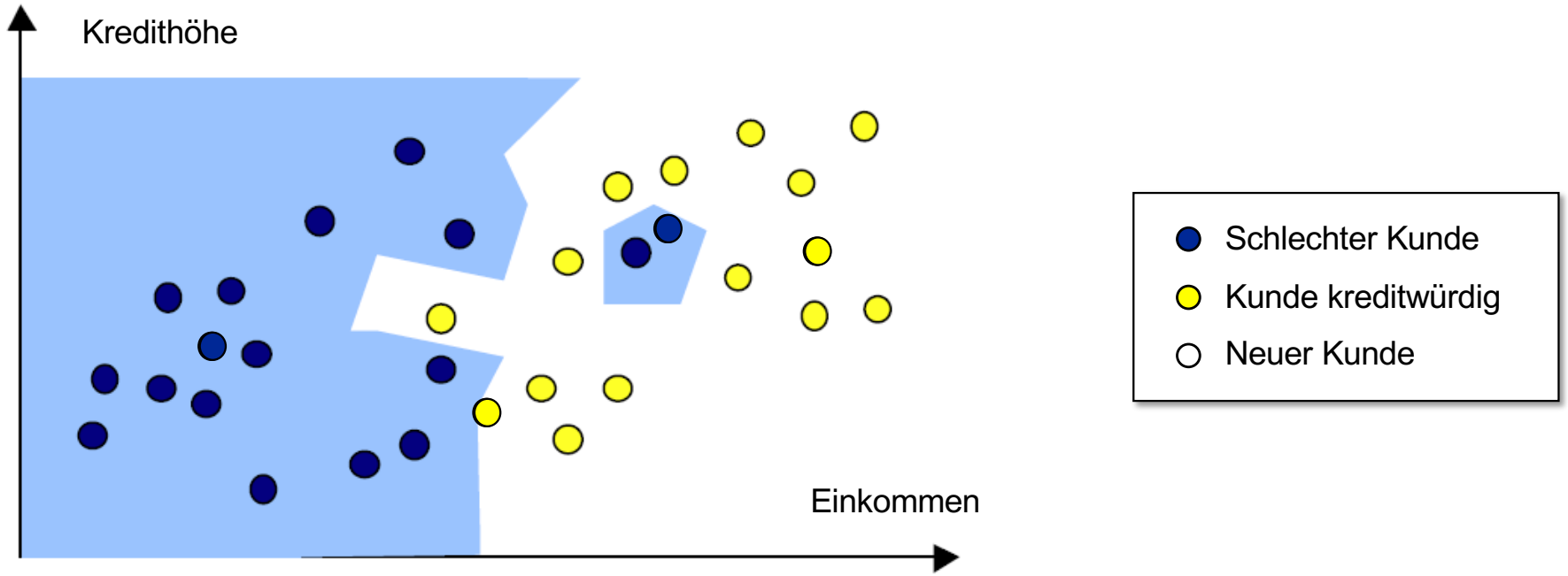


Wie würden Sie entscheiden?

- Schlechter Kunde
- Kunde kreditwürdig
- Neuer Kunde

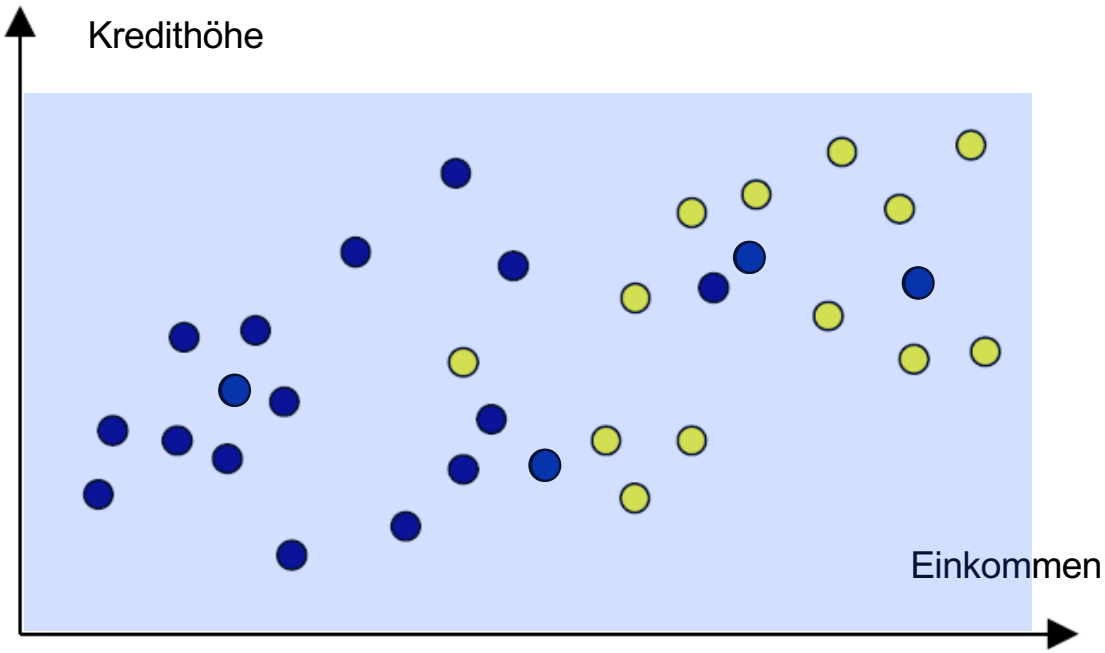
1-NN: Klassifikation mit Hilfe des nächsten Nachbarn

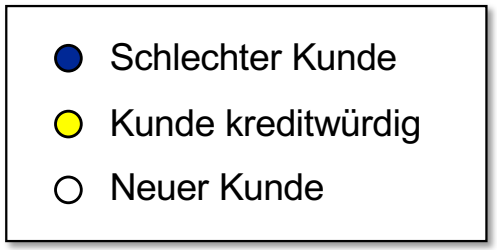
Einfache Lösung: Ordne neue Objekte der Klasse zu, die der nächste Nachbar hat!



