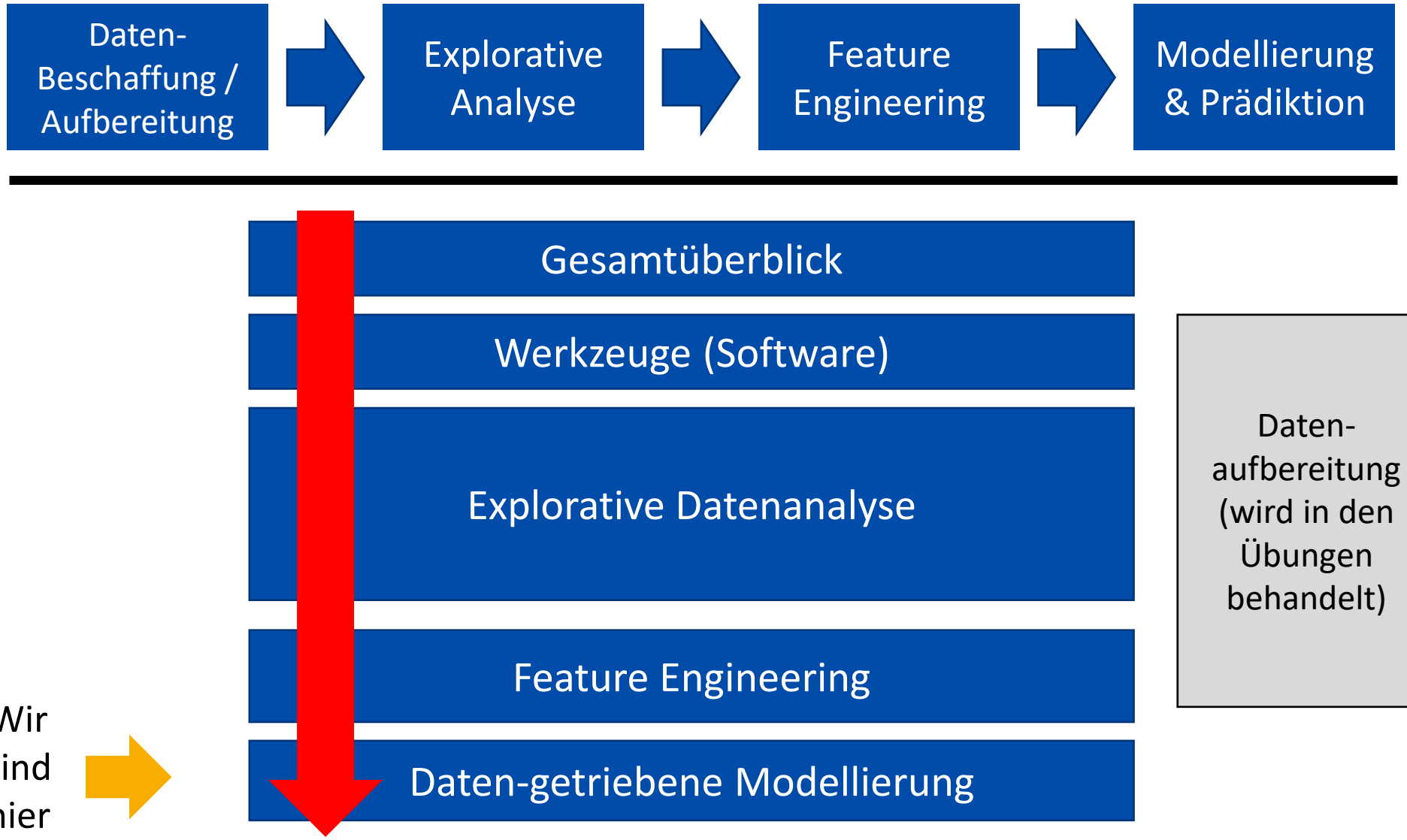
Einführung in Data Science

Unser Plan für heute:

- Daten-getriebene Modellierung
  1. Grundbegriffe aus dem Machine Learning
  2. Training- und Testdaten
  3. kNN Modelle

- Besprechung Evaluation

Data Science



Curriculum

1. Grundbegriffe / Überblick
2. Zentrale Softwarebibliotheken

3. Univariate explorative Analyse (EDA) Visualisierung (Teil 1)
4. Visualisierung (Teil 2), Multivariate explorative Analyse (Teil 1)
5. Multivariate explorative Analyse (Teil 2)
6. Dimensionsreduktion (Teil 1): PCA
7. Dimensionsreduktion (Teil 2): MDS, Isomap
8. Clustering: K-Means, HCA
9. Clustervalidierung

