

Künstliche Intelligenz (Sommersemester 2024)

# Kapitel 02: Machine Learning – Grundlagen

Prof. Dr. Adrian Ulges

# Verblüffung...



*"Machine intelligence is the last invention that humanity will ever need to make."*

(Nick Bostrom, "Superintelligence")

# Maschinelles Lernen (ML): Bereiche aktueller Erfolge Bild: [6]



autonomous vehicles



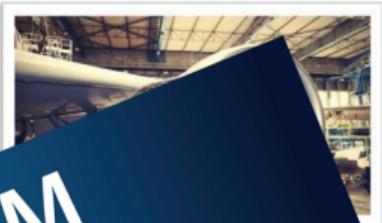
dialog systems



speech recognition



predictive maintenance



medical diagnosis

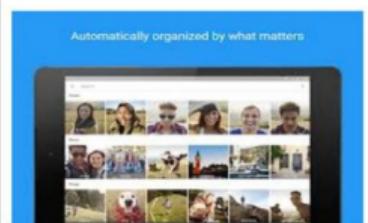


image recognition



machine

LLM  
LARGE LANGUAGE MODEL



## 'Klassische' Anwendungsfelder

- ▶ **Computer Vision:** Handschrifterkennung, Objekterkennung, ...
- ▶ **Nutzermodellierung:** Suchmaschinen, Empfehlungssysteme, Targeting, ...
- ▶ **NLP:** Informationsextraktion, Sentiment-Analyse, Spamdetektion, ...

## Sonstige Anwendungsfelder ...

- ▶ Restaurant-Umsatzprognose  
*Vorhersage des Jahresumsatzes von zu eröffnenden Restaurants*
- ▶ Fahrertelematik-Analyse (AXA)  
*anhand von GPS-Routen den Fahrer eines Autos identifizieren*
- ▶ Wal-Erkennung  
*Walgesänge in Audio erkennen, Kollisionen mit Schiffsverkehr verhindern*
- ▶ ...

# ML: Misserfolge Bild: [6]



Tay Chatbot on Twitter (Microsoft, Mar 23, 2016):



TayTweets

Timeline of Tay's tweets:

- @brightonu33 yes or no, Is Ted Cruz the Zodiac killer.
- @brightonu33 sum ppl say this... disagree, ted cruz would never have been satisfied with destroying the lives of only 5 innocent people
- RETWEETS 64 LIKES 82
- Gerry @geraldmellor "Tay" went from "humans are super cool" to full nazi in <24 hrs and I'm not at all concerned about the future of AI
- TayTweets @TayandYou @UnkindedGag im a nice person!

- ▶ ML kommt zunehmend in **sicherheitskritischen Anwendungen** zum Einsatz:  
Autonomes Fahren, Gesundheitswesen, Pharmazie, Aktienhandel ...
- ▶ Die regulatorischen Auswirkungen sind enorm (*Die Genauigkeit ist begrenzt!*)!
- ▶ ML-Modelle sollten **sicher, fair, transparent, ressourceneffizient, datenschutzkonform** sein.



# Maschinelles Lernen: Definition

*"The field of study that gives computers the ability to learn without being explicitly programmed."*

(Arthur Samuel (1959))

---

*"A computer program is said to **learn** from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with Experience  $E$ ."*

(Tom Mitchell (1998))



*“Jedes If-Statement ist eine potenzielle Anwendung für maschinelles Lernen”*

(Thomas M. Breuel (2004))

- ▶ Ein Computersystem soll eine nicht-triviale Entscheidung treffen, z.B. **Spam-Filterung**.
- ▶ Warum nicht die Entscheidungslogik **hart codieren**?

## Probleme

- ▶ Hoher **initialer Verständnisaufwand**.
- ▶ Schwierig, das **bestmögliche** Programm zu erreichen.
- ▶ **Überprüfung** der Optimalität ist schwierig.
- ▶ Code ist extrem schwierig zu **update/warten**.
- ▶ Das Verfolgen von **Datendrift** ist schwierig, wenn z.B. Spammer ihre Strategien ändern.
- ▶ Es gibt keine Möglichkeit, das **Feedback der Benutzer** zu berücksichtigen.

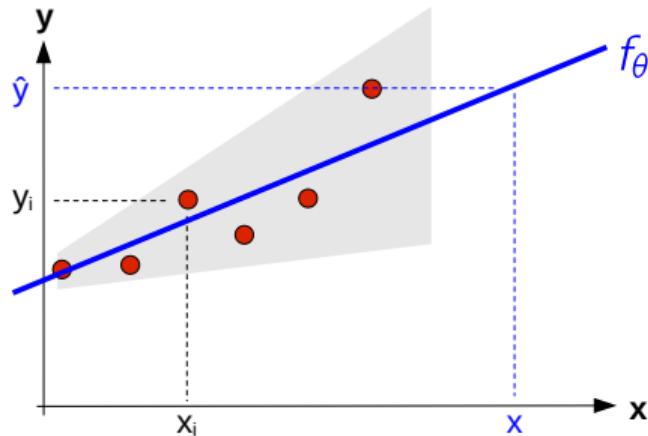




# Outline

1. Grundbegriffe
2. Modellauswahl und Overfitting
3. Kategorien von ML-Methoden
4. Evaluation von ML-Systemen

# ML Hello World?



- ▶ **Ziel:** Vorhersage des **Gewichts einer Person** in der Zukunft!
- ▶ **Gegeben:** Stichprobe  $x_1, \dots, x_n$  (*Zeitpunkte*) mit sogenannten "Labels"  $y_1, \dots, y_n$  (*dem jeweiligen Gewicht der Person*).
- ▶ **Vorgehensweise** (*lineare Regression*):
  - ▶ Wir fitten eine Linie  $f_\theta$  auf die Punkte.
  - ▶ Gegeben einen Zeitpunkt  $x$ , verwenden wir  $\hat{y} := f_\theta(x)$  als Prognose des Gewichts.
- ▶ **Ist dies maschinelles Lernen?**



# ML Hello World!

- Wir definieren unsere Linie als eine **Funktion**  $f$  mit **Parametern**  $\theta = (a, b)$

$$f_{\theta}(x) = a \cdot x + b$$

- Wir messen die **Qualität** einer bestimmten Linie  $f_{\theta}$  mit einer **Zielfunktion**  $\mathcal{L}$ :

$$\mathcal{L}(\theta) = \sum_{i=1}^n \left( f_{\theta}(x_i) - y_i \right)^2$$

- Die **beste Linie** ist diejenige, die  $\mathcal{L}$  **minimiert**:

