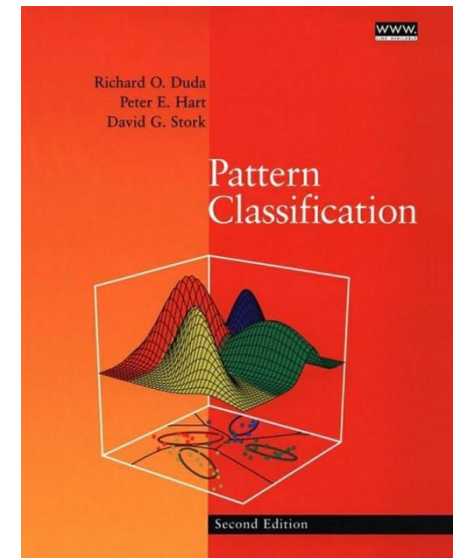




Sensordatenverarbeitung

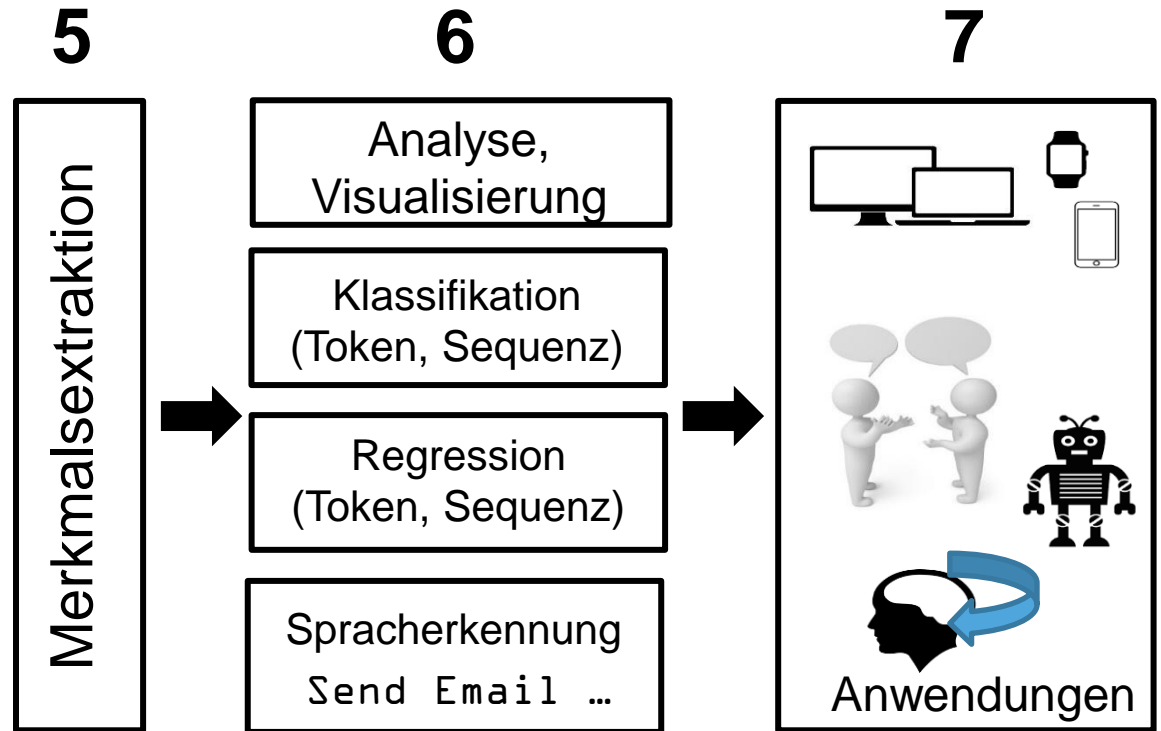
KLASSIFIZIERUNG (11)

06.01.2025



Teila

Nr.	Thema	Icons
1	Einleitung; einführende Beispiele	
2	Datenaufnahme; Audio-Datenaufnahme	
3	Bild-Datenaufnahme	
4	Farbe, Segmentierung, Segmentierungsgetriebene BV	
5	Audiosignal, 1D Frequenzraum, Fouriertransformation	
6	Koordinatensysteme; Bewegungs-Datenaufnahme	
7	2D Frequenzraum, 2D Filter	
8	Kanten, SdV-Paradigmen, direkte Bildmerkmale	
9	Houghtransformation, Bewegungsmerkmale	
10	Audiomerkmale	
11	Klassifizierungsalgorithmen	
12	Entwicklung und Evaluation sensorbasierter Systeme	
13	Bayes-Schätzung & Bayes-Filter	



DIGITALES SIGNAL

Zeit Frequenz

6

Analyse,
Visualisierung

Klassifikation
(Token, Sequenz)

Regression
(Token, Sequenz)

Spracherkennung
Send Email ...