10. Probeklausur

11. Feature Engineering, Datengetriebene Modellierung (Teil 1)
12. Datengetriebene Modellierung (Teil 2)

Überblick / Begriffe

Explorative Analyse (EDA)

Feature Engineering & Modellierung

Daten-getriebene Modellierung | Klassifikation

Bisher: binäre Klassifikation

- Modell unterscheidet zwischen *zwei* Klassen
- einfachstes Modell: Schwellwert-basierte Klassifikation

Multiklassen Klassifikation

- Modell unterscheidet zwischen mehr als zwei Klassen
- einfachstes Modell: Nächste Nachbarn

Nächste Nachbarn Modell (NN Modell)

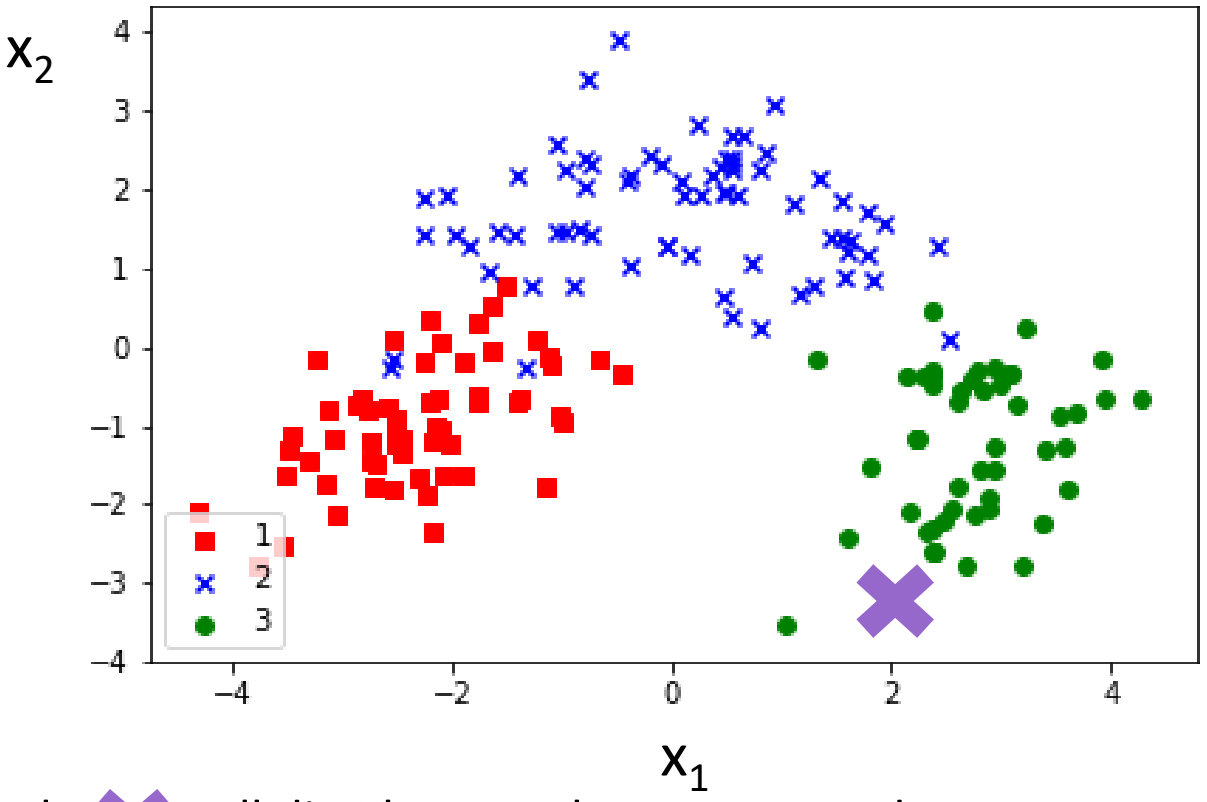
- Nähe (bzw. Distanz) im Merkmalsraum bestimmt vorherzusagende Klasse.

Daten-getriebene Modellierung | NN Modell

Wie lässt sich eine Multiklassen-Klassifikator erstellen?

Beispiel:

- Daten mit drei Klassen
- Generieren eines Nächste Nachbarn Modells (kurz: NN Modell)



Für einen *neuen* Datenpunkt ✖ soll die Klasse vorhergesagt werden.

- Gebe als Vorhersage die Klasse des nächsten Nachbarn von ✖ zurück. (Nähe wird hier durch euklidische Distanz definiert.)