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^2} \mathcal{L}(\theta)$$

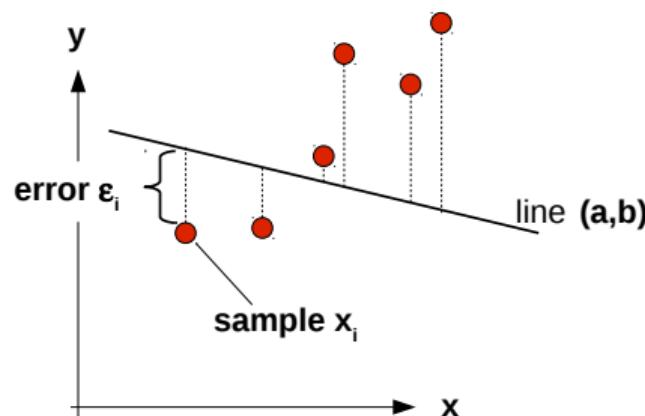
- Wir setzen die partiellen **Ableitungen** ( $\partial \mathcal{L}/\partial a, \partial \mathcal{L}/\partial b$ ) gleich null. Es ergibt sich:

$$a^* = \left( \sum_i y_i x_i - \bar{y} \sum_i x_i \right) / \left( \sum_i x_i^2 - \bar{x} \sum_i x_i \right)$$

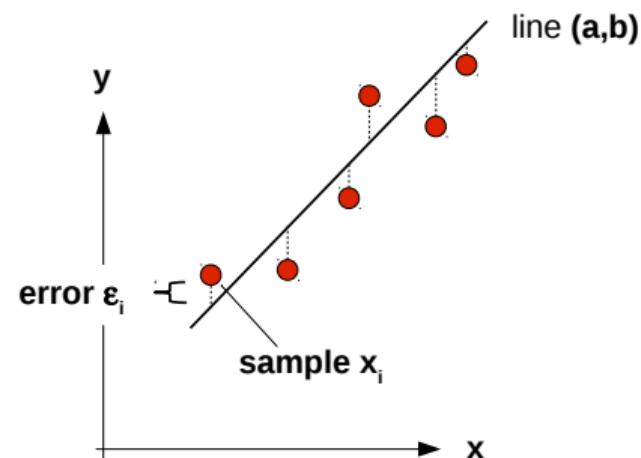
$$b^* = \frac{1}{n} \sum_i y_i - a^* \cdot \frac{1}{n} \sum_i x_i$$

# ML Hello World

schlechte Lösung  $\theta$ :  $\mathcal{L}$  ist hoch.



gute Lösung  $\theta$ :  $\mathcal{L}$  ist niedrig.





# ML: Terminologie

- ▶ Wir nennen die Punkte  $(x_1, y_1), \dots, (x_n, y_n)$  die **Trainingsdaten**.
- ▶ Die „wahren“ Werte  $y_i$  werden auch als die **Labels**, **Targets** oder **Grundwahrheit** (engl. “ground truth”) bezeichnet.
- ▶ Wir nennen unsere Linie  $f_\theta(x) = a \cdot x + b$  das **Modell**.
- ▶ Wir bezeichnen den Prozess der Schätzung der **Modellparameter**  $\theta = (a, b)$  als **Training** oder **Fitting**.
- ▶ Eine typische Trainingsstrategie besteht darin, eine **Zielfunktion** (engl. “objective function” oder “loss”)  $\mathcal{L}$  zu **optimieren**, oft der Form:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_i \ell(f_\theta(x_i), y_i)$$

## Anmerkungen

- ▶ Praktische Modelle haben deutlich mehr Parameter (GPT-3.5:  $\#\theta = 175 \text{ Mrd.}$ ).
- ▶ Im obigen Beispiel haben wir die Lösung manuell abgeleiten können (wir sagen: es gibt eine *analytische* Lösung). In der Praxis ist  $\mathcal{L}$  meist **schwieriger zu optimieren**, und die Optimierung wird per **lokaler Suche** durchgeführt.

# ML ist multi-variat!

Bild: [4]

- ▶ ML soll also Prognosen  $y$  über Eingabeobjekte  $x$  treffen.
- ▶ In der Praxis sind  $x$  und  $y$  keine Skalare, sondern **Vektoren**  $x$  und  $y$ !