DIGITALES SIGNAL

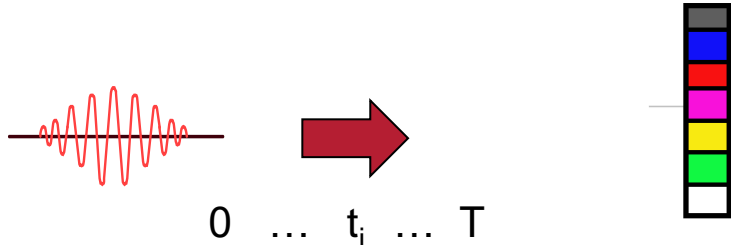
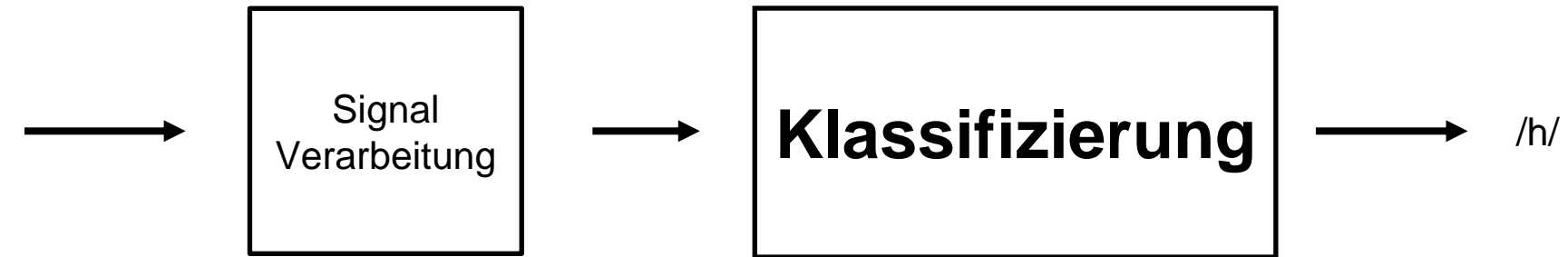
Zeit Frequenz

Klassifikation (Token, Sequenz)

- **Klassifizierung** bezeichnet das Zusammenfassen von Objekten zu Klassen (Gruppen, Mengen, Kategorien)
- Klassifizierung ist also **der Vorgang der Zuordnung von Klassen**
- Das Endprodukt ist die **Klassifikation** der Objekte
- Die Begriffe werden im Sprachgebrauch allerdings uneinheitlich verwendet, bzw. häufig Klassifizierung und Klassifikation gleichbedeutend verwendet
- Man unterscheidet die Klassifizierung von Tokens vs. Sequenzen
 - Bei Token-Klassifikation werden einzelne Merkmalsvektoren einer Klasse zugeordnet
 - Bei Sequenzen-Klassifikation werden Sequenzen von Merkmalsvektoren in ihrem Kontext zugeordnet

Eingabe
Sprachsignal

Ausgabe
Klasse



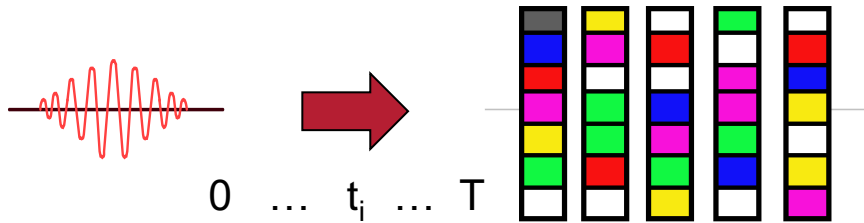
Token, d.h.
Merkmalsvektor
zum Zeitpunkt t_i

Ordne diesem Vektor **eine Klasse**
(aus einer endlichen vordefinierten
Menge von Klassen) zu

Beispiele:

- **Audio:** Laute einer Sprache {/a/, /e/, ...}
- **Video:** Objekte {Fußgänger, Auto, ...}
- **Bewegung:** Aktivitäten {Sitzen, Gehen, ...}

Eingabe
Sprachsignal



Ordne jedem Vektor der Sequenz je **eine Klasse** zu

In der Sequenzklassifikation werden Entscheidungen idR im Kontext getroffen, d.h. $p(h|\text{Wortanfang})$, $p(e|h)$, $p(l|h,e)$, $p(o|h,e,l)$...