Baseline: Default-Regel für die Klassifikation

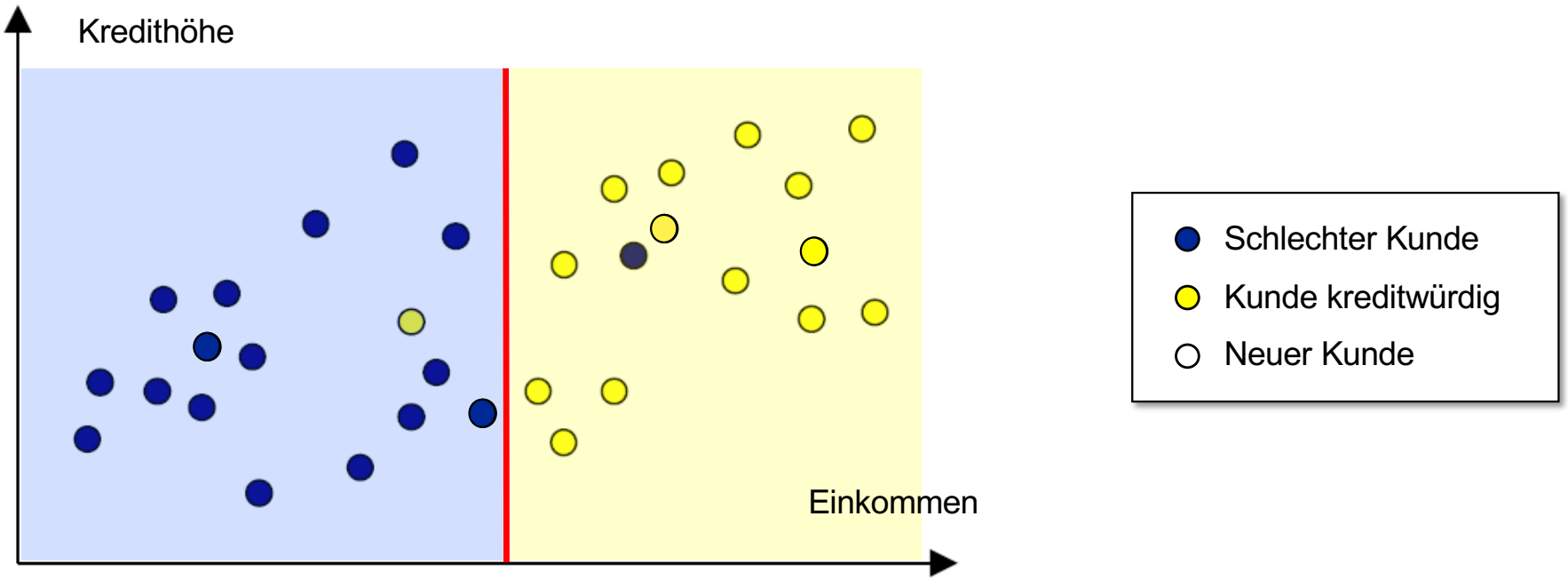
Wähle immer die häufigste Klasse aus den Trainingsdaten. → Hier: „**Schlechter Kunde**“





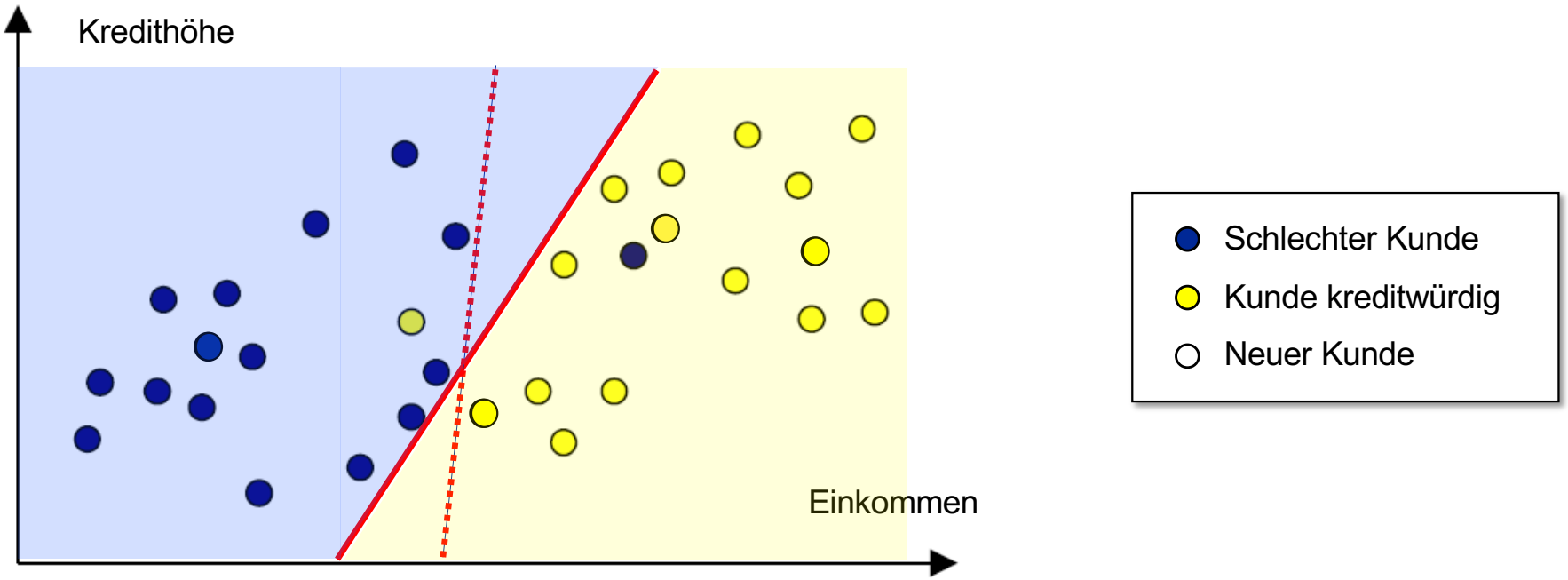
Baseline: 1R – One Rule

Entscheidung basierend auf den Ausprägungen eines einzigen Merkmals. → Hier: **Einkommen**