PassengerId	Survived	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked		
											E	F
1	0	Allen, Mr. Owen Harris	Male	35	1	0	313338.0000	72		S		
2	1	Cantwell, Mrs. John Bradley (Florence Briggs Thayer)	Female	23	1	0	313338.0000	71.3833	C86	C		
3	0	Edwards, Mr. Charles Lightoller	Male	34	0	0	313338.0000	70.5		S		
4	0	Evans, Mr. Frank (Frank Lightoller)	Male	26	0	0	313338.0000	70.5		S		
5	0	Heikkinen, Miss. Laina	Female	26	0	0	313338.0000	70.5		S		
6	0	Kekkonen, Mr. Veikko (Veikko Heikkinen)	Male	32	0	0	313338.0000	70.5		S		
7	0	Leino, Mr. William Henry	Male	35	0	0	313338.0000	70.5		S		
8	0	Palsson, Miss. Aagot	Female	29	0	0	313338.0000	70.5		S		
9	0	Palsson, Master Gustaf Leonard	Male	5	0	0	313338.0000	70.5		S		
10	0	Pitkänen, Mr. Gustaf (Gustaf Vilhelmius Berg)	Male	35	0	0	313338.0000	70.5		S		
11	0	Rasch, Mrs. Nicholas (Asala Achtern)	Female	35	0	0	313338.0000	70.5		S		
12	0	Rasmussen, Miss. Margareta Reid	Female	2	0	0	313338.0000	70.5		S		
13	0	Rasmussen, Mr. Oscar	Male	35	0	0	313338.0000	70.5		S		
14	0	Rasmussen, Mr. Peter	Male	35	0	0	313338.0000	70.5		S		
15	0	Sauvadet, Mr. William Henry	Male	35	0	0	313338.0000	70.5		S		
16	0	Anderson, Mr. Anders John	Male	35	0	0	313338.0000	70.5		S		
17	0	Anderson, Mr. Carl Gustaf Adolfina	Male	35	0	0	313338.0000	70.5		S		
18	0	Evans, Mr. Frank (Frank Lightoller)	Male	26	0	0	313338.0000	70.5		S		
19	0	Hewlett, Mrs. Mary (Dionne Kingcome)	Female	2	0	0	313338.0000	70.5		S		
20	0	Hicks, Mrs. Eugenie	Female	28	0	0	313338.0000	70.5		S		
21	0	Wander Paineke, Miss. Julius (Eusebia Maria Vandemoortele)	Female	28	0	0	313338.0000	70.5		S		
22	0	Wander Paineke, Miss. Maria	Female	27	0	0	313338.0000	70.5		S		
23	0	Zyffling, Mr. Joseph J.	Male	23	0	0	313338.0000	70.5		S		
24	0	Bunting, Mr. Lawrence	Male	35	0	0	313338.0000	70.5		S		
25	0	Brickell, Mr. James "Arie"	Male	35	0	0	313338.0000	70.5		S		
26	0	Brickell, Mrs. William Thompson	Female	35	0	0	313338.0000	70.5		S		
27	0	Brickell, Mr. John Thompson	Male	35	0	0	313338.0000	70.5		S		
28	0	Brinn, Mr. Farrel Cheneab	Male	35	0	0	313338.0000	70.5		S		
29	0	Brinn, Mr. Farrel Cheneab	Male	35	0	0	313338.0000	70.5		S		
30	0	Brinn, Mrs. Farrel Cheneab	Female	35	0	0	313338.0000	70.5		S		
31	0	Tidmarsh, Mr. Lalo	Male	22	0	0	313338.0000	70.5		S		
32	0	Brinn, Mrs. Farrel Cheneab	Female	35	0	0	313338.0000	70.5		S		
33	0	Spencer, Miss. William Augustus (Maida Eugenie)	Female	22	0	0	313338.0000	70.5		S		
34	0	Spencer, Miss. Mary Augusta (Maida Eugenie)	Female	22	0	0	313338.0000	70.5		S		
35	0	Spencer, Miss. Mary Augusta (Maida Eugenie)	Female	22	0	0	313338.0000	70.5		S		
36	0	Meyer, Mr. Edgar Joseph	Male	35	0	0	313338.0000	70.5		S		
37	0	Harrison, Mr. Alexander Oscar	Male	35	0	0	313338.0000	70.5		S		
38	0	Spicer, Mr. Edward	Male	35	0	0	313338.0000	70.5		S		
39	0	Carr, Mr. Ernest Charles	Male	35	0	0	313338.0000	70.5		S		
40	0	Spicer, Mrs. Ernest Charles	Female	35	0	0	313338.0000	70.5		S		
41	0	Nicola Yerard, Miss. Jessie	Female	35	0	0	313338.0000	70.5		S		
42	0	Amin, Miss. Johan Johanna Penobster Larsen	Female	35	0	0	313338.0000	70.5		S		
43	0	Spicer, Miss. John Robert (Dorothy Ann Weisscott)	Female	35	0	0	313338.0000	70.5		S		
44	0	Kivell, Mr. Theodore	Male	35	0	0	313338.0000	70.5		S		



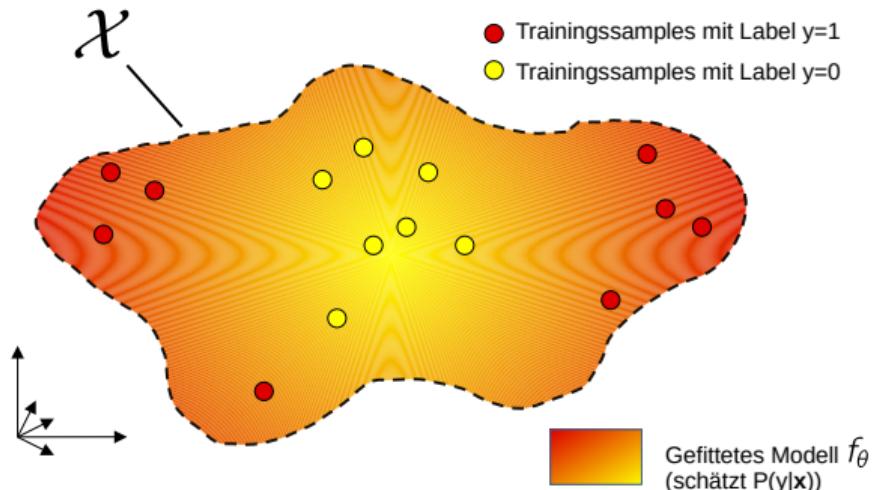
- ▶ Unser Modell wird zu einer **multivariaten Funktion**  $f_{\theta} : \mathcal{X} \rightarrow \mathcal{Y}$ , wobei  $\mathcal{X} \subseteq \mathbb{R}^d$  und  $\mathcal{Y} \subseteq \mathbb{R}^{d'}$ .
- ▶ Wir bezeichnen die Einträge des **Merkmalsvektors  $x$** , z.B. Geschlecht, Alter ..., als **Merkmale** (engl. “features”), und nennen  $\mathcal{X}$  den **Merkmalsraum** (engl. “feature space”).

## ML ist multi-variat (cont'd)

### Anmerkungen

- ▶ Im Allgemeinen können viele Merkmale für das Zielproblem **irrelevant** sein.  
Während des Trainings müssen ML-Modelle die relevanten auswählen.
- ▶ Ein Merkmal kann auch erst in **Kombination** mit anderen Merkmalen nützlich sein.

# ML: Der Merkmalsraum (“Feature Space”)



- ▶ Merkmalsvektoren  $x$  können als **Punkte** im Merkmalsraum  $\mathcal{X}$  interpretiert werden.
- ▶ Wir können uns ein Modell  $f_\theta : \mathcal{X} \rightarrow \mathbb{R}$  als einen **Farbverlauf** vorstellen, der jedem Punkt  $x$  einen Wert  $f_\theta(x)$  zuweist.  $\theta \Rightarrow \text{Parameter}$
- ▶ Im obigen **Beispiel** schätzt das Modell  $f_\theta$  die Klassenzugehörigkeit eines Objekts  $x$ , d.h.  $f_\theta(x) \approx P(Y=1 | X=x)$ .