- Token Klassifizierung klassifiziert individuelle Merkmalsvektoren als Laute
 - ABER: Laute, Gesten, Aktivitäten sind länger als EIN Merkmalsvektor
 - Laute variieren in Abhängigkeit der Nachbarlaute (Koartikulation)
- Außerdem:
- Sollen in der Spracherkennung ganze Sätze oder Wörter erkannt werden
 - Man könnte die einzelnen Laute zu Wörtern zusammenbauen
 - ABER nicht alle Lautsequenzen ergeben Wörter
- Wissen über Lautsequenzen einbauen, die Wörtern entsprechen
- Variationsmöglichkeiten einzelner Laute zulassen
- Klassischer Ansatz zur Lösung dieser Probleme sind **Hidden-Markov-Modelle (HMM)**



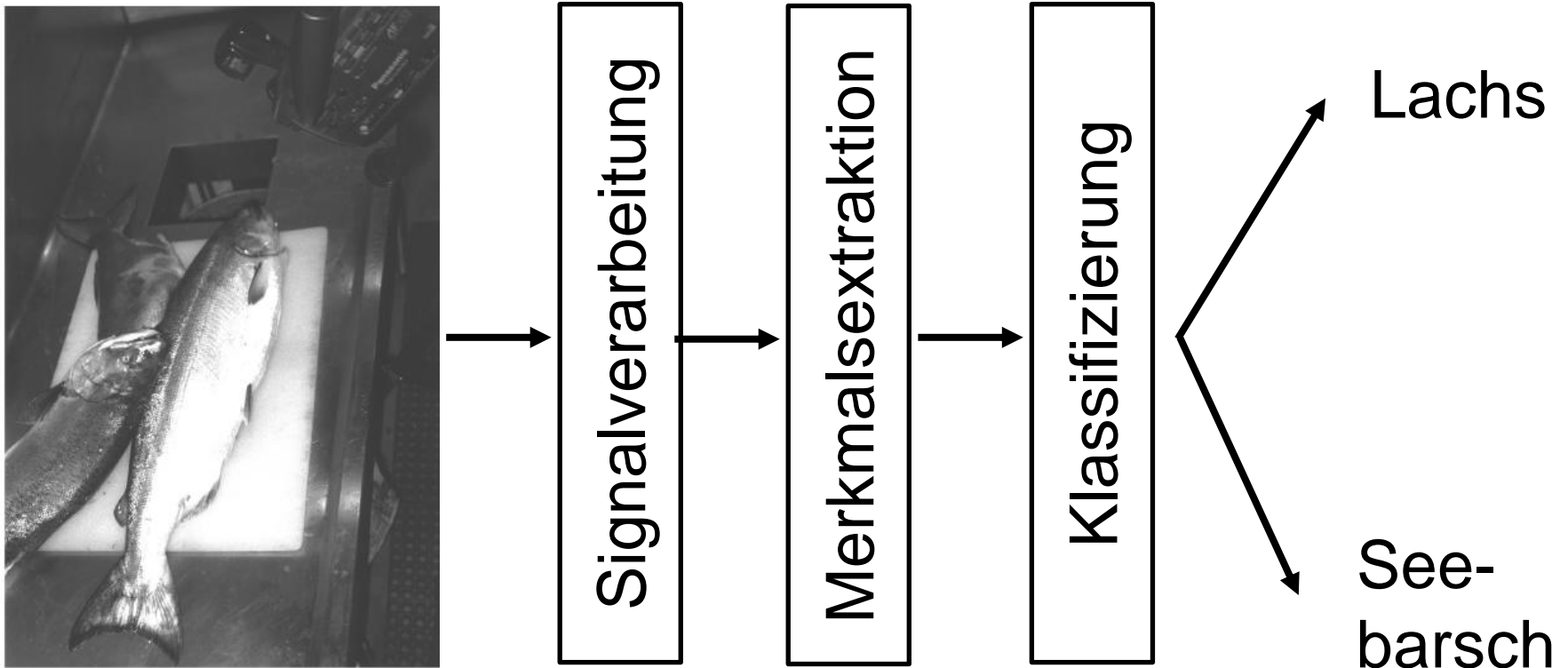
- Können wir in dieser Vorlesung aus Zeitgründen leider nicht besprechen
 - Automatische Spracherkennung (jedes WS, Masterkurs)
 - Biosignale und Benutzerschnittstellen (jedes SS, Bachelorkurs)