Beurteilung eines Klassifikators | Grundkonzepte

Multiklassen-Klassifkation

→ Vorhersagen über mehr als zwei Klassen

Beispiel: Detektion der drei Klassen „Covid-19“, „Influenza“, „Nicht erkrankt“

Verschiedene Ansätze:

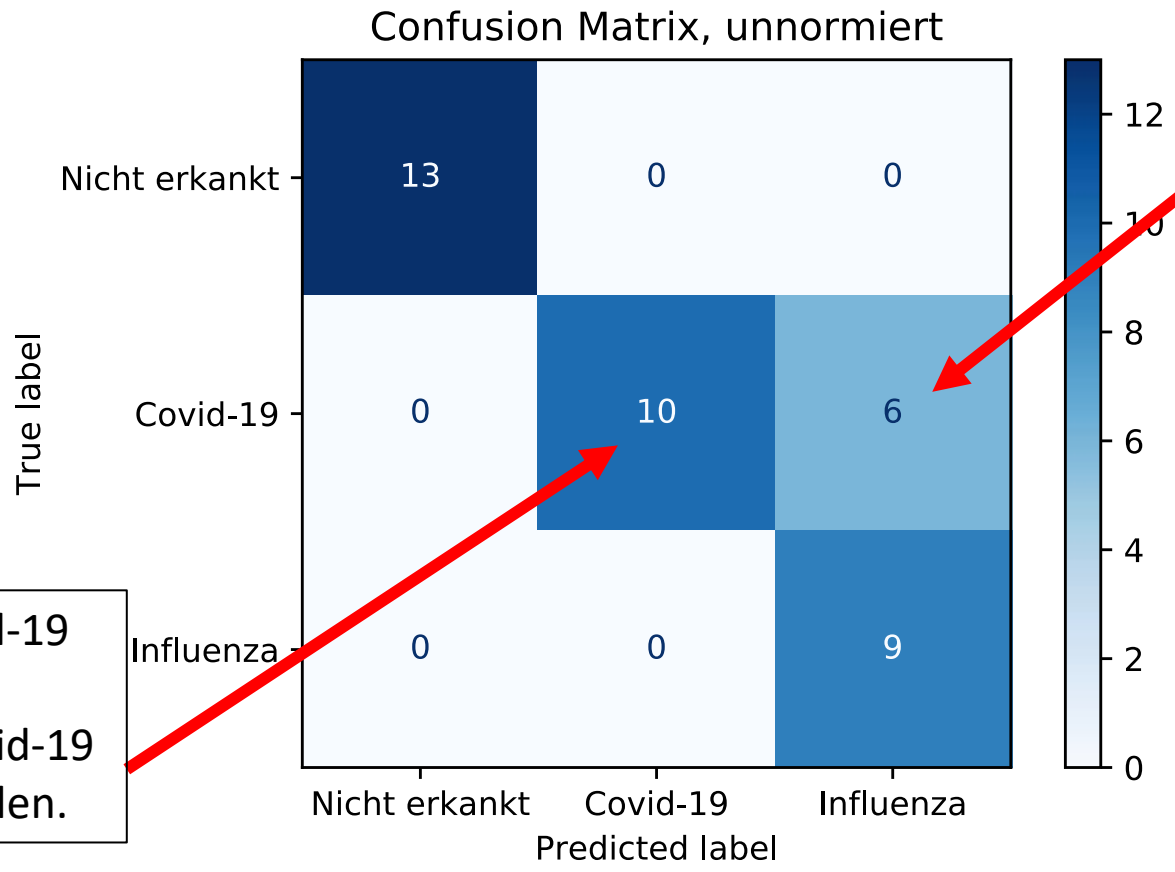
a) Top-K Accuracy (oft: K=5)
Vorhersage wird als True Positive *TP* gewertet, sofern die wahre Klasse unter den Top-K der wahrscheinlichsten Klasse liegt.
(Grenzfall K=1 entspricht der normalen *Accuracy*)

b) One-vs-All Ansatz
Ermitteln des F1-Scores für jede Klasse (z.B. Covid-19 vs „Nicht-covid-19“
Mittelwert über die F1-Scores aller Klassen bilden.

c) Confusion Matrix (Wahrheitsmatrix) → auf der nächsten Folie

Beurteilung eines Klassifikators | Confusion Matrix

Confusion Matrix C_{ij} : Anzahl Elemente der Klasse i die als Klasse j vorhergesagt werden.