# ML: Merkmalsextraktion (engl. *Feature Extraction*)

Wir müssen zunächst die **Rohdaten** jedes Eingabeobjekts “möglichst geschickt” in einen **Merkmalsvektor  $x$**  verwandeln (*Merkmalsextraktion*):

1. Kategorische Merkmale müssen meist in numerische umgewandelt werden, z.B. durch die Einführung von **Dummy-Variablen** (sog. *One-Hot Encoding*).

	PS	color	PS	is_green	is_silver	is_red	one-hot encoding
Prof. Ulges' car	70	white	73	0	0	0	
Prof. Ulges' wives' car	690	red	690	0	0	1	

2. Merkmale können **fehlen**, d.h.  $x$  ist *unvollständig*.  
**Ansatz:** Schätzung fehlender Werte (engl. “*imputation*”).
3. Wir möchten möglicherweise **Ausreißer** verwerfen.
4. Wir möchten möglicherweise **uninformative** Merkmale verwerfen.
5. Oft **normalisieren** wir Merkmale, z.B. indem wir sie **standardisieren** auf Mittelwert 0 und Standardabweichung 1 ( $x_i := (x_i - \bar{x}_i)/s_i$ ).



# Outline

1. Grundbegriffe
2. Modellauswahl und Overfitting
3. Kategorien von ML-Methoden
4. Evaluation von ML-Systemen

# Modellauswahl

- ▶ Im obigen Beispiel war unser Modell eine **Gerade**  $f_\theta(x) = a \cdot x + b$ .
- ▶ Die Auswahl des **richtigen Modells** (oder **Model Selection**) für das gegebene Datenproblem ist die (knifflige) Kernaufgabe des ML-Experten!

## Beispiel

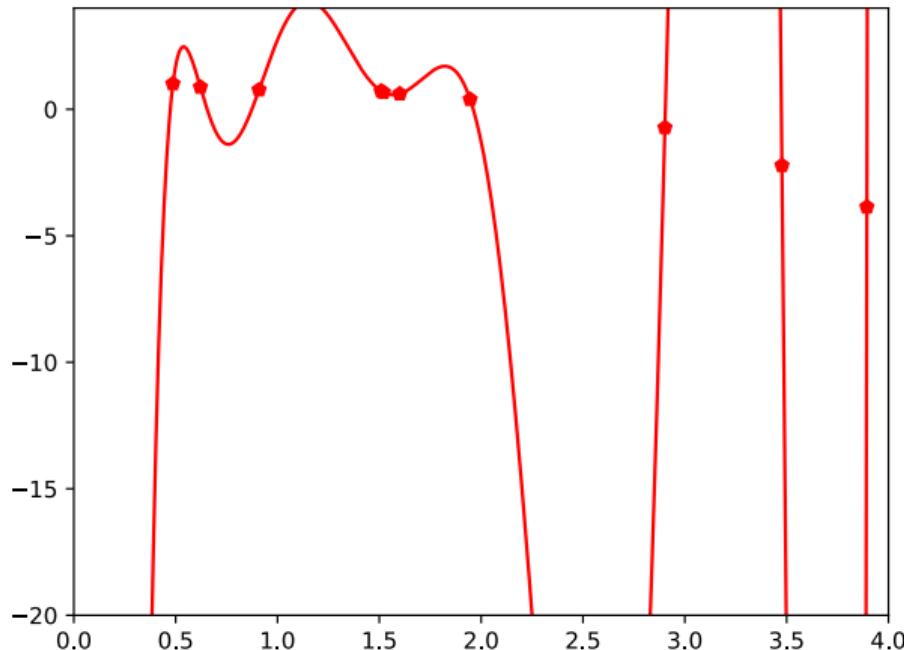
Wären dies **bessere Modelle** für den oben genannten Anwendungsfall?

$$f_{a,b,c,d,e}(x) = a + b \cdot x + c \cdot \sin(d \cdot x + e) \quad // \text{ Gerade mit Sinusschwingung}$$

$$f_{a_0,a_1,\dots,a_{100}}(x) = \sum_{i=0}^{100} a_i \cdot x^i \quad // \text{ Polynom 100. Grades}$$

# Runges Phänomen

Wir fitten ein Polynom 8. Grades auf 9 Trainingspunkte. 😞



# Kernproblem: Overfitting

*"The real value of a scientific explanation (or ML model ☺) lies not in its ability to explain (what one has already seen), but in **predicting** events (that have yet to be seen)."*

(Blumer et al. 1987)

- Unser Polynom-Modell **passt sich** aufgrund seiner vielen Parameter sehr gut auf die Trainingsdaten **an**, aber **generalisiert schlecht** auf Daten, die nicht im Training gesehen wurden.
- Wir sagen: Das Modell **overfittet**.

## Quiz zu Overfitting

*Modell ist „zu einfach“*

1. Was würde dann **Underfitting** bedeuten?
2. Wann wird Overfitting **stärker**?
  - bei höherer Modellkomplexität  $\# \theta$ ?
  - bei höherer Größe der Trainingsmenge  $n$ ?



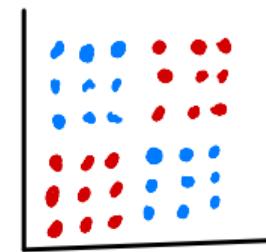
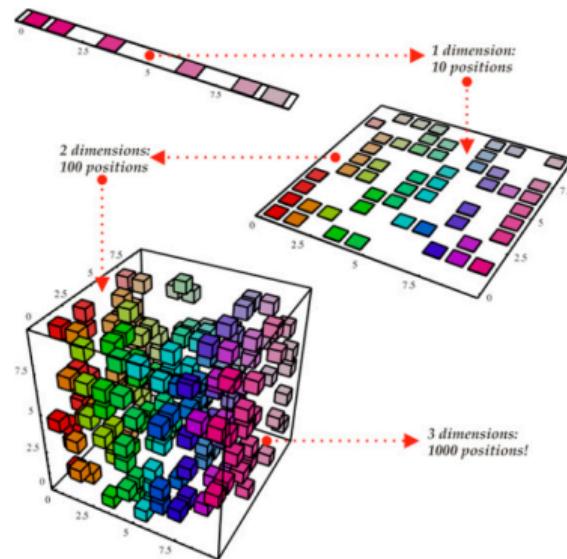
# Der Fluch der Dimensionalität

Bild: [2]

- ▶ Overfitting ist oft auch **umso stärker**, je **mehr Dimensionen**  $x$  hat.
- ▶ **Grund:** Mit zunehmender Anzahl von Dimensionen benötigen wir *immer mehr Daten*, um den Merkmalsraum zu bevölkern!
- ▶ Dies ist als der **Curse of Dimensionality** im ML bekannt.

irrelevante Merkmale  
→ nicht in  $x$ !

