Angewandte Statistik

Julian Hinz — Universität Bielefeld

Session 7

Klassifikation — Supervised Machine Learning

Lernziel

Bisher

- Wie kann ich eine Zielgröße “am besten” erklären?
  → parametrische und nicht-parametrische Regression

Heute

- Welches Label sollte ich einer Beobachtung geben?

Lernziel

- Einführung in machinelles Lernen
- Verstehen der grundlegenden Konzepte der Klassifikation
- Kennenlernen verschiedener Klassifikationsmethoden

Machine Learning

> “Machine Learning is the science of getting computers to learn without being explicitly programmed.”
>
> — Arthur Samuel, 1959

Ein Computer spielt Dame

- einfaches Spiel, aber Strategie
- Computer mit wenig Rechenpower: Entscheidungsbaum “pruning”
- Belohnungsfunktion mit höchster Gewinnwahrscheinlichkeit
- Machine learning: Programm erinnert sich an alle bisherige Positionen

Machine Learning

- Programmieren von Computern, um ein Leistungskriterium zu optimieren, indem Beispiel-Daten oder vergangene Erfahrungen verwendet werden
- beschäftigt sich mit der Frage, wie man Computerprogramme erstellt, die sich automatisch durch Erfahrung verbessern

Machine Learning Definition

> “Ein Computerprogramm *lernt* aus Erfahrung E in Bezug auf eine Klasse von Aufgaben T und ein Leistungsmaß P, wenn sich seine Leistung bei Aufgaben T, gemessen an P, mit Erfahrung E verbessert.”

Beispiele

Handschrifterkennung:

- **Aufgabe T**: Erkennung und Klassifizierung handgeschriebener Wörter in Bildern
- **Leistungsmaß P**: Prozentsatz der korrekt klassifizierten Wörter
- **Erfahrung E**: Ein Datensatz handgeschriebener Wörter mit Klassifikationen

Autonomes Fahren:

- **Aufgabe T**: Fahren auf Autobahnen mit visuellen Sensoren
- **Leistungsmaß P**: Durchschnittliche zurückgelegte Distanz vor einem Fehler
- **Erfahrung E**: Eine Sequenz von Bildern und Lenkbefehlen, die beim Beobachten eines menschlichen Fahrers aufgezeichnet wurden

Machine learning heute

Vielzahl von Anwendungen:

- Bilderkennung
- Spracherkennung
- Verarbeitung natürlicher Sprache
- Empfehlungssystemen
- …

Arten von Machine Learning

1. **Überwachtes Lernen (Supervised Learning)**: Training eines Modells mit gekennzeichneten Daten
2. **Unüberwachtes Lernen (Unsupervised Learning)**: Training eines Modells mit ungekennzeichneten Daten
3. **Bestärkendes Lernen (Reinforcement Learning)**: Training eines Modells durch Versuch und Irrtum

Überwachtes Lernen (Supervised Learning)

- Lernalgorithmus mit Datensätzen trainiert und validiert, die für jede Eingabe einen passenden Ausgabewert enthalten
- im Vorhinein festgelegten zu lernenden Ausgabe, deren Ergebnisse bekannt sind
- Ergebnisse des Lernprozesses können mit den bekannten, richtigen Ergebnissen verglichen werden, also „überwacht“ werden
- Datensätze werden als “markiert” oder “gelabelt” bezeichnet

Lernphase und Validierung

- **Lernphase**: statistisches Modell mit Teil der Beispieldaten (Trainingsdatensatz) aufgebaut
- **Validierung**: Qualität des Modells mit anderem Teil der Beispieldaten (Testdatensatz) überprüfen
- Ziel: Modell soll auch bei neuen, unbekannten Daten gefordertes Verhalten zeigen
- Herausforderung: Modell muss gut an die Trainingsdaten angepasst sein, dabei Overfitting vermeiden

Anwendungsbeispiele

- **Regression**: Vorhersage eines kontinuierlichen Werts
  → Vorhersage der Mietpreise basierend auf Merkmalen wie Größe und Lage
- **Klassifikation**: Zuordnung von Eingaben zu vordefinierten Klassen
  → Spam-Erkennung in E-Mails

Beispiel

- **Daten**: Geschlecht und Alter der Patienten
- **Labels**: „gesund“ oder „krank“

Geschlecht	Alter	Label
M	48	krank
M	67	krank
F	53	gesund
M	49	krank
F	32	gesund
M	34	gesund
M	21	gesund

Unüberwachtes Lernen (Unsupervised Learning)

- Algorithmus soll Muster und Strukturen in Datensätzen erkennen, die keine Labels enthalten
- Algorithmus muss eigenständig sinnvolle Gruppen oder Strukturen in den Daten finden

Anwendungsbeispiele

- **Clustering**: Gruppierung von ähnlichen Datenpunkten
  → z.B. Kundensegmentierung im Marketing
- **Dimensionalitätsreduktion**: Reduktion der Anzahl von Variablen
  → z.B. Auswahl von makroökonomischen Indikatoren (Inflation, Arbeitslosenquote, BIP, Zinssätze, etc.) zur Prognose der wirtschaftlichen Entwicklung

→ nächste Woche mehr!