- Zur Klassifizierung benötigt man eine Instanz (hier Box), die die Merkmalsvektoren in Klassen einordnet

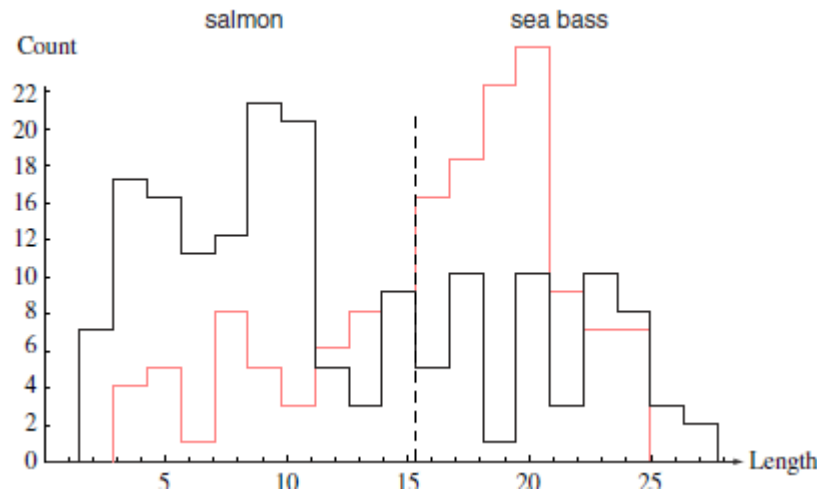
Klassifizierung

- Diese Instanz nennt man einen **Klassifikator**
- In der Informatik, insbesondere der Mustererkennung ist dieser Klassifikator meist ein **Algorithmus** oder eine **mathematische Funktion** (im Unterschied zu mechanischen Klassifikatoren, die bspw. Münzen oder Schrauben nach Größe sortieren)
- Meist wird nicht zwischen Verfahren und Instanz unterschieden, daher werden die Begriffe **Klassifikator** und Klassifikations- / **Klassifizierungsverfahren** gleichbedeutend verwendet.

- Beispiel: Eine Fischfabrik will gefangene Fische **automatisch zu einer von zwei Klassen zuordnen**:
Lachs (salmon) oder Seebarsch (sea bass)

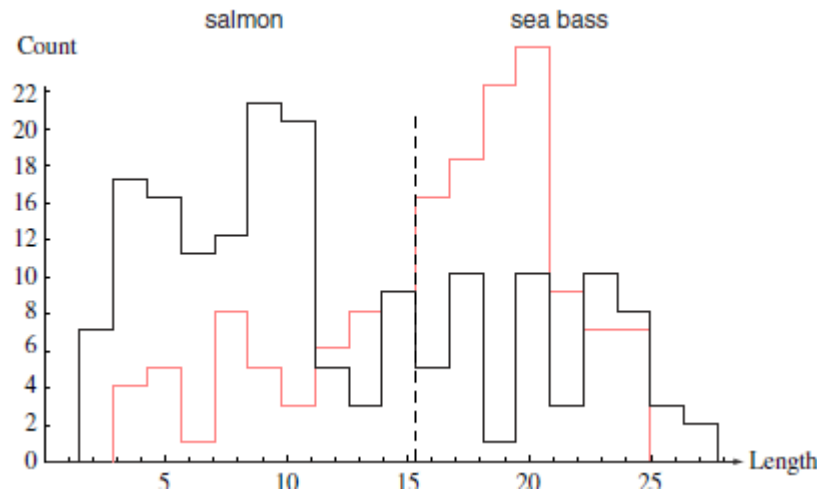


- Die Vorverarbeitung erzeugt **Merkmale**, z.B. die **Länge** eines Fisches
- Betrachte Häufigkeiten des Merkmals „Länge“ über alle Daten (Fische)
- Darstellung der Häufigkeiten von Merkmalen nennt man **Histogramm**
- Finde eine **Entscheidungsgrenze** im Histogramm (siehe Abbildung):
 - Grenze (gestrichelte Linie): Ab Länge 15 cm ist es ein Seebarsch