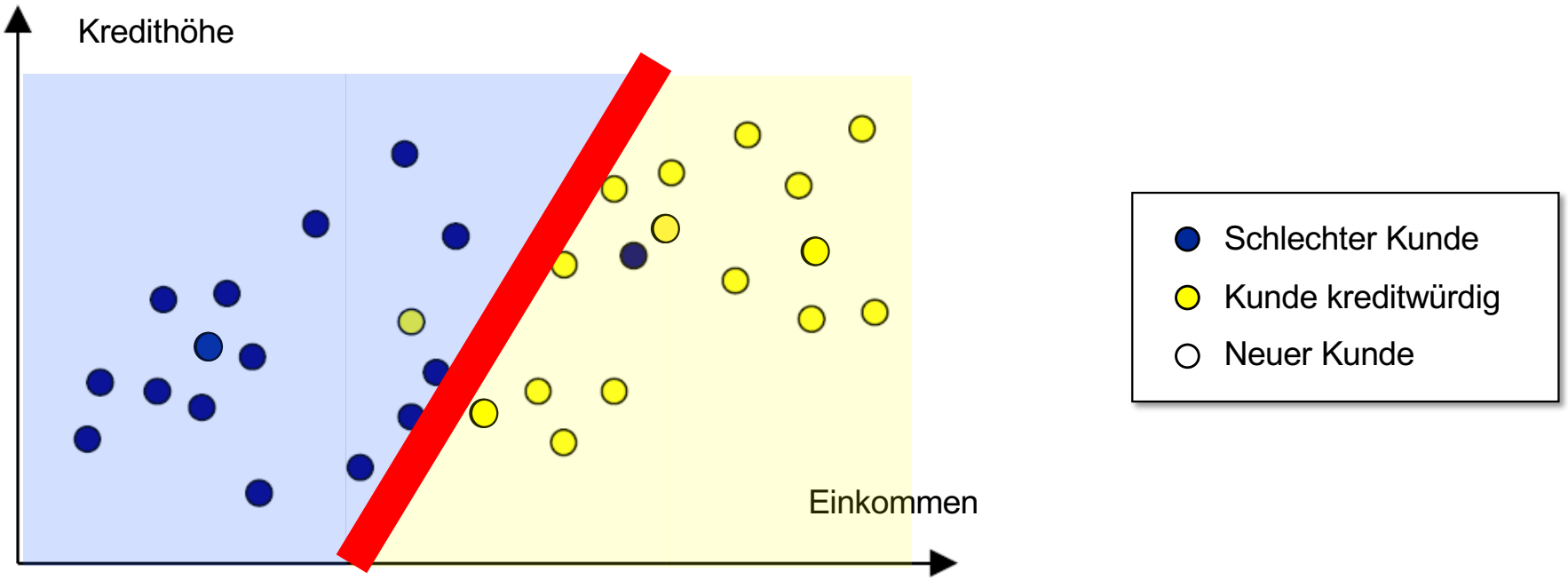
Lineare Klassifiktoren

Eine Gerade (allgemein Hyperebene) trennt die Objekte der beiden Klassen



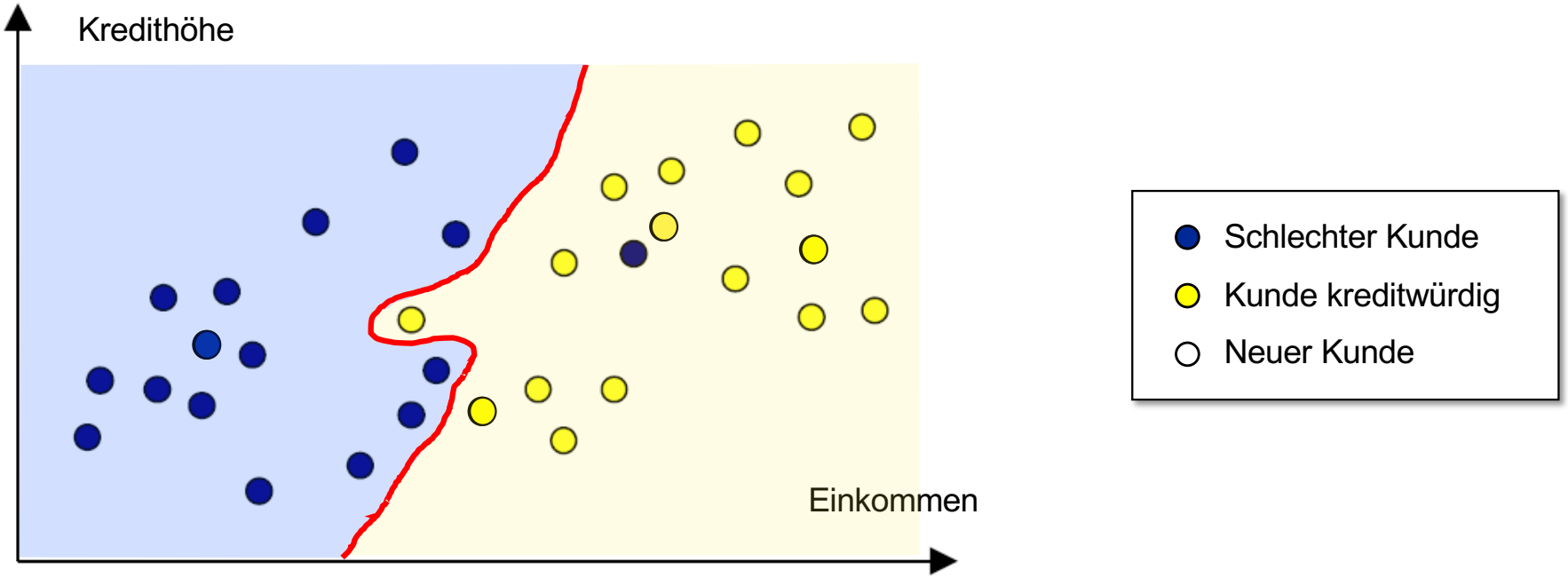
Linearer Klassifiktor mit „maximal margin“