Bestärkendes Lernen (Reinforcement Learning)

- ein “Agent” lernt durch Interaktion mit einer Umgebung, indem er Belohnungen maximiert
- Agent lernt durch “Trial and Error”, welche Aktionen die höchsten Belohnungen bringen
- Ziel: Lernen einer optimalen Strategie, die die langfristige Belohnung maximiert
- Herausforderung: Balanceakt zwischen Erkundung (neue Aktionen ausprobieren) und Ausnutzung (bekannte, belohnende Aktionen wiederholen)

Anwendungsbeispiele

- **Spielstrategien**: Entwicklung von Algorithmen, die Brettspiele wie Schach oder Go spielen und gewinnen können
- **Robotersteuerung**: Lernen von Bewegungsabläufen und Handlungen zur Optimierung der Ausführung

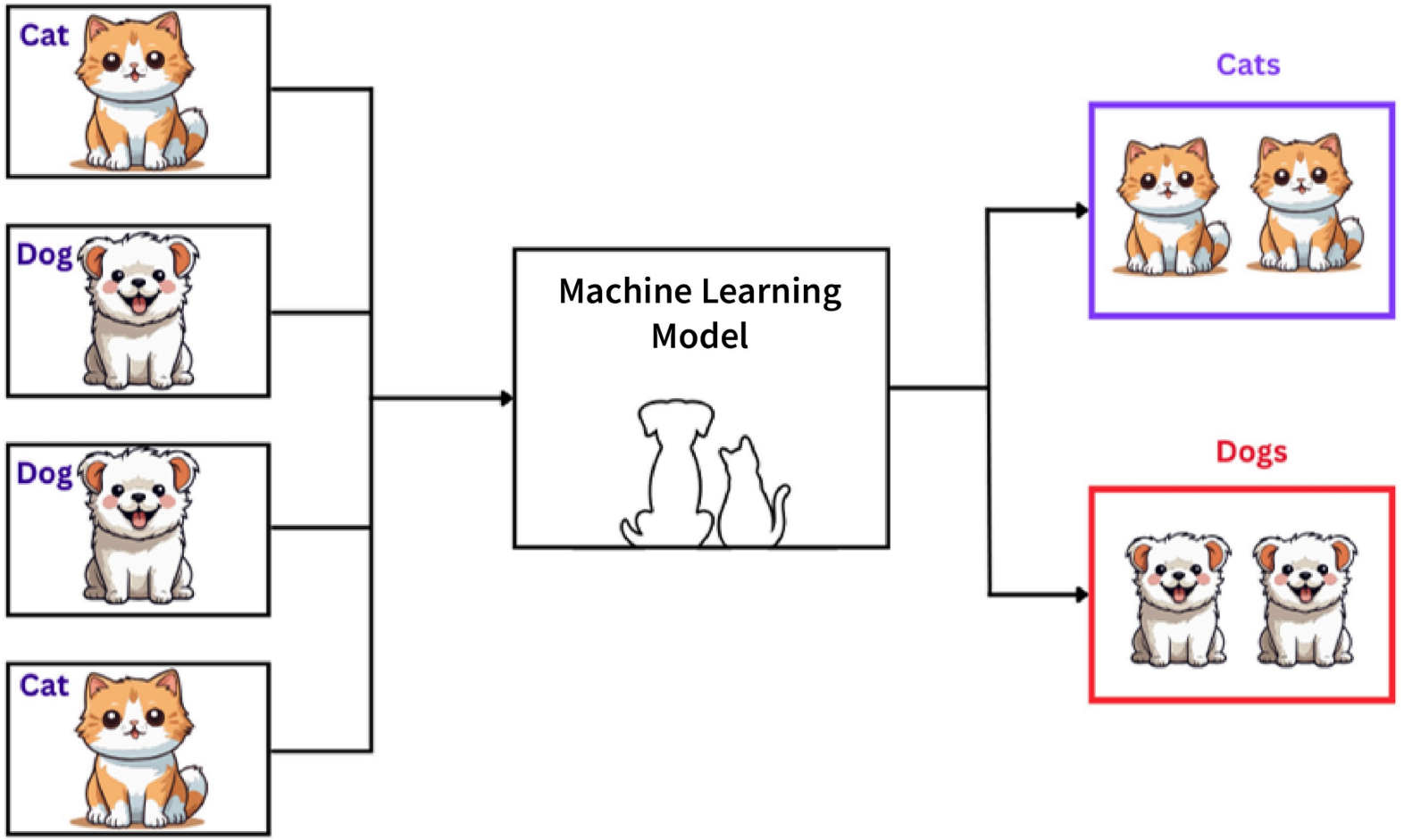
Klassifikation

(Supervised Machine Learning)

Arten von überwachtem Lernen

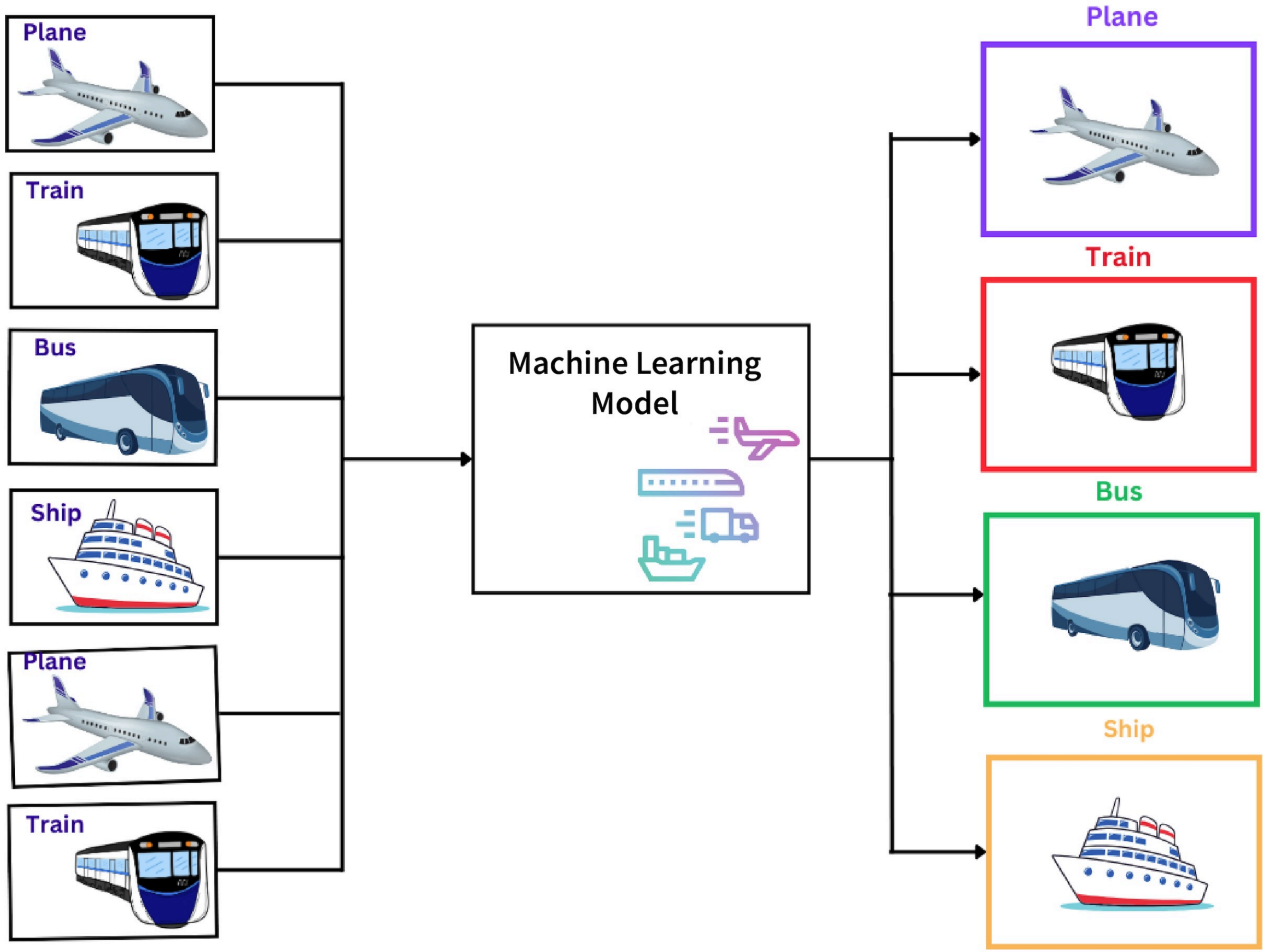
- Binäre Klassifikation: eine von zwei möglichen Kategorien
- Multiklassen-Klassifikation: eine von drei oder mehr Kategorien
- Multilabel-Klassifikation: Komplexere Szenarien, ein Input kann mehreren Labels zugeordnet werden