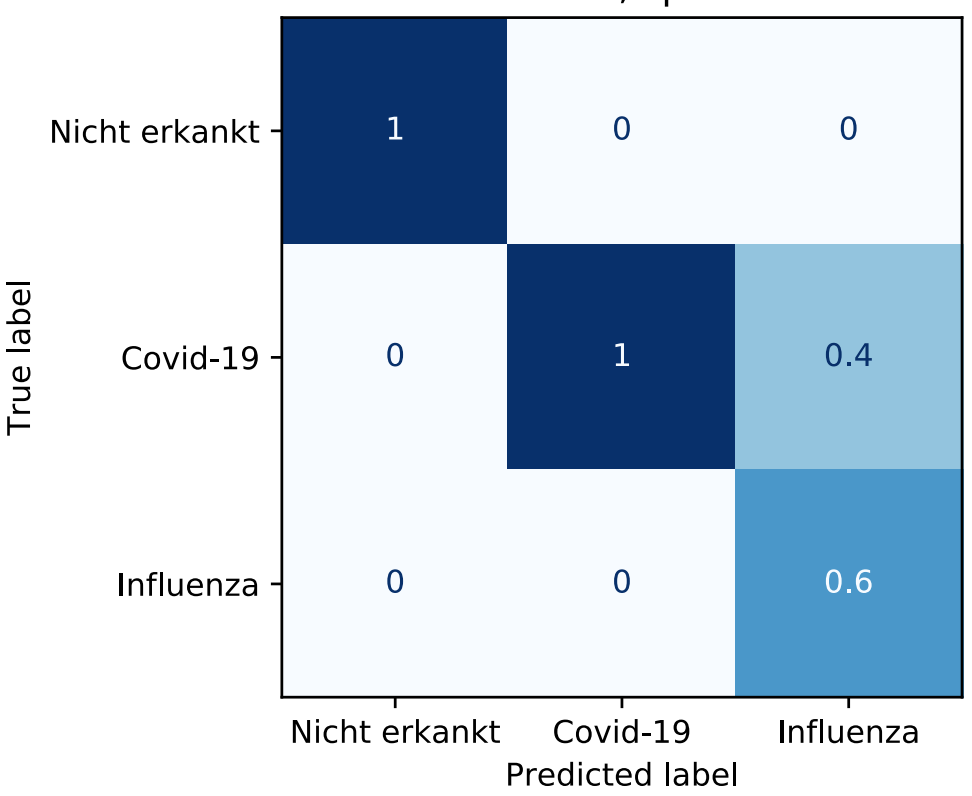


Anzahl der Covid-19 Patienten, die als Influenza-Fälle fehlklassifiziert wurden

Anzahl der Covid-19 Patienten, die (korrekt) als Covid-19 klassifiziert wurden.

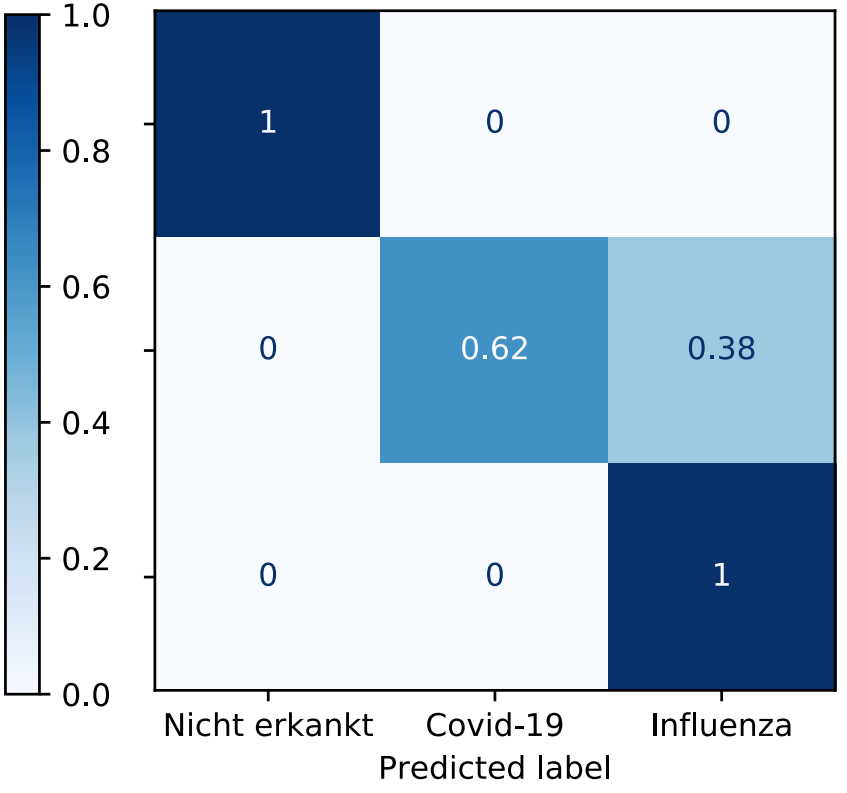
Normierte Confusion Matrizen

$$\tilde{C}_{ij} = C_{ij} / (\sum_k C_{kj})$$



Diagonaleinträge entsprechen PPV (Precision) der jeweiligen Klasse!

$$\tilde{C}_{ij} = C_{ij} / (\sum_k C_{ik})$$



Diagonaleinträge entsprechen TPR (Recall) der jeweiligen Klasse!