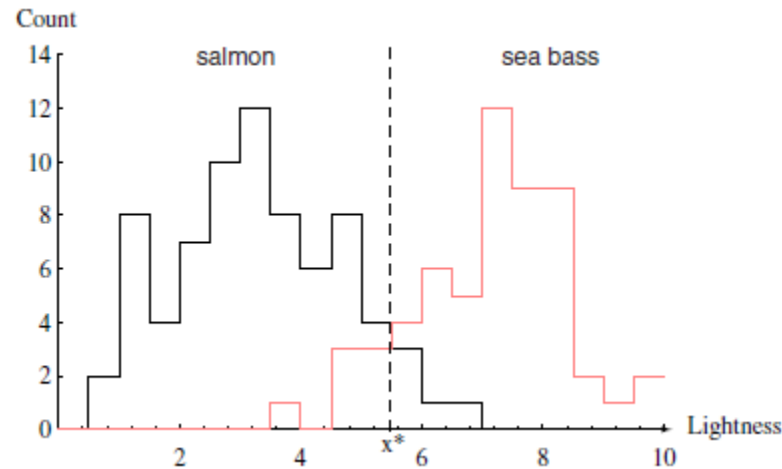
- **Frage an alle: seht ihr ein Problem? Welches??**

- Die Vorverarbeitung erzeugt **Merkmale**, z.B. die **Länge** eines Fisches
- Betrachte Häufigkeiten des Merkmals „Länge“ über alle Daten (Fische)
- Darstellung der Häufigkeiten von Merkmalen nennt man **Histogramm**
- Finde eine **Entscheidungsgrenze** im Histogramm (siehe Abbildung):
 - Grenze (gestrichelte Linie): Ab Länge 15 cm ist es ein Seebarsch

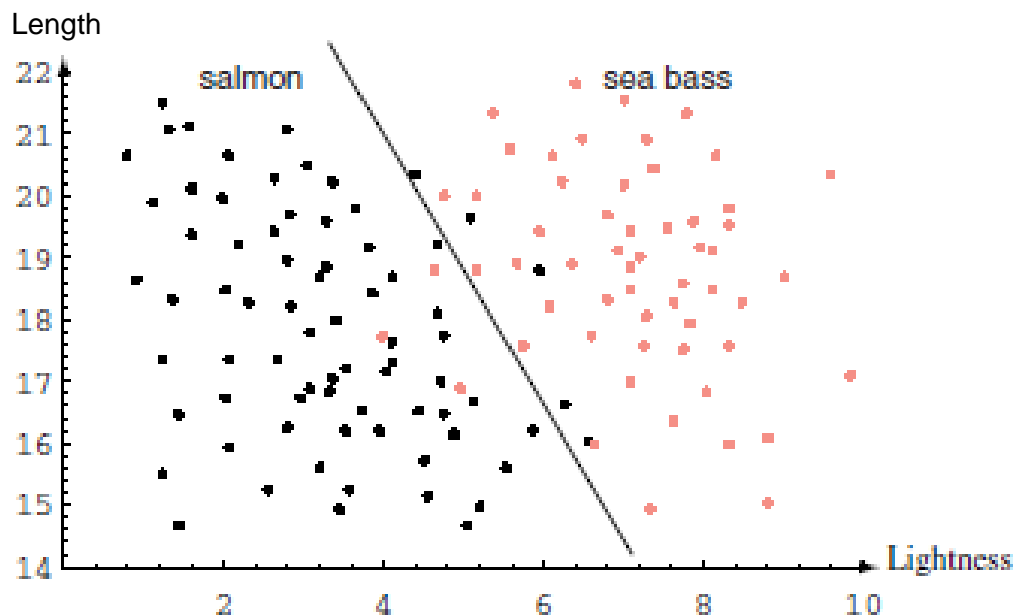


- **Histogramme überlappen sich**
⇒ **Seebarsche <15cm und Lachse >15cm werden falsch zugeordnet**