Eine Gerade (allgemein Hyperebene) trennt die Objekte mit größtmöglichem Abstand



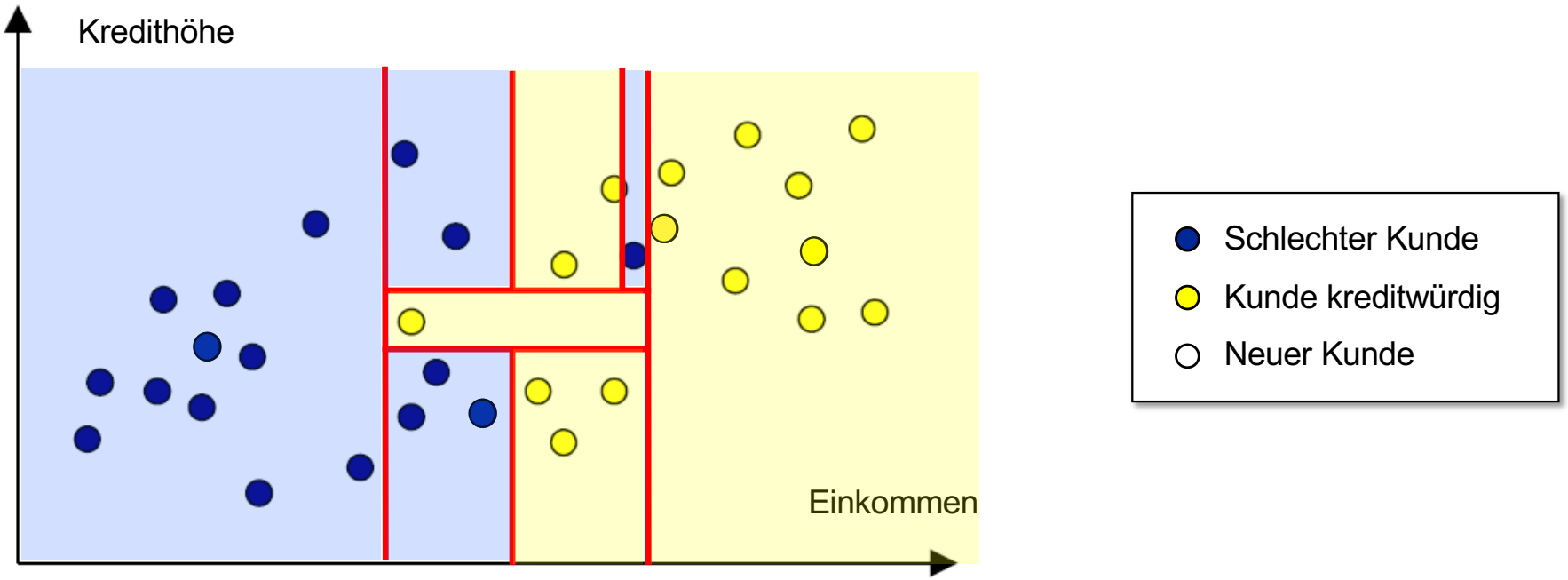
Nicht-Linearer Klassifikator

Neuronale Netze können beispielsweise beliebige Funktionen approximieren



Klassifikation mit einem Entscheidungbaum

Aufteilung des Eingaberaums anhand einzelner Attribute bis die Klassen getrennt sind

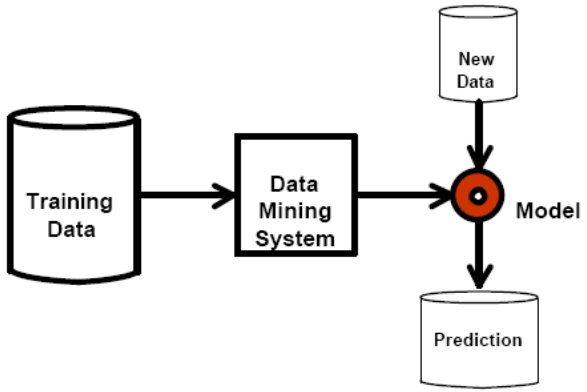


Überwachtes Lernen von Vorhersagemodellen

Basierend auf einer Trainingsmenge wollen wir

- die unbekannte Zielgröße Y für ungesehene Objekte basierend auf beobachteten Merkmalswerten X möglichst genau vorhersagen
- bei einem Klassifikationsproblem (auch) die Wahrscheinlichkeit (Konfidenz) zum Vorhersagewert ausgeben
- verstehen welche Eingaben die Ausgaben wie beeinflussen
- die Vorhersagequalität beurteilen

past-expenses	age	bonus	gender	accept
low	elder	high	female	no
low	elder	high	male	no
average	elder	high	female	yes
high	average	high	female	yes
high	young	normal	female	yes
high	young	normal	male	no
average	young	normal	male	yes
low	average	high	female	no
low	young	normal	female	yes



K-Nächste-Nachbar-Klassifikation

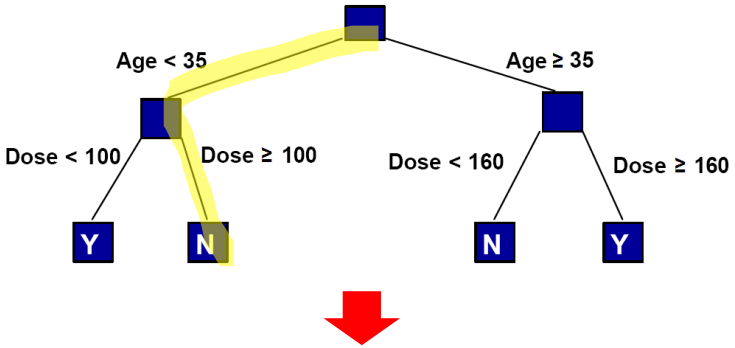
- Alle Trainingsbeispiele stellen das Modell dar
  → keine echte Lernphase (lazy learning)
  - Finde die k nächsten Objekte, die häufigste Klasse bekommt den Zuschlag
  - Um das Risko eines Gleichstands zu verringern, wähle k ungerade
  - Einfach umzusetzen aber zeitintensiv in der Anwendung (Nachbarschaftssuche)