Binäre Klassifikation



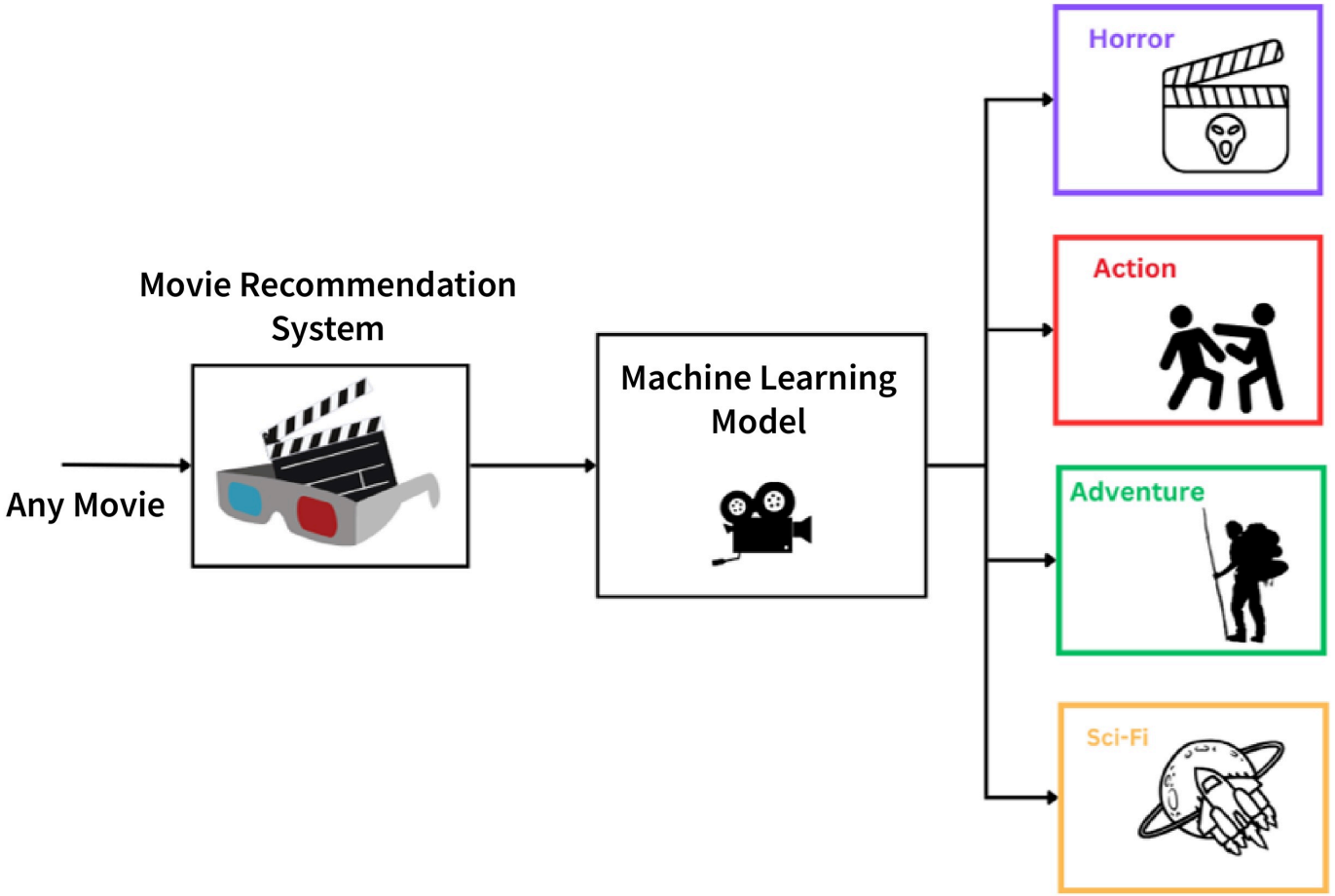
Quelle: scaler.com

Multiklassen-Klassifikation



Quelle: scaler.com

Multilabel-Klassifikation



Quelle: scaler.com

Methoden

- Logistische Regression
- Entscheidungsbäume (und Random Forests)
- K-Nearest Neighbors (k-NN)
- Naive Bayes
- Support Vector Machines (SVM)