Daten-getriebene Modellierung[1]

1) Wir verstehen hierunter Verfahren aus dem Machine Learning. Sie können in der Bachelor-Veranstaltung „Machine Learning“ viele Modellierungsansätze im Detail kennenlernen.

Wdh | Daten-getriebene Modellierung

- Modelle ermöglichen Vorhersagen

Daten-getriebene Modellierung	vs	**Theorie-getriebene Modellierung**
Erstellung einer mathematischen Beschreibung mithilfe von Daten		Erstellung einer mathematischen Beschreibung mithilfe von Grundprinzipien (*first principles*)
Beispiele: Machine Learning Modelle (z.B. Bild-/ Spracherkennung)		Beispiele: Physikalische Modelle (z.B. Wettervorhersage)

Typen von Vorhersagen daten-getriebener Modelle:

	Klassifikation	**Regression**
Ergebnis der Vorhersage:	Kategorie / Klasse	numerischer Größe
Modell wird oft genannt:	Klassifikator (*classifier)*	Regressor (*regressor)*

Grundbegriffe aus dem Machine Learning

Beispiel

<u>Übung 11.2:</u>

Detektion eines epileptischen Anfalls (y=1; kein Anfall: y=0) anhand von Features X

Supervised Learning
(Überwachtes Lernen)

gesucht:

Unbekannte Target Function

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

Daten sind Input-Output-Paare:

$$(\mathbf{x_1}, y_1), \ldots (\mathbf{x_N}, y_N)$$

Ziel:
Approximiere unbekanntes f mit g

Unsupervised Learning
(Unüberwachtes Lernen)

gesucht:

Unbekannte Funktion, die die Daten beschreibt:

$$f : \mathcal{X} \mapsto ?$$

Daten: $(\mathbf{x_1}, \ldots, \mathbf{x_N})$

Es gibt keine „labels“, nur Input-Daten.