- **Besondere Herausforderung:**
  → Angemessene Abstandsberechnung in hochdimensionalen Räumen

Entscheidungsbäume

Entscheidungsbaum = Folge geschachtelter Wenn-Dann-Regeln

- Innere Knoten: Verzweigung (Bedingung)
- Blätter: Entscheidung (Konsequenz)
- Pfade von der Wurzel zu den Blättern entsprechen Regeln



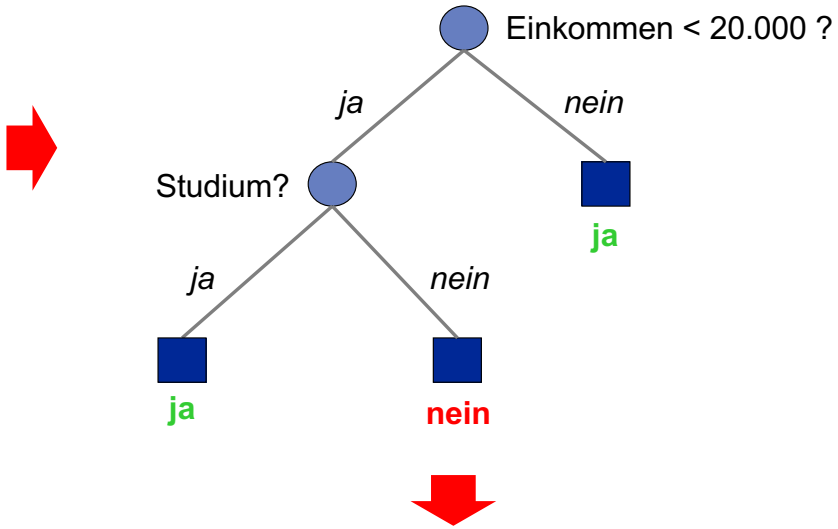

```
if Age <  35 and Dose <  100 then Y
if Age <  35 and Dose >= 100 then N
if Age >= 35 and Dose <  160 then N
If Age >= 35 and Dose >= 160 then Y
```

Vorteile

- Verständlichkeit (wenn der Baum nicht zu groß ist)
- Einfach in SQL-Abfrage zu überführen
- Baumerstellung vergleichsweiße schnell
- Flexible nicht-lineare Entscheidungsgrenzen sind abbildbar
  → **Gefahr: Overfitting**

Wie kommt man von Daten zum Baum?

Einkommen	Abschluss	Kredit
10.000	Keiner	nein
40.000	Abitur	ja
19.000	Abitur	nein
75.000	Studium	ja
18.000	Studium	ja




```
if Einkommen ≥ 20.000 then ja
if Einkommen < 20.000 and Studium then ja
if Einkommen < 20.000 and not Studium then nein
```

Noch ein Beispiel

Ziel: Aufbau eines Entscheidungsbaumes für die Vorhersage, ob wir an einem Tag Tennis spielen

Beschreibung der Merkmale

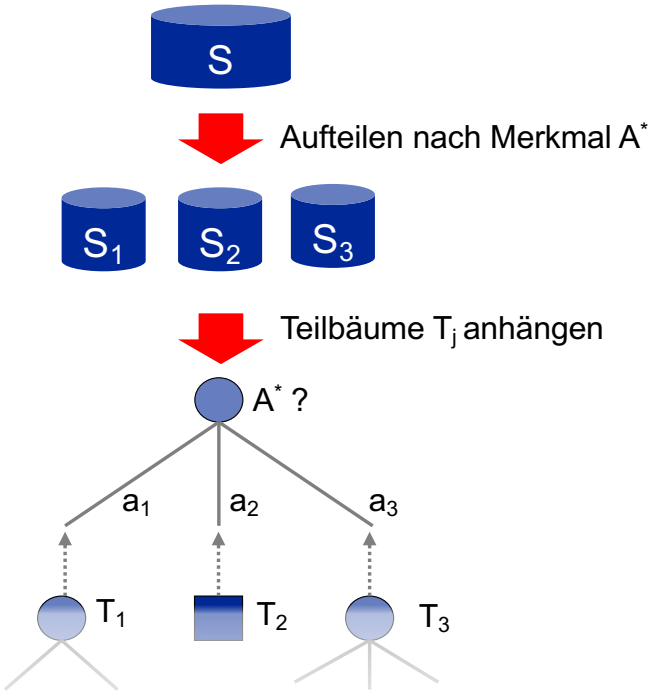
- outlook ∈ {sunny, overcast, rain}
- temperature ∈ {cool, mild, hot}
- humidity ∈ {high, normal}
- windy ∈ {true, false}

outlook	temperature	humidity	windy	play
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	flase	P
rain	mild	high	true	N

Allgemeiner Algorithmus zum Baumaufbau

Top Down Induction of Decision Trees

- **Input:** training set S with known class labels
- **Algorithm (recursive):**
  - If S is empty
    then create leaf node with decision according to default rule
  - If all instances belong to the same class
    then create leaf node with decision for this class
  - Else (instances belong to different classes) then
    - Choose attribute A^* that best splits data into subsets S_i
    - Grow decision trees T_i for each subset S_i
    - Construct decision tree T with subtrees T_i according to the splits
- **Output:** decision tree T



Welche Aufteilung ist die beste?

Idee: Erzeuge Teilmengen der Daten mit homogeneren Klassenverteilungen

Wie lässt sich die Homogenität oder Reinheit der Klassenverteilung messen?