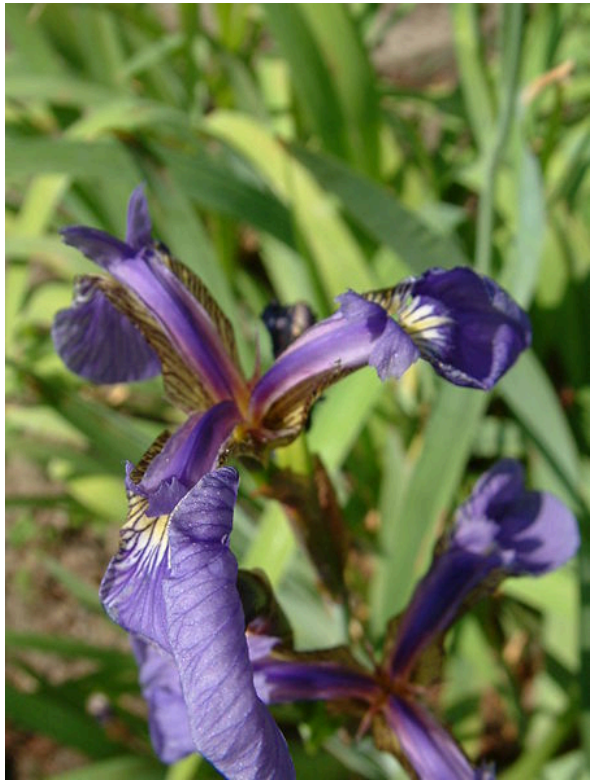
Evaluierung

- Unterschiedliche Formen der Performance Messung
- Am einfachsten: Konfusionsmatrix

Wahrheitsmatrix (Konfusionsmatrix)

	Person ist krank $(r_\mathrm{p} + f_\mathrm{n})$	**Person ist gesund** $(f_\mathrm{p} + r_\mathrm{n})$	
Test positiv $(r_\mathrm{p} + f_\mathrm{p})$	richtig positiv (r_p)	falsch positiv (f_p)	**Σ: 100 % der positiven Tests**
Test negativ $(f_\mathrm{n} + r_\mathrm{n})$	falsch negativ (f_n)	richtig negativ (r_n)	**Σ: 100 % der negativen Tests**
	Σ: 100 % der kranken Personen	**Σ: 100 % der gesunden Personen**	