Ziel:
Finde ein f, das die Daten gut beschreibt

Beispiele

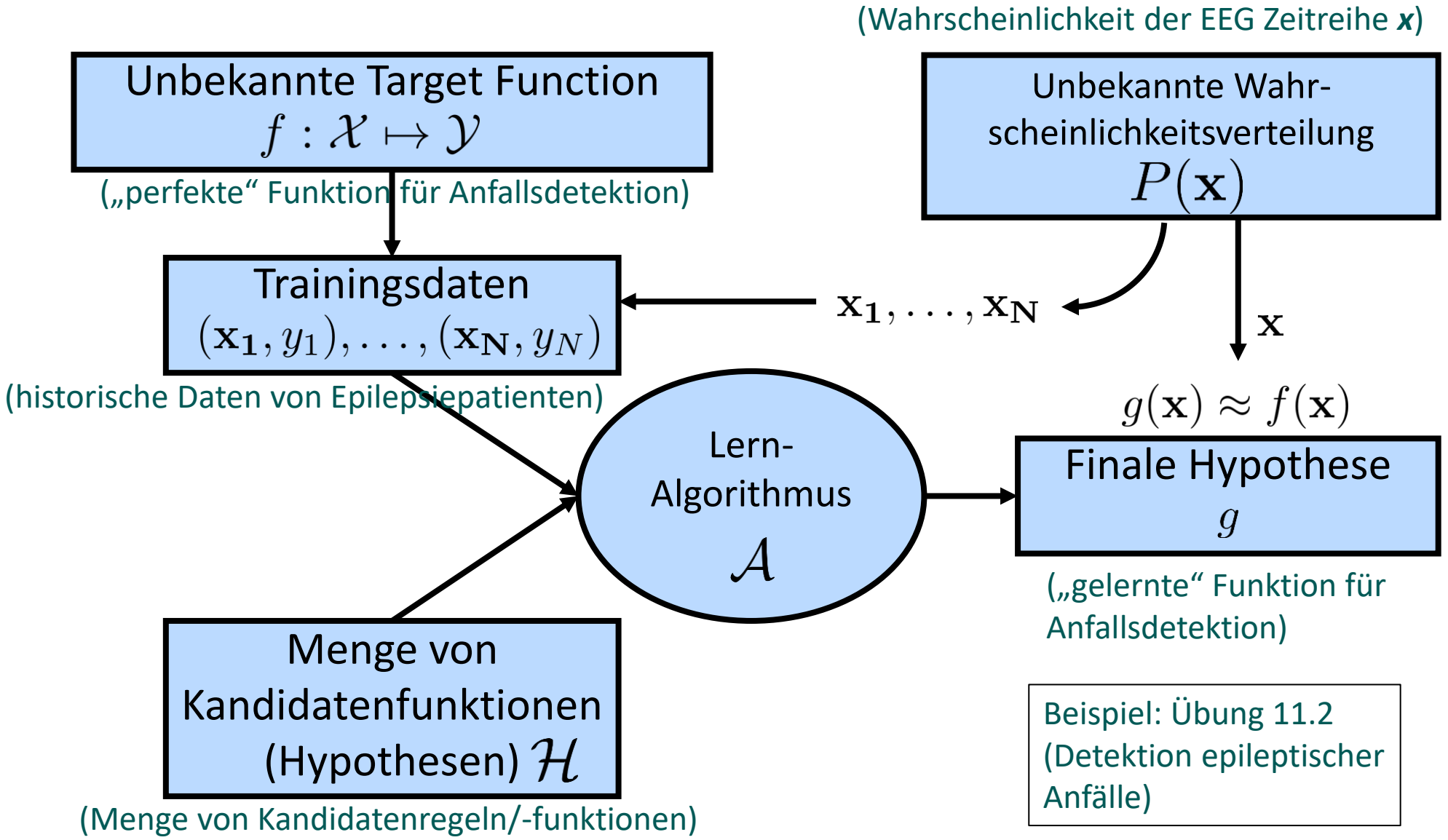
<u>Übung 7.2:</u>

Dimensions-reduktion des schweizer Rollkuchens

<u>Übung 8.1</u>

Clustering eines Datensatzes

Grundbegriffe aus dem Supervised Machine Learning



Daten-getriebene Modellierung | Grundbegriffe

Beispiel Detektion epileptischer Anfälle (aus Übung 11.2)

Daten:

Frage: Was waren die Features X? Was waren die Labels y? F

Training- und Testset

Daten $\mathcal{D}$ (*N* Datenpunkte)

Trainingsset $\mathcal{D}_{\text{train}}$ (*N-K* Datenpunkte)

Testset $\mathcal{D}_{\text{test}}$ (*K* Datenpunkte)

1. Teile Daten $\mathcal{D}$ **zufällig** in Trainingsset und Testset auf (wir wollen keine Verzerrungen bei der Einteilung erzeugen)

   In der Praxis: Typische Wahl ist $K = N / 5$

2. Erhalte das finale Modell auf dem Trainingsset: $g \in \mathcal{H}$

3. Berechne den Vorhersagefehler[1] des Modells mithilfe von g auf dem Testset.

1) z.B. Klassifikationsfehler, Regressionsfehler; siehe Vorlesung 10

Daten-getriebene Modellierung | Vorgehen

1. Sie haben Daten mit Features und Labels vorliegen.
   Klassifikationsproblem: Labels sind diskret (Kategorien)
   Regressionsproblem: Labels sind reellwertig (reelle Zahlen)

2. Sie teilen die Daten in Trainings- und Testdaten auf.

3. Sie finden ein Modell *g* (finale Hypothese) mithilfe der Trainingsdaten.

4. Sie evaluieren den Fehler Ihres Modells auf den Testdaten, um die Qualität Ihres Modells für bisher „ungesehene Daten“ einschätzen zu können.

Sie können aus verschiedenen Modellklassen (Mengen von Kandidatenfunktionen) wählen. Im Kurs „Machine Learning“ lernen Sie prominente Modellklassen kennen.

Nächste Nachbarn Modelle[1]

1) Diese Modelle lassen sich der Denkschule der Analogisten zuordnen, für die die Ähnlichkeit zwischen Objekten zentraler Ausgangspunkt für die Entwicklung von Lernmodellen ist.

Ähnlichkeit quantifizieren

Beachten Sie, dass es sehr unterschiedliche Konzepte gibt, um Ähnlichkeit zwischen Objekten zu quantifizieren.

Beispiele:

- Euklidische Distanz (Unähnlichkeitsmaß): $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$
- Kosinus-Ähnlichkeit: $\mathrm{CosSim}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \cdot \|\mathbf{x}'\|} \in [-1, 1]$

  Entspricht dem Kosinus des Winkels zwischen den Richtungsvektoren von x und x‘:

  $$\mathbf{x}^T \mathbf{x}' = \|\mathbf{x}\| \cdot \|\mathbf{x}'\| \cdot \cos(\theta)$$

- Jaccard-Koeffizient: $J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \in [0, 1]$
  (zur Charakterisierung der Ähnlichkeit der Mengen S_1 und S_2)

Nächste Nachbarn Modelle

Nächste Nachbarn (Nearest Neighbor) Modelle …

- ... zählen zu den einfachsten Machine Learning Modellen.
- ... können je nach Problemstellung schnelle und gute Vorhersagen liefern
- … kommen öfters zum Einsatz zur Schätzung einer Baseline

Typische Modelle:

- Nearest Neighbors (NN)
- k-Nearest Neighbors (kNN Klassifikation)
- k-Nearest Neighbors (kNN Regression)

Nearest Neighbors (NN)

- Trainingset: $\mathcal{D} = (\mathbf{x}_1, y_1), \ldots (\mathbf{x}_N, y_N)$
- Ähnlichkeitsmaß / Distanzmaß $d(\mathbf{x}, \mathbf{x}')$
  (wir wählen hier die euklidische Distanz: $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$)
- Betrachten Sie einen beliebigen Punkt $\mathbf{x}$ im Featureraum:
  Die Featurevektoren $\mathbf{x}_i$ des Trainingssets liegen gemäß d
  in unterschiedlichen Abständen zu $\mathbf{x}$.
  - Featurevektoren des Trainingset seien nun benannt nach ihrer Nähe zu **x**: $\mathbf{x}_{[1]}$ bezeichne den nächsten Nachbarn von **x**, $\mathbf{x}_{[2]}$ bezeichne den zweitnächsten Nachbarn von **x** (und so weiter): $d(\mathbf{x}, \mathbf{x}_{[1]}) \leq d(\mathbf{x}, \mathbf{x}_{[2]}) \leq \ldots \leq d(\mathbf{x}, \mathbf{x}_{[N]})$

Nearest Neighbor (NN) Modell
Das Modell lautet: $g(\mathbf{x}) = y_{[1]}(\mathbf{x})$

Nearest Neighbors (NN)

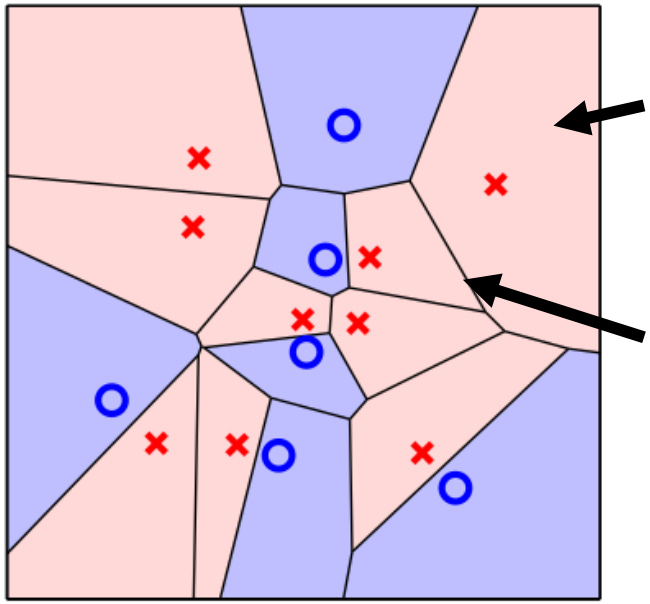
Nearest Neighbor (NN) Modell
Das Modell lautet: $g(\mathbf{x}) = y_{[1]}(\mathbf{x})$

Auf Deutsch: Für einen beliebigen Datenpunkt **x**, suche den zu **x** nächsten Punkt aus dem Trainingset ($\mathbf{x}_{[1]}$) und gebe das Label ($y_{[1]}$) dieses Punktes zurück.

Frage: Wann findet das Training dieses Modells statt? **F**

Nearest Neighbors (NN)

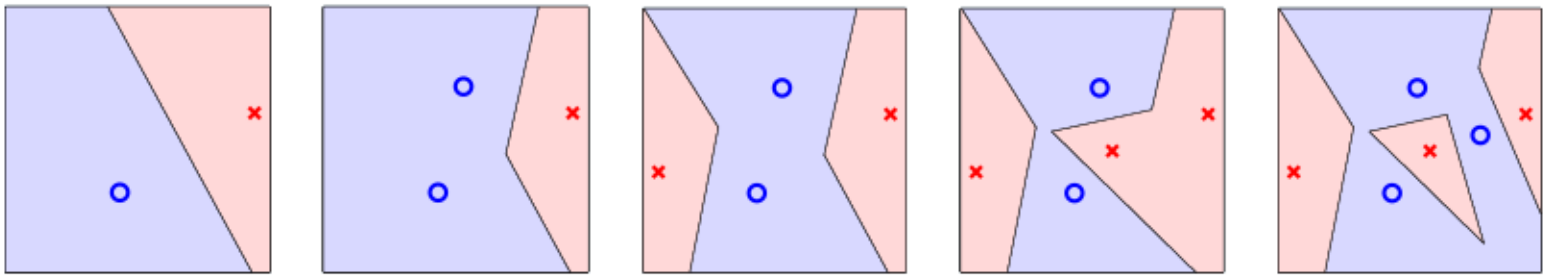
Nearest Neighbor (NN) Modell
Das Modell lautet: $g(\mathbf{x}) = y_{[1]}(\mathbf{x})$



Durch NN erzeugte **Voronoi-Regionen:**

- Voronoi-Region: Menge aller Punkte, die näher an einem Zentrum (hier: Trainingsdatenpunkt) liegen als an allen anderen Zentren
- Voronoi-Diagramm entspricht den gesammelten Grenzen zwischen den Regionen.
- Jeder Trainingsdatensatz induziert ein Voronoi-Diagramm.