> Informationstheorie (auf Basis der Entropie)
  - Informationsgewinn und GrainRatio → ID3, C4.5 und Nachfolger

> Statistische Kennzahlen zur Bestimmung der Abhängigkeit mit der Klasse
  - Chi-Quadrat-Kennzahl → CHAID (chi-squared automatic interaction detector)
  - Gini-Index → CART (classification and regression trees)

Über Information und Unsicherheit

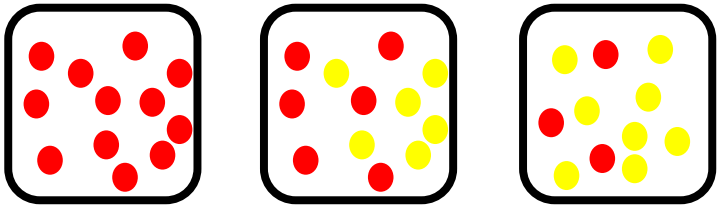
Was ist eine gute Aufteilung?

> Finde das Merkmal, das am meisten Information in Bezug auf die Zielgröße trägt oder gleichwertig
> Finde das Merkmal, das die Unsicherheit in Bezug auf die Zielgröße am stärksten reduziert

Wie lässt sich Information oder Unsicherheit messen?

Ganz intuitiv:

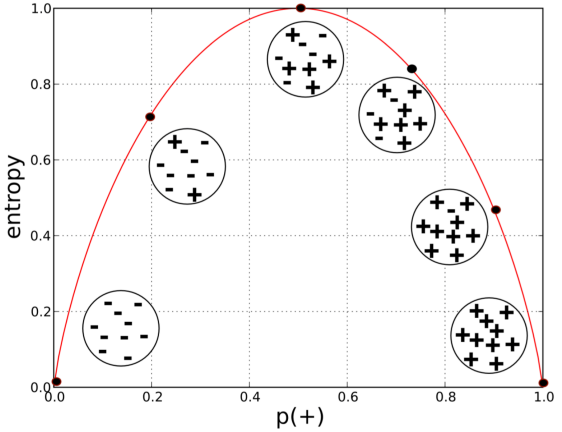
Wenn zufällig eine Kugel aus einer der Kisten entnommen wird, wie steht es um die Unsicherheit bezüglich der Farbe?



Entropie und Informationsgewinn

- Entropie misst die Homogenität der Klassenverteilung
- Für Datenmenge S mit c Klassen:

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$



- Der Informationsgewinn misst die erwarte Reduktion der Entropie durch eine Aufteilung nach Merkmal A:

$$\text{Gain}\ (S,\ A) = \text{Entropy}\ (S) - \sum_{v \in \text{Values}\ (A)} \frac{|S_v|}{|S|} \text{Entropy}\ (S_v)$$

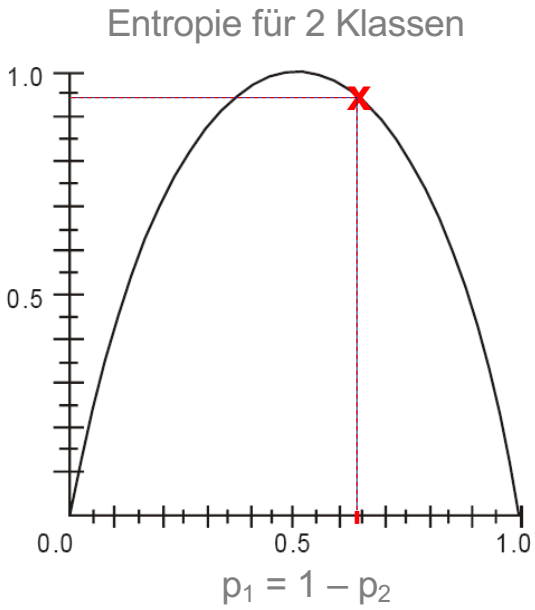
Entropie beim Tennis-Datensatz

Entropie der Klassenverteilung

> Datenmenge S mit 14 Beispielen (Tagen):
  - 9 positive (P)
  - 5 negative (N)

> Wahrscheinlichkeiten
  - p_1 = P(play=P) = 9/14
  - p_2 = P(play=N) = 1 - p_1 = 5/14

> Entropie von S
  - Entropie(p_1, p_2) = - 9/14 $\log_2$ 9/14 - 5/14 $\log_2$ 5/14
    ≈ 0.4098 + 0.5305 ≈ **0.9403**



Informationsgewinn bei Aufteilung nach „outlook“

> Teilmengen und Klassenverteilungen nach der Aufteilung:

Ausprägung	P	p_1	N	p_2	P+N	Entropie
sunny	2	0.4	3	0.6	5	0.9710
overcast	4	1.0	0	0.0	4	0.0000
rain	3	0.6	2	0.4	5	0.9710

> Entropie in den Teilmengen

– $\text{Entropie}(S_{sunny}) = \text{Entropie}(S\text{rain}) = -2/5 \cdot \log_2 2/5 - 3/5 \cdot \log_2 3/5 \approx 0{,}9710$

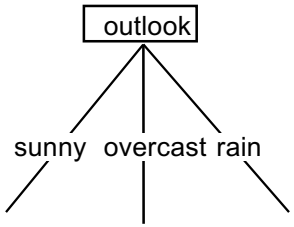
– $\text{Entropie}(S \mid \text{outlook}) = 5/14 \cdot \text{Entropy}(S_{sunny}) + 4/14 \cdot \text{Entropy}(S_{overcast}) + 5/14 \cdot \text{Entropy}(S_{rain})$
$\approx 5/14 \cdot 0.9710 + 4/14 \cdot 0 + 5/14 \cdot 0.9710 = 0.6935$

> Informationsgewinn

$\text{Gain}(S \mid \text{outlook}) = \text{Entropy}(S) - \text{Entropy}(S, \text{outlook}) \approx 0.9403 - 0.6935 = 0.2468$