Datensatz für die Beispiele: iris

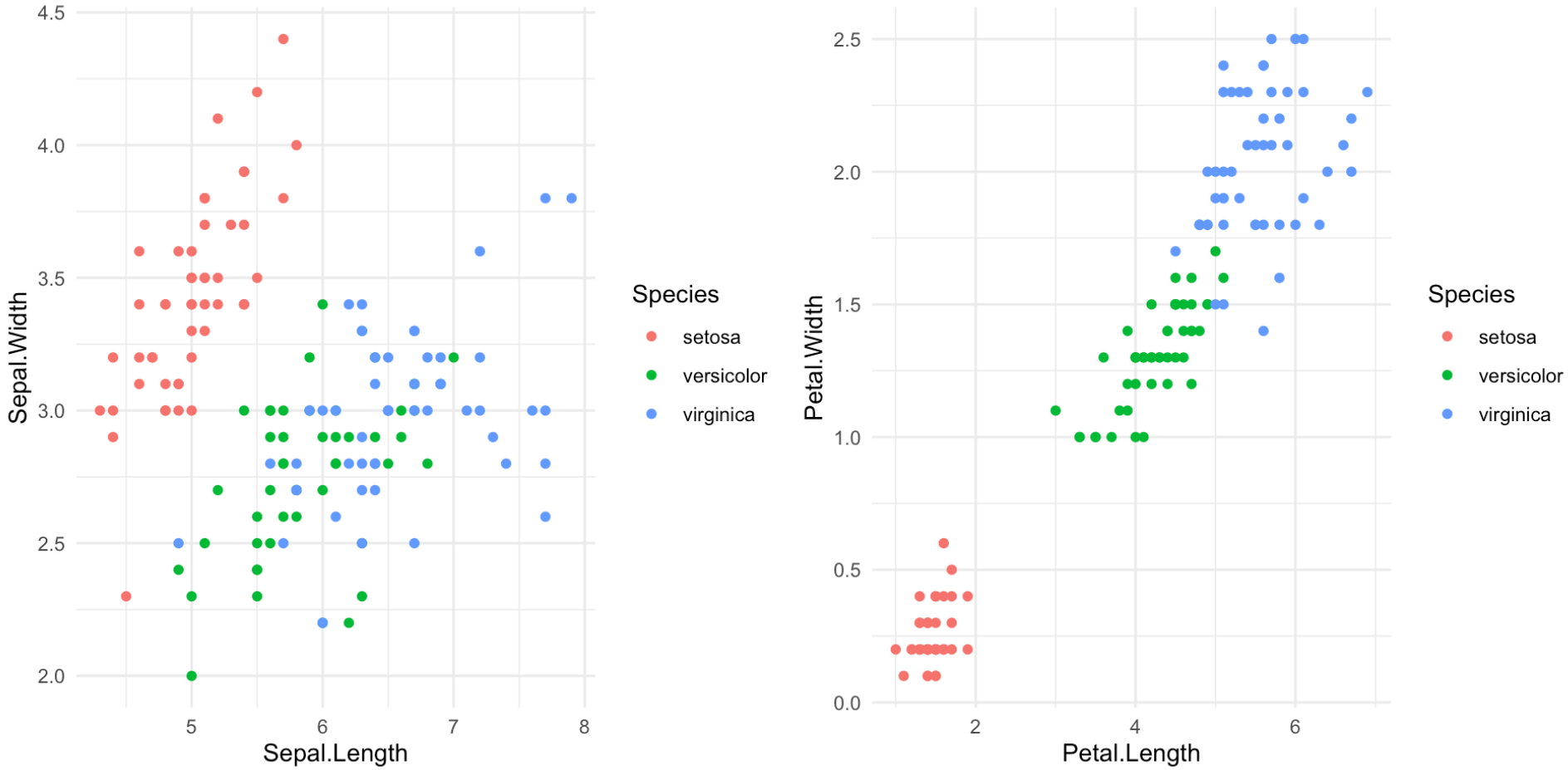


Iris setosa

Iris versicolor

Iris virginica

Datensatz für die Beispiele: iris



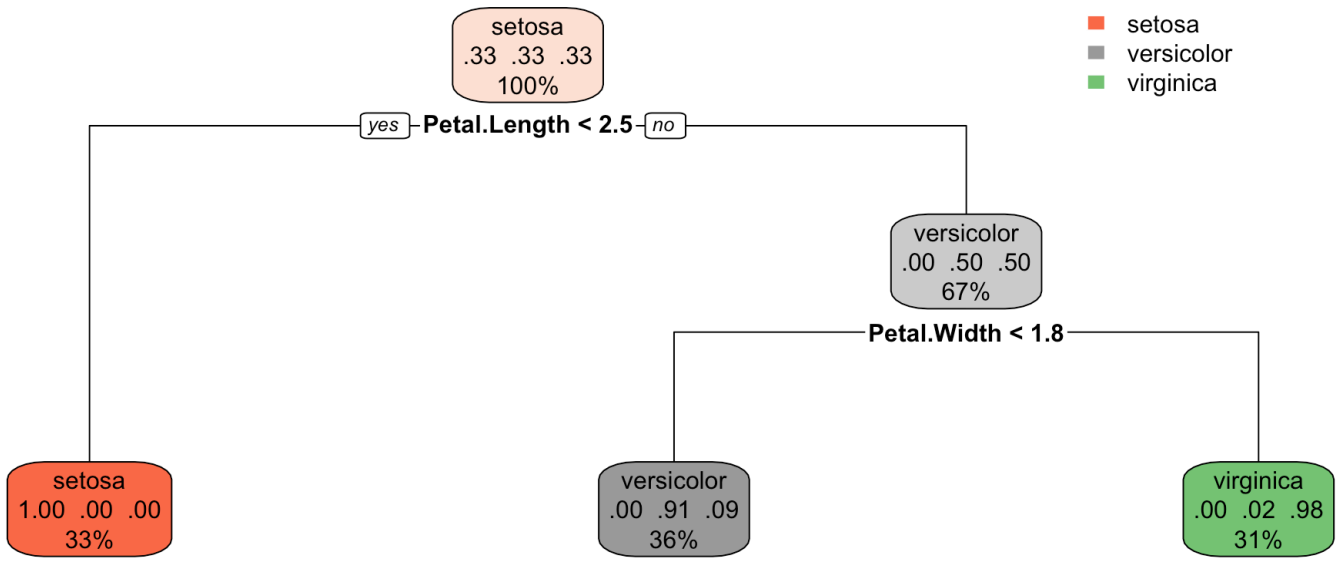
Entscheidungsbäume

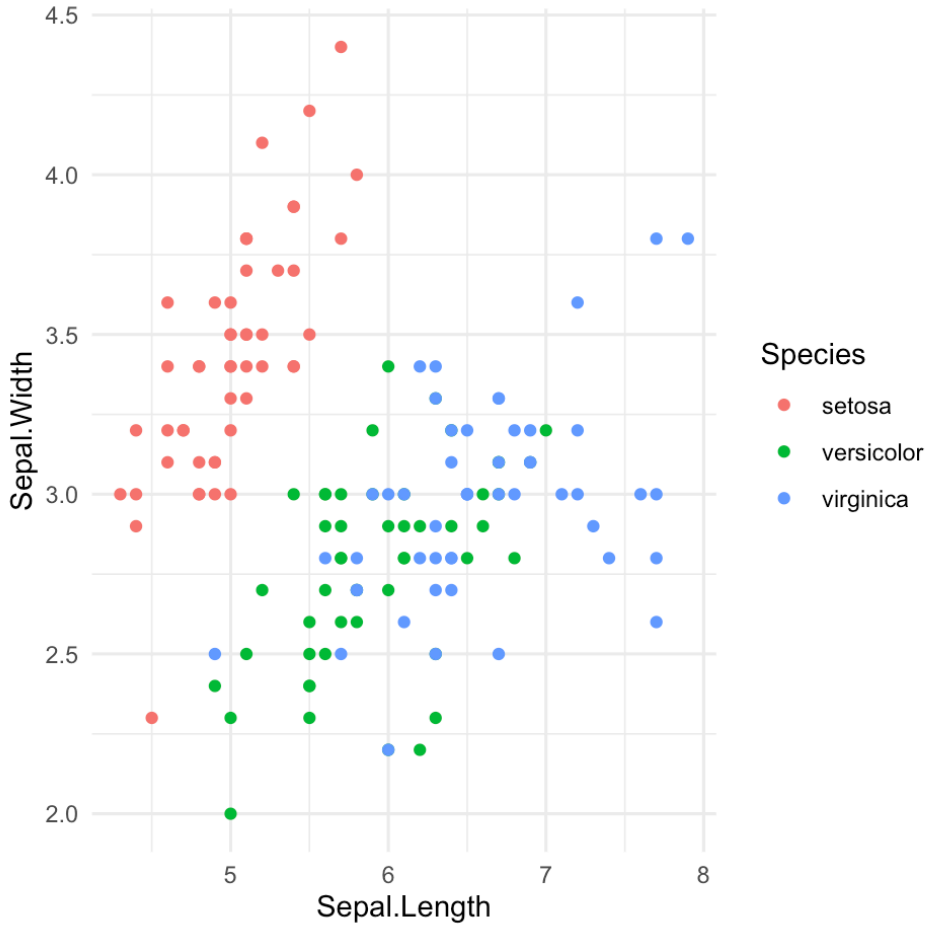
- Baumstruktur zur Entscheidungsfindung
- Klassifikation: Unterschiedliche Algorithmen
  - CART (Classification and Regression Trees)
  - Diskrete Attribute: ID3
  - Kontinuierliche Attribute: C4.5
  - Skalierbar für große Datensätze: z.B. Bagging