- Um die Zuordnung zu verbessern, verwende ein alternatives Merkmal
 - Helligkeit der Haut: **Lightness**
- Weniger Überlappungen der Histogramme, aber immer noch einige Fehler



- Betrachte den **zweidimensionalen Merkmalsraum**, der durch beide Merkmale **Länge** und **Helligkeit** aufgespannt wird
- Aus einer Entscheidungsgrenze wird eine Entscheidungsgerade
- Jetzt gibt es deutlich weniger Fehler in der Zuordnung der Fische



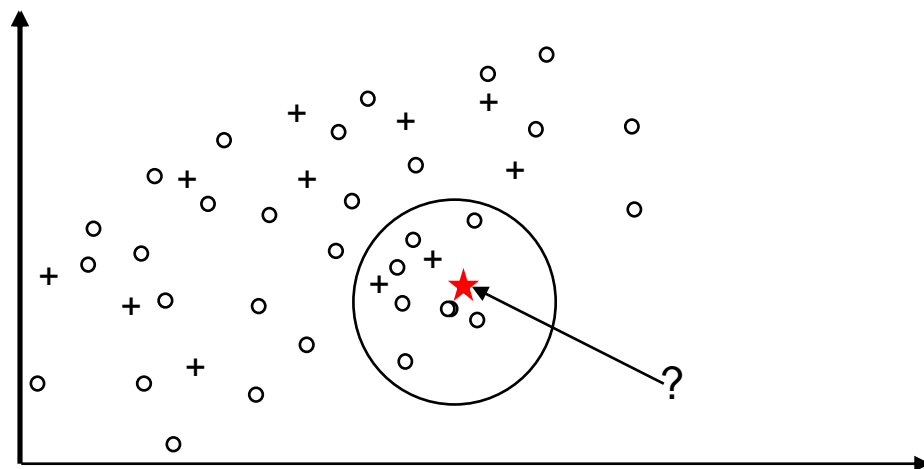
- Aber wie findet man die Entscheidungsgrenze?

Statt einer Entscheidungsgrenze (die bestimmt werden muss), entscheidet man entsprechend der Nachbarschaft des zu klassifizierende Fische

Klassifizierungsalgorithmus:

- Betrachte die k nächsten Nachbarn des zu klassifizierenden Fische (★)
- Bestimme die häufigste Klasse unter diesen k Nachbarn
- Weise den Fisch ★ dieser häufigsten Klasse zu
- Beispiel: $k=9$, 7x Lachs (Kreis), 2x Seebarsch (Kreuz)
=> Fisch ★ wird als Lachs klassifiziert
- Dieser Algorithmus heisst **k-Nearest-Neighbor = kNN**

• **Probleme, Herausforderungen?**



Statt einer Entscheidungsgraden (die bestimmt werden muss), entscheidet man entsprechend der Nachbarschaft des zu klassifizierende Fische

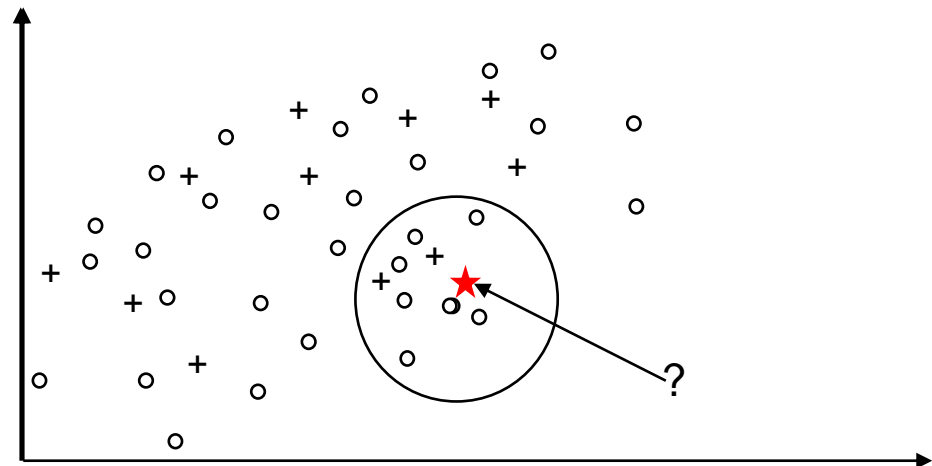
Klassifizierungsalgorithmus:

- Betrachte die k nächsten Nachbarn des zu klassifizierenden Fische (\star)
- Bestimme die häufigste Klasse unter diesen k Nachbarn
- Weise den Fische \star dieser Klasse zu
- Beispiel: $k=9$, 7x Lachs (Kreis), 2x Seebarsch (Kreuz)
=> Fische \star wird als Lachs klassifiziert
- Dieser Algorithmus heisst k-Nearest-Neighbor = kNN

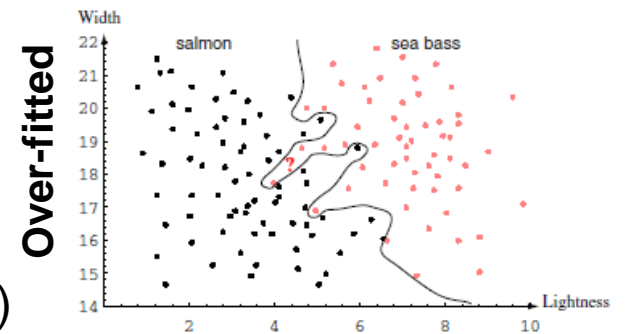
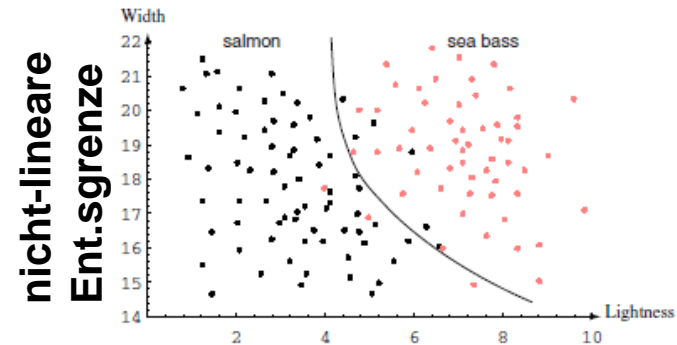
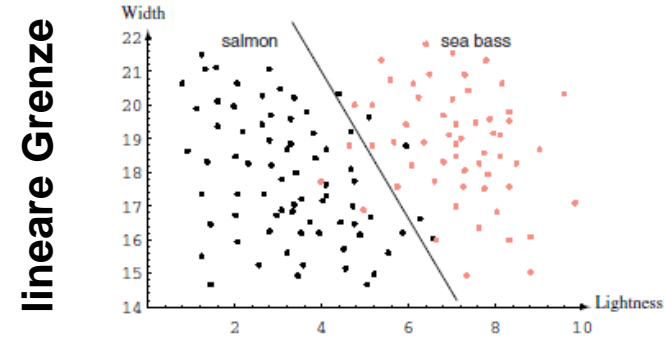
• **Probleme, Herausforderungen?**

• **Wahl von k**

• **Es müssen möglichst viele annotierte Daten (hier Kreise, Kreuze) vorgehalten werden**



- Bei Lösung 3 könnte man die Entscheidungsgerade ($y=ax+b$, also 2 Parameter) lernen
 - Dann müsste man keine Daten mehr vorhalten (kNN), sondern hätte **ein Modell** bzw. **eine Funktion** (hier Gerade)
- Allerdings: beste Entscheidungsgrenze muss nicht unbedingt eine Gerade sein
 - **Lineare vs. Nicht-lineare Grenzen**
- Bestimmen guter Entscheidungsgrenzen ist **das zentrale Thema** beim Klassifizieren
 - Trainieren eines Klassifikators = Lernen der Entscheidungsgrenze(n) aus Trainingsdaten
- Herausforderung: optimale Grenzen, d.h. gute Generalisierung (passende # Parameter)
 - Nicht Überspezialisieren (**over-fitting**)
 - Keine dominanten Trends übersehen (**under-fitting**)



- Mustererkennung = Klassifizierungsverfahren
- Mustererkennung = Entwicklung von Klassifikatoren
- Mustererkennung = Lernen von Entscheidungsgrenzen

Moderner Begriff für Probleme der Mustererkennung: **Machine Learning (ML)**