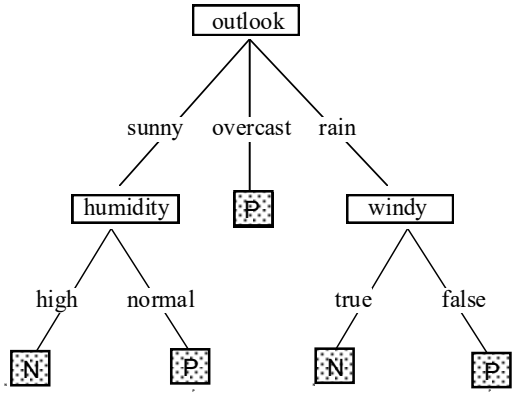
Auswahl der Aufteilung (Merkmal) beim Tennis-Datensatz

- Wähle das Merkmal mit dem größten Informationsgewinn
  - ***gain(S, outlook) = 0.2468***
  - ***gain(S, temperature) = 0.0292***
  - ***gain(S, humidity) = 0.1518***
  - ***gain(S, windy) = 0.0481***
- Wähle „outlook“ zum Aufteilen (Test) an der Wurzel
- Führe Baumaufbau rekursiv für die Teilmengen mit den Ausprägungen *sunny, overcast* und *rain* fort

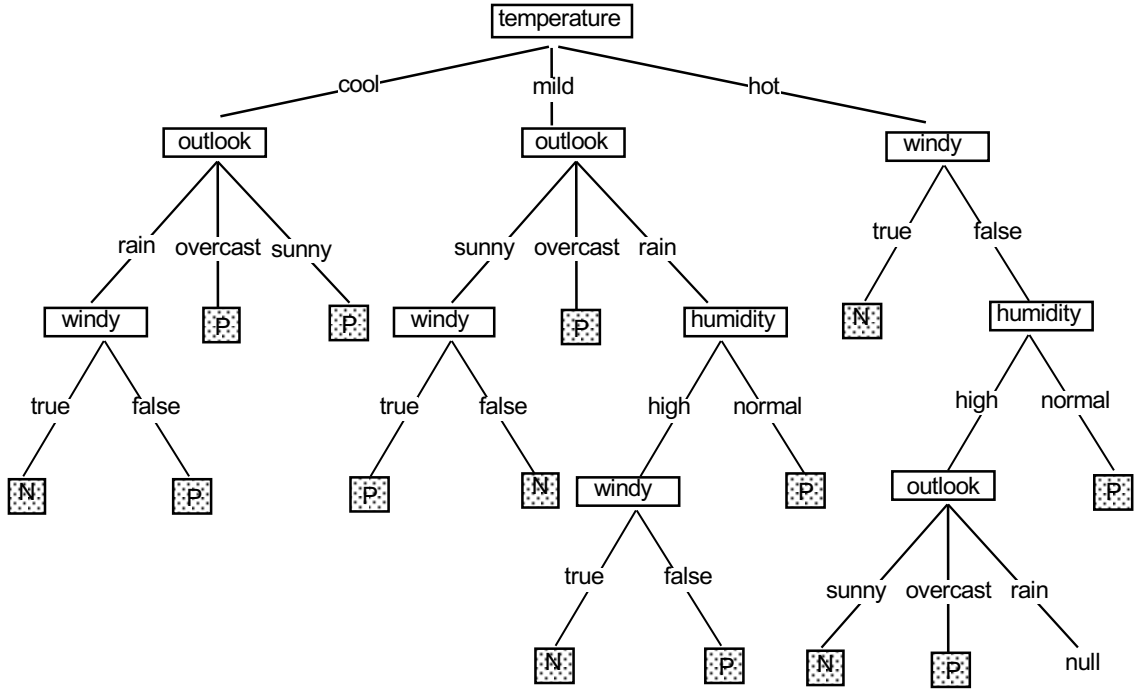


Entropiebasierte vs. zufällige Aufteilung

entropiebasierte Aufteilung



zufällige Aufteilung



GainRatio – das Informationsgewinnverhältnis

- Informationsgewinn bevorzugt Merkmale mit vielen Ausprägungen

  → breite Bäume

- Idee: Bestrafe Merkmale mit zu vielen Ausprägungen, damit Bäume kompakt bleiben

  → Informationsgewinnverhältnis (GainRatio):

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(S, A)} \quad \text{mit} \quad \text{SplitInfo}(S, A) = -\sum_{v \in \text{Values}(A)}^{m} \frac{|S_v|}{|S|} * log_2\left(\frac{|S_v|}{|S|}\right)$$

Varianten beim Baumaufbau

- Aufteilung → Binär oder beliebig?
- Auswahlkriterium → Welche Aufteilung (welches Merkmal) ist am besten?
- Pre-Pruning → Wann soll der Baumaufbau beendet werden?
- Post-Pruning → Sollen nach dem Baumaufbau nützliche Zweige entfernt werden?
- Gruppierung → Sollen Merkmale mit vielen Ausprägungen gruppiert werden? Wie?

Merkmale mit vielen möglichen Ausprägungen

Wenn für jede Ausprägung eines Merkmals verzweigt wird, wird der Baum sehr breit

→ Gruppierung von Attributwerten

- **Nominale Merkmale**
  Jede Form der Gruppierung möglich, insbesondere eine Ausprägung vs. alle anderen
- **Ordinale Merkmale**
  Die natürliche Ordnung sollte bei der Gruppierung berücksichtigt werden
- **Numeric attributes**
  Gruppierung erzeugt Intervalle → Diskretisierung, Binning