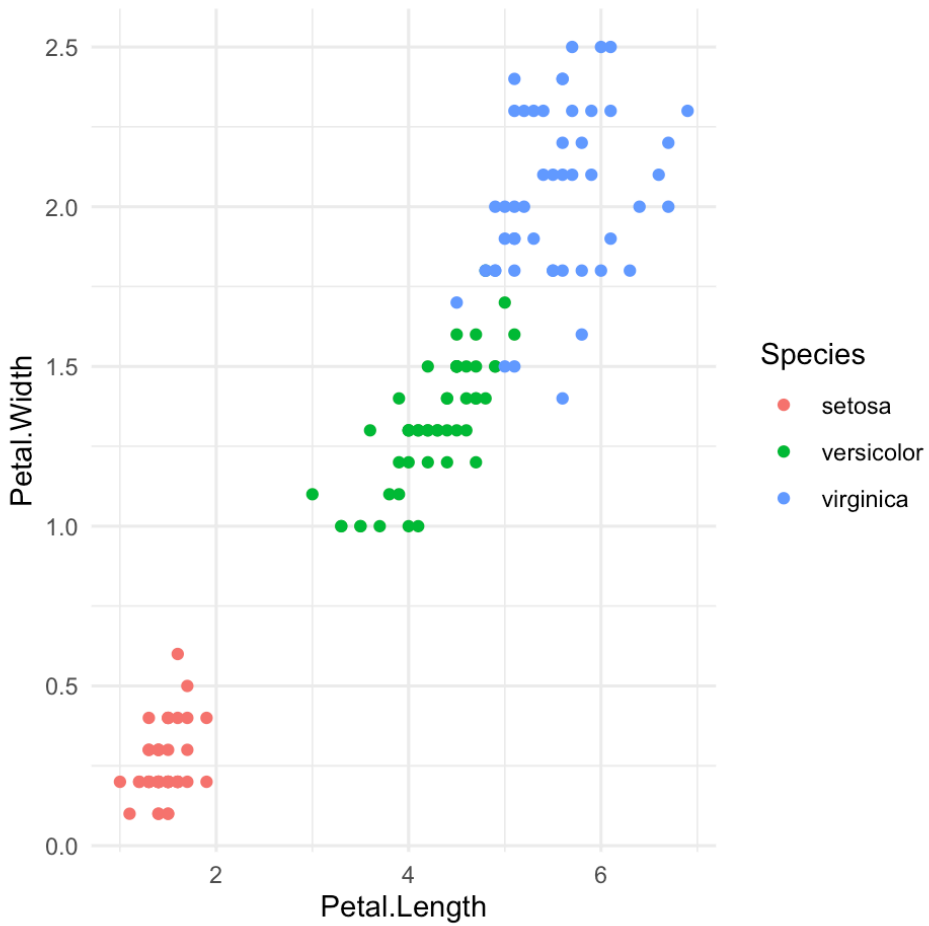
Vor- und Nachteile

- Einfach zu verstehen und zu interpretieren
- Kann sowohl numerische als auch kategoriale Daten behandeln
- Anfällig für Overfitting
- Nicht robust gegen kleine Änderungen in den Daten

```r
# Laden der benötigten Bibliotheken
library(rpart)
library(rpart.plot)

# Beispiel-Datensatz laden
data(iris)

# Entscheidungsbaum-Modell trainieren
tree_model <- rpart(Species ~ ., data = iris, method = "class")

# Entscheidungsbaum visualisieren
rpart.plot(tree_model)
```




4.5
4.0
3.5
3.0
2.5
2.0
Sepal.Width
5
6
7
8
Sepal.Length
Species
setosa
versicolor
virginica


2.5
2.0
1.5
1.0
0.5
0.0
Petal.Width
2
4
6
Petal.Length
Species
setosa
versicolor
virginica

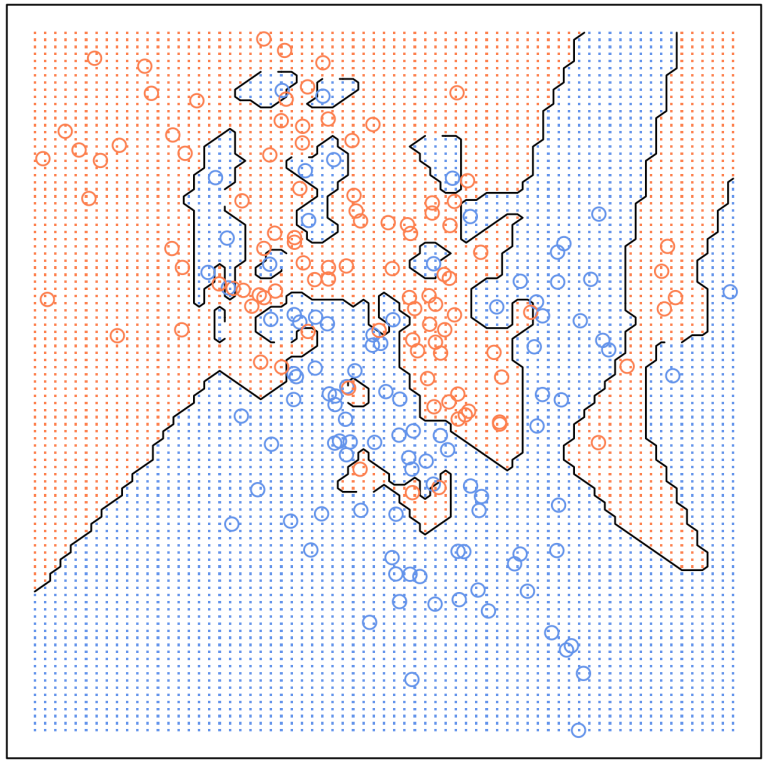
K-Nearest Neighbors (k-NN)