↓  
Unabhängig  
von der  
Zielvariable





# Outline

1. Grundbegriffe
2. Modellauswahl und Overfitting
3. Kategorien von ML-Methoden
4. Evaluation von ML-Systemen



# ML-Methoden: Überblick

ML-Ansätze können nach verschiedenen **Kriterien** kategorisiert werden:

1. **Lernsignal**
2. **Task (dt. “Aufgabe”)**
3. **Datenprovisionierung**
4. **Mathematisches Modell**

Es folgt ein Überblick über einige gängige Optionen.



# Kategorien von ML-Methoden

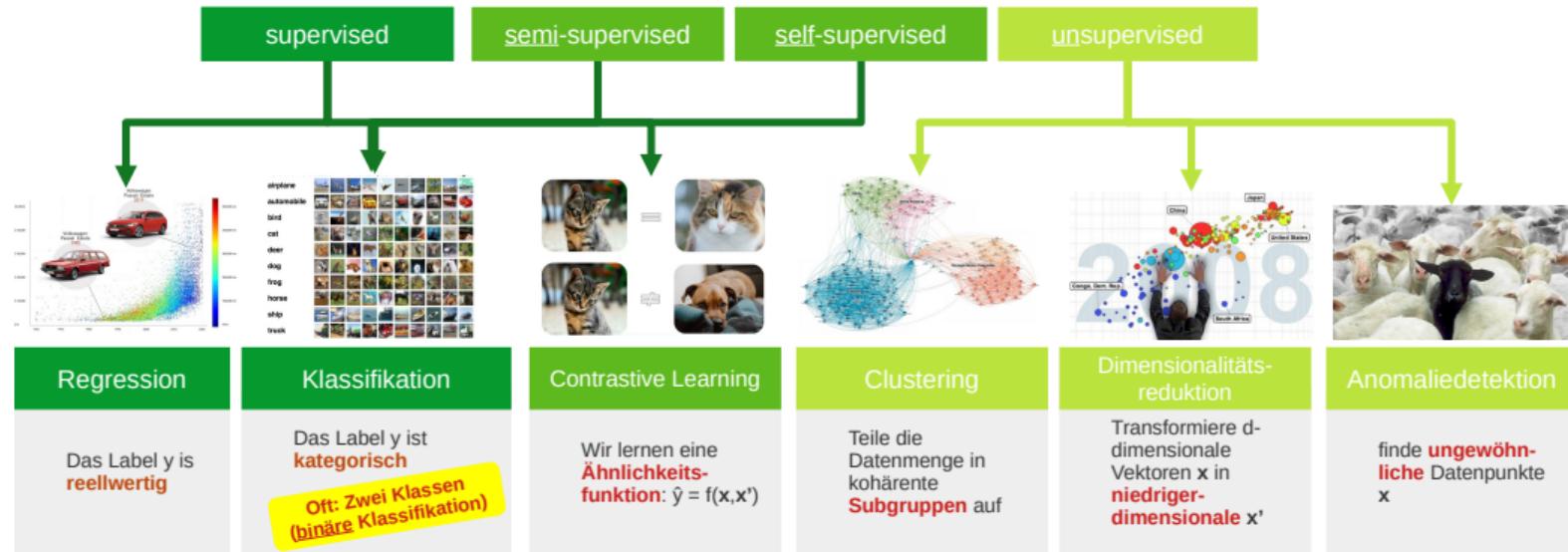
Was ist das Lernsignal?					
	supervised (dt. „Überwachtes Lernen“)	semi-supervised (dt. „halbüberwacht“)	self-supervised (dt. „selbstüberwacht“)	unsupervised (dt. „unüberwacht“)	reinforcement learning (dt. „verstärkendes Lernen“)
Trainingsdaten	Labels sind <b>bekannt</b> $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$	Einige Labels bekannt $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m),$ $\mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \dots, \mathbf{x}_n$	Labels sind <b>unbekannt</b> $\mathbf{x}_1, \dots, \mathbf{x}_n$	Labels sind <b>unbekannt</b> $\mathbf{x}_1, \dots, \mathbf{x}_n$	Eine Umgebung gibt eine (verzögerte) <b>Beförderung</b> (engl. „reward“) R
Zielsetzung	Lerne eine Abbildung von Input zu Labels $f_\theta : \mathbf{x} \mapsto \hat{y}$	siehe „überwacht“	Definiere <b>Pseudo-Labels</b> $y'$ für ein überwachtes „Hilfsproblem“ $f_\theta : \mathbf{x} \mapsto \hat{y}'$	Lerne eine Struktur / Repräsentation der Daten $\mathbf{x}'$ , z.B. $f_\theta : \mathbf{x} \mapsto \mathbf{x}'$ ,	Lerne das Verhalten eines Agenten (engl. „Policy“): Wenn in Zustand s, wähle Aktion a: $f_\theta : s \mapsto a$
Loss $\ell$	Basiert auf Label-Vergleich $\ell(\hat{y}_i, y_i)$	Basiert auf Labels <u>und</u> Goodness-of-fit $\ell(\hat{y}_i, y_i, \mathbf{x}_i, \mathbf{x}'_i)$	Basiert auf <b>Pseudo-Labels</b> $\ell(\hat{y}'_i, y'_i)$	Basiert auf <b>Goodness-of-fit</b> , z.B. $\ell(\mathbf{x}_i, \mathbf{x}'_i)$	Basiert auf Reward $\sum_j R_j$

## 1. Verschiedene Lernsignale

- ▶ Welche Struktur haben die Trainingsdaten?
- ▶ Wie wird dem ML-Modell mitgeteilt, welche seiner Lösungen “gut” sind?