Entscheidungsbäume – Overfitting und Pruning

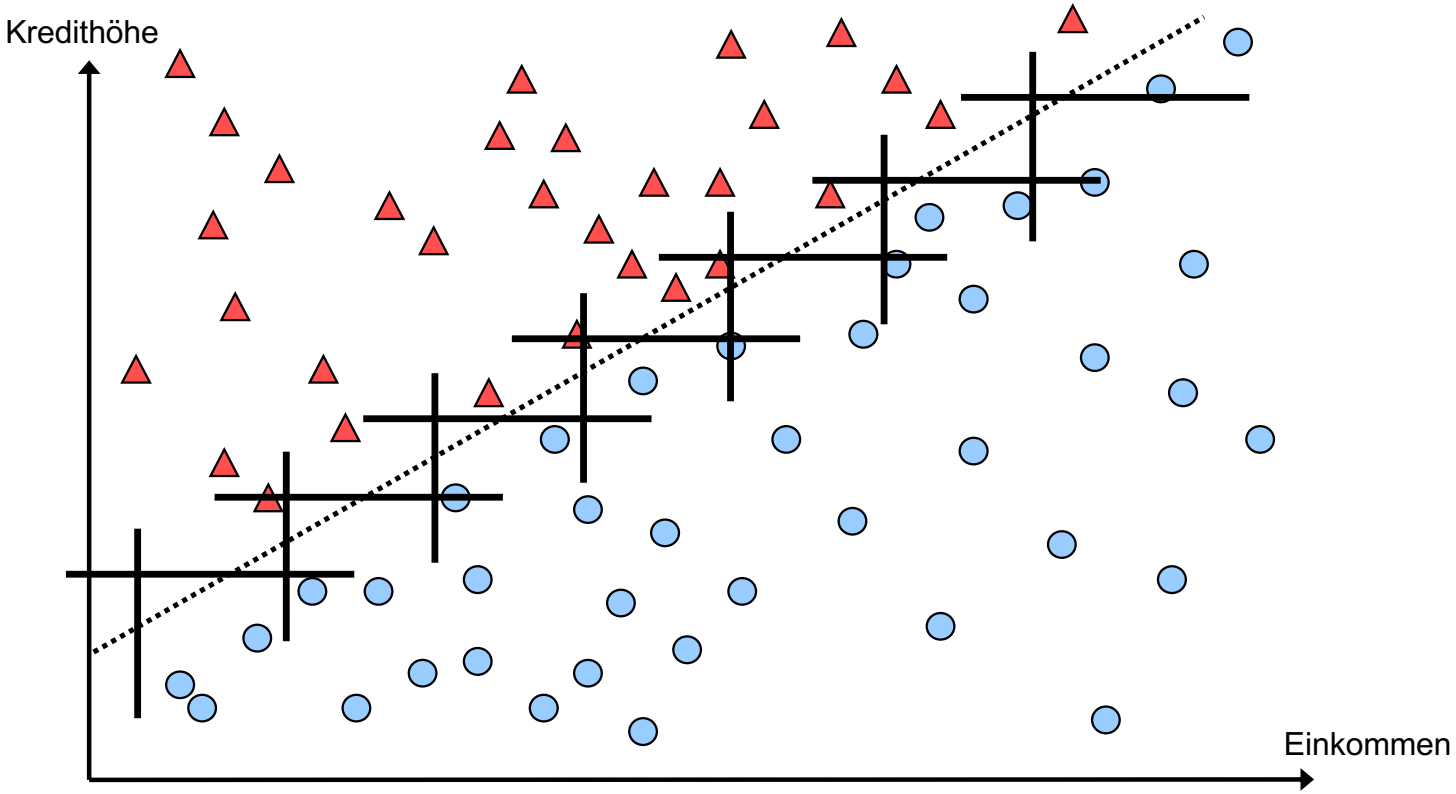
Entscheidungsbäume neigen zum Overfitting (Überanpassung an die Trainingsdaten)

- Ein Baum, der zu sehr an die Trainingsdaten angepasst ist, hat zu viele Verzweigungen
  - Der Baum ist spezifisch, hat ggf. einzelne Lernbeispiele auswendig gelernt
  - Generalisierungsfähigkeit leidet, Leistungsfähigkeit bei neuen Daten deutlich schlechter
- Pruning (Stutzen) vermeidet oder reduziert Overfitting
  - **Pre-Pruning**: Baumaufbau frühzeitig stoppen
    - Z.B. wenn zu wenig Daten in einem Knoten vorliegen
    - Allgemein: Wenn eine Aufteilung statistisch nicht signifikant ist
  - **Post-Pruning**: Einzelne Zweige nach dem vollständigen Aufbau wieder entfernen
    - Z.B. solange die Leistungsfähigkeit auf Validierungsdaten nicht abnimmt Zweige entfernen

Vorteile von Entscheidungsbäumen

- ✓ Einsetzbar für Klassifikation und Regression
- ✓ Schnelles Lernverfahren, skaliert auf große Datenmengen
- ✓ Kann mit gemischte Skalenniveaus umgehen
- ✓ Ignoriert redundante Merkmale (robust)
- ✓ Kann mit fehlenden Werten umgehen (separate Verzweigung)
- ✓ Kann nicht-lineare Entscheidungsgrenzen abbilden
- ✓ Kleine Bäume lassen sich gut interpretieren
- ✓ Vorhersagegüte kann mit vielen Lernverfahren mithalten

Nachteil von Entscheidungsbäumen



→ **Problem:**
Entscheidungsgrenzen verlaufen ausschließlich parallel zu den Achsen des Eingaberaums

→ **Abhilfe:**
Feature Engineering – Konstruktion von Merkmalen die, den Zusammenhang besser beschreiben

Zusammenfassung

- Klassifikation ist die häufigste Data-Mining-Aufgabe
- Lösung Klassifikation durch überwachtes Lernen
- Zahlreiche Lernverfahren stehen zur Verfügung
- Es gibt kein Verfahren, das immer das beste ist (*no-free-lunch-Theorem*)
- K-Nächste-Nachbar-Klassifikation ist sehr intuitiv
- Adäquate Nachbarschaftssuche erfordert sorgfältige Datenvorverarbeitung
- Entscheidungsbaum-Verfahren liefern auch ohne besondere Datenaufbereitung und Parametereinstellung konkurrenzfähige Ergebnisse
- Entscheidungsbäume ist anfällig für Overfitting, Abhilfe durch Pruning
- Unterstützung des Baumaufbaus durch Konstruktion höherwertigerer Merkmale