- Nicht-parametrische Methode
- Klassifiziert eine Beobachtung basierend auf den Klassen der k nächsten Nachbarn
- Euklidische Distanz von p zu anderen Punkten q in n Dimensionen:

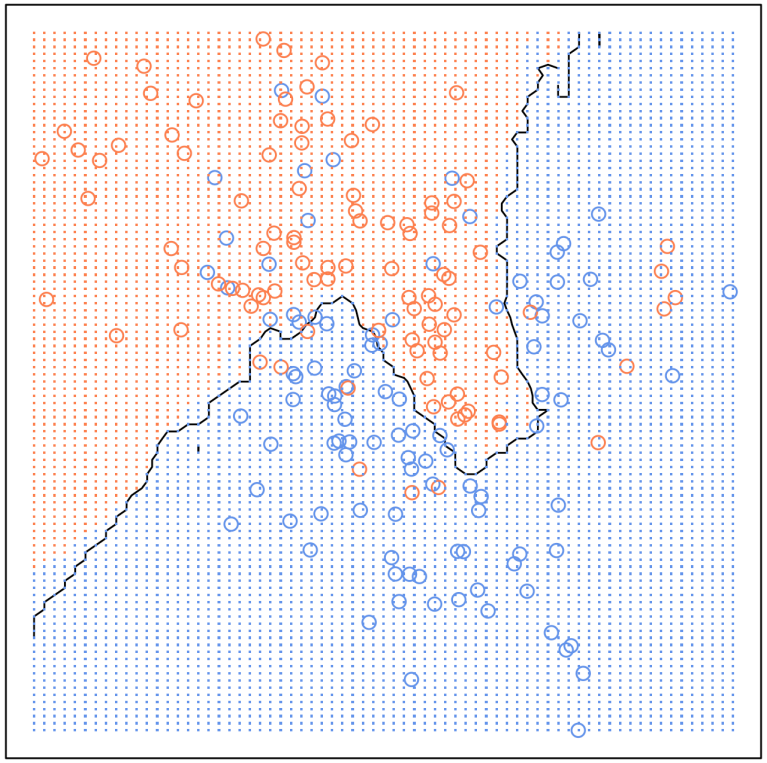
$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

Bias-Varianz Trade-Off

1-nearest neighbours



20-nearest neighbours



Vor- und Nachteile

- Einfach zu implementieren und zu verstehen
- Keine Annahmen über die Datenverteilung
- Rechenintensiv bei großen Datensätzen
- Empfindlich gegenüber irrelevanten oder redundantem Input

Beispiel in R

```r
# Laden der benötigten Bibliotheken
library(class)

# Beispiel-Datensatz laden
data(iris)

# Vorbereitung der Daten
train_indices <- sample(1:nrow(iris), size = 0.7 * nrow(iris))
train_data <- iris[train_indices, ]
test_data <- iris[-train_indices, ]

# k-NN Modell trainieren und Vorhersagen treffen
knn_model <- knn(train = train_data[, -5], test = test_data[, -5], cl = train_data$Species, k = 3)

# Evaluation
table(test_data$Species, knn_model)
```

```
            knn_model
             setosa versicolor virginica
setosa         10          0         0
versicolor      0         15         1
virginica       0          0        19
```

Naive Bayes

- Wahrscheinlichkeitsbasiertes Klassifikationsverfahren
- Annahme: Bedingte Unabhängigkeit der Merkmale

Vor- und Nachteile

- Einfach und schnell zu trainieren
- Gut geeignet für große Datensätze
- Annahme der bedingten Unabhängigkeit ist oft nicht realistisch.
- Performance kann durch Korrelationen zwischen Merkmalen beeinträchtigt werden

Beispiel in R

```r
# Laden der benötigten Bibliotheken
library(e1071)

# Beispiel-Datensatz laden
data(iris)

# Naive Bayes Modell trainieren
nb_model <- naiveBayes(Species ~ ., data = iris)

# Vorhersagen treffen
predictions <- predict(nb_model, iris)

# Evaluation
table(iris$Species, predictions)
```

```
            predictions
             setosa versicolor virginica
setosa         50          0         0
versicolor      0         47         3
virginica       0          3        47
```

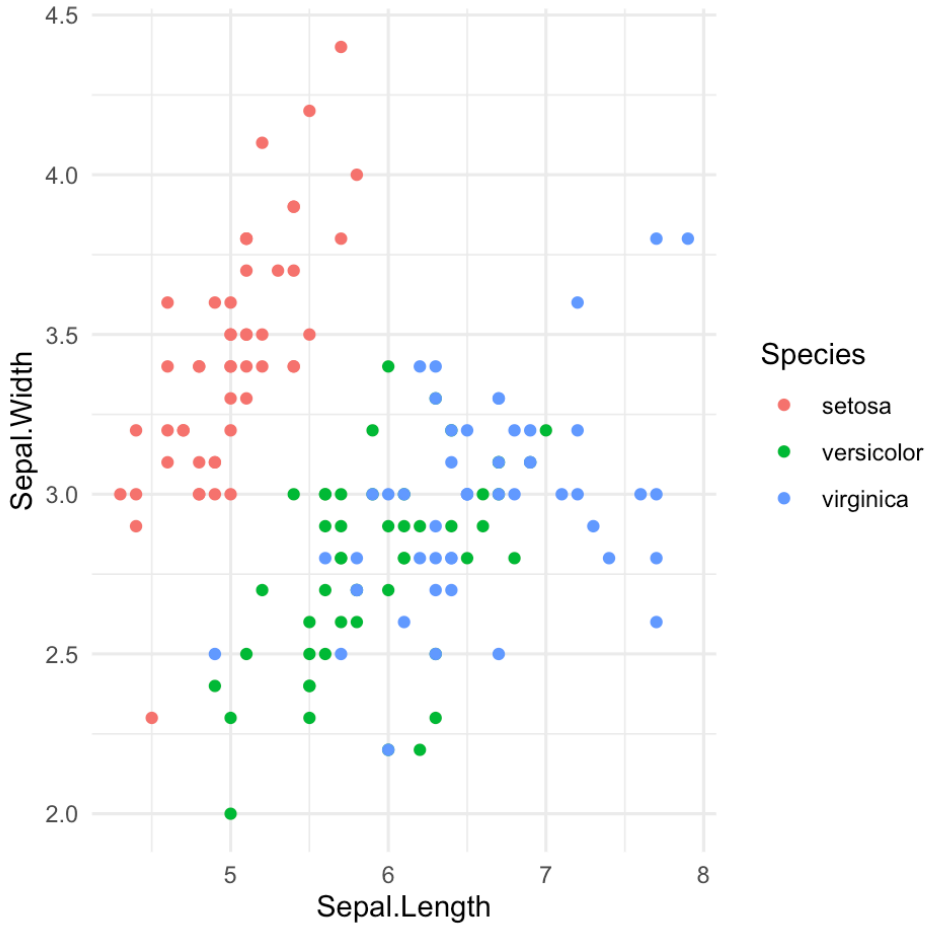
Support Vector Machines (SVM)