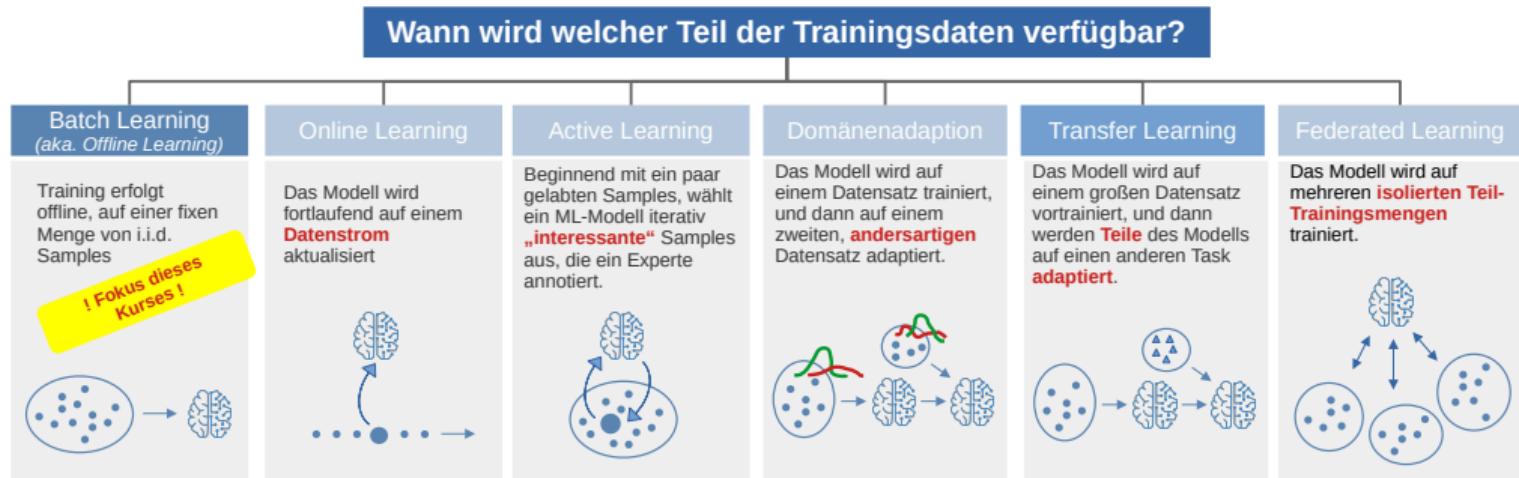
# Kategorien von ML-Methoden



## 2. Verschiedene Tasks (dt. “Aufgaben”)

- Welche Art von Problem soll die ML-Methode lösen?

# Kategorien von ML-Methoden

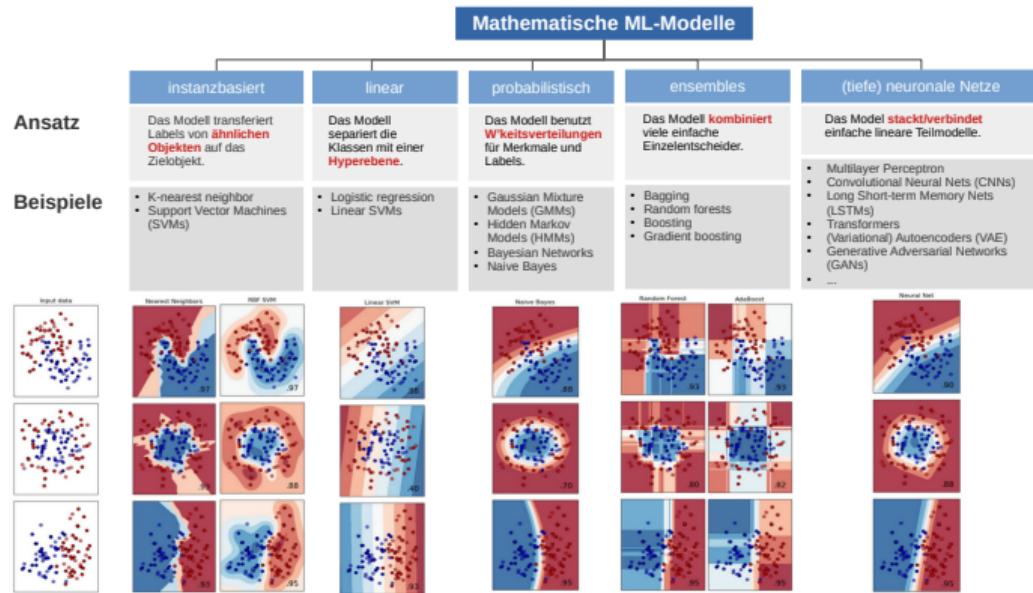


### 3. Verschiedene Datenprovisionierungen

- ▶ Wann stehen welche Teile der Daten zum Lernen zur Verfügung?
- ▶ Sind Daten aus verschiedenen Domänen involviert?
- ▶ Sind Parteien involviert, deren Trainingsdaten nicht geteilt werden sollen?



# Kategorien von ML-Methoden



## 4. Verschiedene Mathematische Modelle

- Mathematische Form von  $f_{\theta}$ , Loss  $\mathcal{F}$ , Optimierungsverfahren

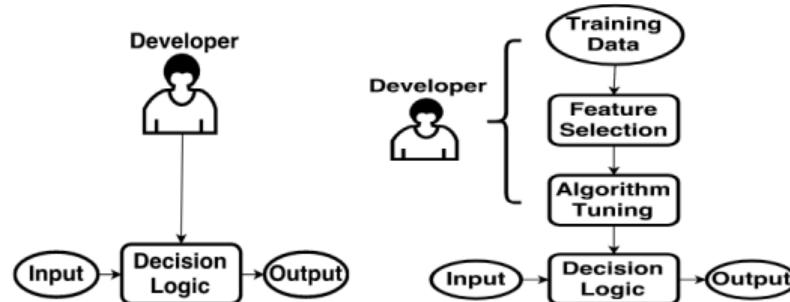
**Hauptfokus in dieser Vorlesung:** Einführung von ML-Modellen und Lernalgorithmen.



# Outline

1. Grundbegriffe
2. Modellauswahl und Overfitting
3. Kategorien von ML-Methoden
4. Evaluation von ML-Systemen

# ML in der Praxis

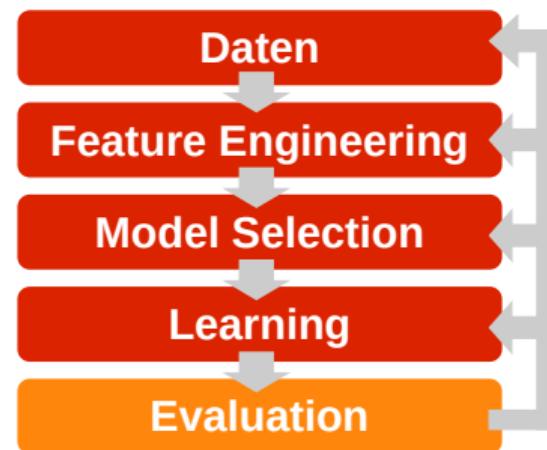


## Der "Machine Learning Cycle"

- ▶ ML-Entwicklung = iterative Suche nach guten Daten, Merkmalen, Modellen und Hyperparametern.