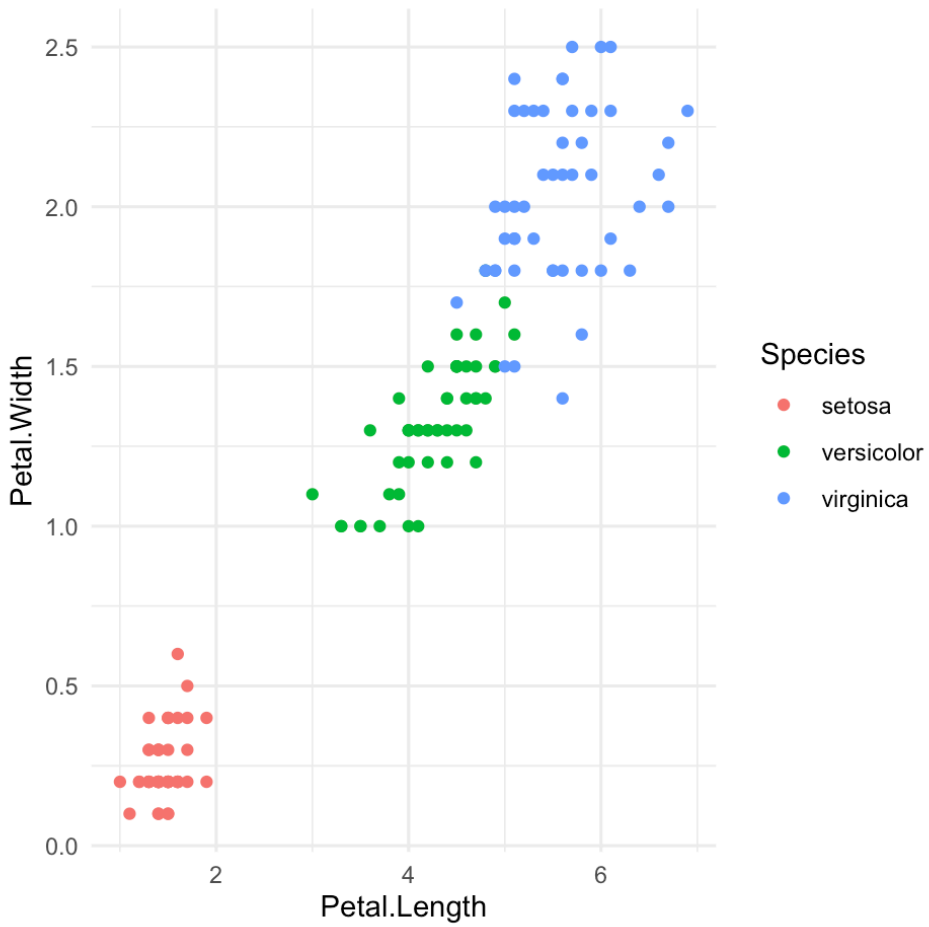
- Klassifikation durch Finden der optimalen Trennlinie (Hyperplane) zwischen Klassen
- Ziel: Maximierung des Abstands (Margin) zwischen den nächsten Punkten (Support Vectors) der Klassen

Vor- und Nachteile

- Effektiv bei hohen Dimensionen
- Robust gegenüber Overfitting, besonders bei richtig gewähltem Kernel
- Rechenintensiv bei großen Datensätzen
- Performance hängt stark von der Wahl des Kernels ab

Idee von SVM


4.5
4.0
3.5
3.0
2.5
2.0
Sepal.Width
5
6
7
8
Sepal.Length
Species
setosa
versicolor
virginica


2.5
2.0
1.5
1.0
0.5
0.0
Petal.Width
2
4
6
Petal.Length
Species
setosa
versicolor
virginica

Beispiel in R

```{r, echo = T # Laden der benötigten Bibliotheken library(e1071)
```

Beispiel-Datensatz laden

data(iris)

SVM Modell trainieren

svm_model <- svm(Species ~ ., data = iris)

Vorhersagen treffen

svm_predictions <- predict(svm_model, iris)

Evaluation

table(iris$Species, svm_predictions)

```
---

## Beispiel in R: Autokauf

![Quelle: Tesla.com](slides7/tesla.png)

---

## Datensatz: autokauf.csv

```{.r}
     nutzer_id geschlecht alter gehalt gekauft
         <int>     <char> <int>  <int>   <int>
  1:  15624510       Male    19  19000       0
  2:  15810944       Male    35  20000       0
  3:  15668575     Female    26  43000       0
  4:  15603246     Female    27  57000       0
  5:  15804002       Male    19  76000       0
 ---
396:  15691863     Female    46  41000       1
```
```

Klassifikation mit Entscheidungsbaum