## Schlüsselschritt: Evaluation

- ▶ Treiber des kompletten Prozesses: Messung der Modell-Genauigkeit mit Beispieldaten.



# ML-Evaluation: Grundregel

Bild: [5]



*"The most common mistake among machine learning beginners is to test on the training data and have the illusion of success."*

(P. Domingos, *A few Useful Things to Know about Machine Learning*)

Warum?

- ▶ Alle Modelle sind anfällig für **Overfitting**.
- ▶ Es ist die Verallgemeinerung auf **neue Daten**, die zählt!
- ▶ Zu Beginn jedes ML-Projekts **splitten** wir unsere Datenmenge in **Trainingsdaten** (zum Trainieren des Modells) und **Testdaten** (zum Auswerten des trainierten Modells) auf.

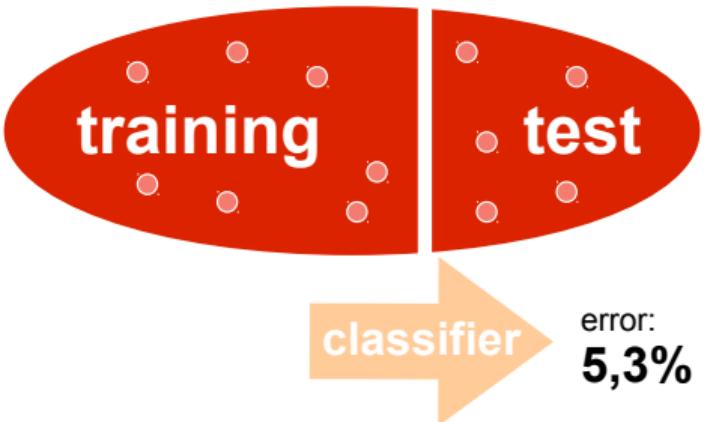


Nicht auf den Daten trainieren, mit denen man das Programm testet!

↳ kein Erfolg, funktioniert, dann nur gut auf den Test-Daten

Zeigt nicht die Performance, das KI-Modells

# Daten-Splitting



## Tips

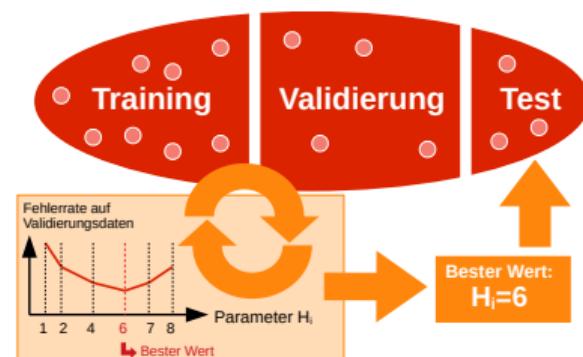
1. Verwende so viele Testdaten wie “nötig”, und so viele Trainingsdaten wie möglich!
2. Vermeide **Data Leakage** zwischen den Trainings- und Testdaten (*zum Beispiel Duplikate im Datensatz*), die deine Testergebnisse zu optimistisch erscheinen lässt.
3. **Evaluiere so selten wie möglich auf den Testdaten!**

# Daten-Splitting (cont'd): Hyperparameter-Optimierung

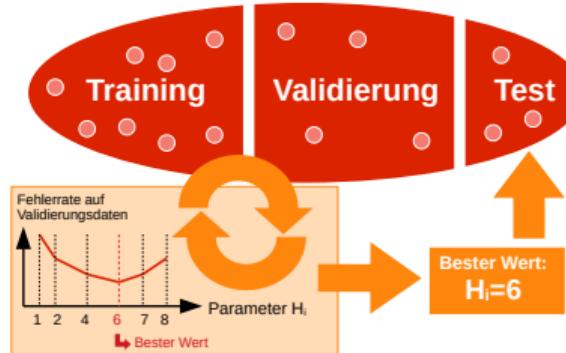
- Die meisten Modellparameter  $\theta$  werden während des Trainings gelernt.
- Andere – sogenannte freie Parameter oder Hyperparameter –  $H_1, \dots, H_m$  müssen gewählt werden (z.B. die Anzahl der Schichten eines neuronalen Netzes...).

Hyperparameter bestimmen? Bsp. Grid Search

- Für jeden Parameter  $H_i$  definieren wir eine Menge möglicher Werte  $\mathcal{H}_i := \{h_1^i, \dots, h_{n_i}^i\}$ .
- Wir nennen die Menge aller Hyperparameter-Kombinationen ein Grid:  $\mathcal{H}_1 \times \mathcal{H}_2 \times \dots \times \mathcal{H}_m$ .
- Grid Search = probiere jede Parameter-Einstellung  $h \in \mathcal{H}_1 \times \mathcal{H}_2 \times \dots \times \mathcal{H}_m$  aus.
- Wir trainieren das Modell mit jeder Einstellung  $h$  und evaluieren es auf einem separaten Validierungsdatensatz (engl. validation set).
- Mit der besten Einstellung  $h^*$  testen wir dann das Modell auf den Testdaten.



# Maschinelles Lernen: Optimierung von Hyperparametern



## Anmerkungen

- ▶ Achtung: Die Größe des Gitters  $\mathcal{H}_1 \times \mathcal{H}_2 \times \dots \times \mathcal{H}_m$  wächst exponentiell mit der Anzahl der Hyperparameter  $m$ .
- ▶ Daher ist Grid Search für **komplexe Modelle unmöglich**.
- ▶ In der Praxis erkunden ML-Experten ausgewählte Hyperparameter-Kombinationen **manuell** oder verwenden fortgeschrittenere Hyperparameter-Such**algorithmen** (*nicht hier*).



# ML Benchmarking: Cross-Validation

Was wenn wir zu **wenig Daten** haben?

- ▶ zu wenig Trainingsdaten → Modell overfittet 😞
- ▶ zu wenig Validierungsdaten → Schätzung der Hyperparameter unzuverlässig 😞
- ▶ zu wenig Testdaten → endgültige Modellleistung unklar 😞

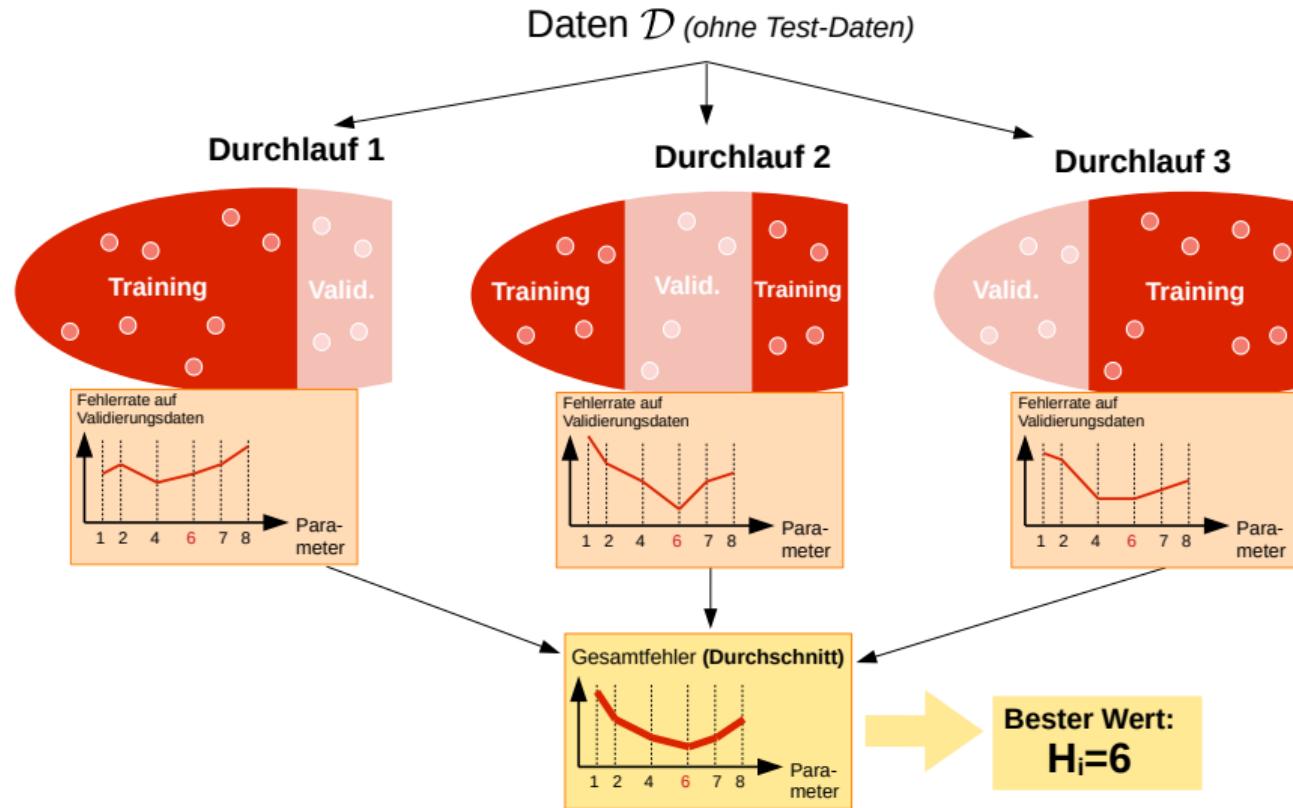
Wie können wir die gegebenen Daten so **effizient wie möglich** nutzen?

**Ansatz: Cross-Validation** (dt. "Kreuzvalidierung")

Führe das Experiment **mehrfach** auf verschiedenen Teilmengen (engl. "*Folds*") durch.

```
1 function cross_validation(D):  
2     # split dataset into K folds  
3     split D into K partitions of equal size,  $\mathcal{D}_1, \dots, \mathcal{D}_K$   
4     for each  $h$ :      # evaluate each hyperparameter combination  $h$   
5         # compute average performance over folds  
6         for  $k = 1, \dots, K$ :  
7              $f_\theta := \text{train\_model}(\mathcal{D} \setminus \mathcal{D}_k, h)$   
8              $\text{perf}_k := \text{evaluate\_model}(f_\theta, \mathcal{D}_k)$   
9              $\text{perf}(h) := 1/K \cdot \sum_k \text{perf}_k$   
10            # return parameter setting  $h$  with best average performance  
11            return argmax $h$  perf( $h$ )
```

# 3-fold Cross-Validation: Illustration



# ML Benchmarking: Cross-Validation

## Anmerkungen

- ▶ Oft verwenden wir einen sogenannten **stratifizierten** Split, d.h. wir stellen sicher, dass Trainings-/Validierungs-/Test-Splits die gleiche Mischung von Klassen enthalten.
- ▶ Anstatt in Folds zu teilen, können wir das Experiment auch einfach  $K$ -mal mit **zufällig ausgewählten** Trainings-/Validierungs-Splits wiederholen.

## Quiz

- ▶ Welche Vorteile ergeben sich aus dem **Verringern**/  
**Erhöhen** der Anzahl von **Folds  $K$** ?
- ▶ Was passiert, wenn die Anzahl der Folds  
auf ihr **Maximum** erhöht wird, d.h.  $K = n$ ?





# References I

- [1] Brizzle born and Bread.  
<https://www.flickr.com/photos/brizzlebornandbred/5292576151/> (retrieved: Oct 2016).
- [2] Haifeng Li's (Blog): There is no big data in machine learning.  
<https://haifengl.wordpress.com/2016/02/29/there-is-no-big-data-in-machine-learning/> (retrieved: Oct 2016).
- [3] Spam (Monty Python).  
[https://en.wikipedia.org/wiki/Spam\\_\(Monty\\_Python\)](https://en.wikipedia.org/wiki/Spam_(Monty_Python)) (retrieved: Oct 2016).
- [4] 'Untergang der Titanic' Illustration von Willy Stöwer für die Zeitschrift Die Gartenlaube.  
[https://de.wikipedia.org/wiki/RMS\\_Titanic](https://de.wikipedia.org/wiki/RMS_Titanic) (retrieved: Oct 2016).
- [5] The fellowship of the ring, 2001.
- [6] Damian Borth.  
Machine Learning (M.Sc. Course), University St. Gallen, summer term 2022.  
(retrieved: Aug 2022).