Nearest Neighbors (NN)



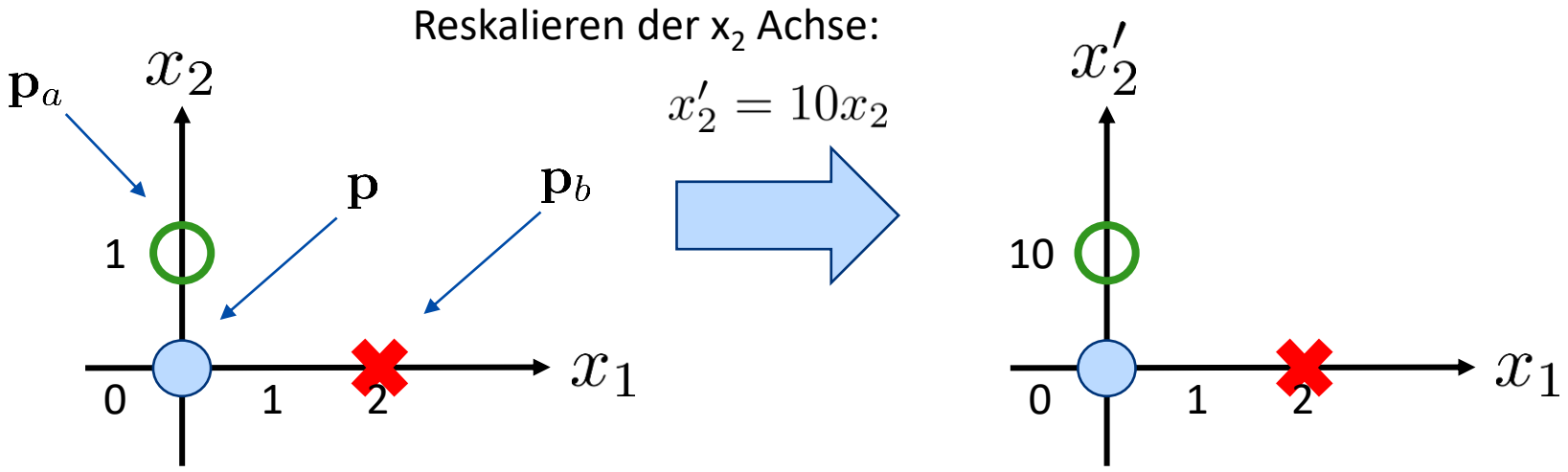
Das NN Modell kann bei steigender Datenpunktanzahl zu sehr komplexen Entscheidungsgrenzen führen.

Nearest Neighbors (NN)

Skalierung der Daten beeinflusst NN Modelle.

Beispiel:

- Zweidimensionaler Featureraum mit Features x_1 und x_2
- Betrachte Punkte ***p***, $\boldsymbol{p_a}$, $\boldsymbol{p_b}$



- $\boldsymbol{p_a}$ ist nächster Nachbar von ***p***
  weil $d(\mathbf{p}, \mathbf{p}_a) = 1 < d(\mathbf{p}, \mathbf{p}_b) = 2$
- $\boldsymbol{p_b}$ ist nächster Nachbar von ***p***
  weil $d(\mathbf{p}, \mathbf{p}_b) = 2 < d(\mathbf{p}, \mathbf{p}_a) = 10$

k-Nearest Neighbors (kNN)

Eine Verallgemeinerung des NN Modells ist kNN (k-Nächste Nachbarn):

Sei $k \geq 1$ eine ganze Zahl.

Signumsfunktion (gibt das Vorzeichen des Arguments zurück)

k-Nearest Neighbor (kNN) Modell (binäre Klassifikation)

Das Modell lautet:

$$g(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{k} y_{[i]}(\mathbf{x})\right);\ y \in \{-1, 1\}$$

k=1 → NN Modell (= 1NN Modell)

Falls dieser Ausdruck 0 ist, wird meist zufällig g(x) = 1 oder -1 ausgegeben.

Auf Deutsch: Für einen beliebigen Datenpunkt **x**, suche die k zu **x** nächsten Punkte aus dem Trainingset und gebe das Label der Mehrheit dieser nächsten Punkte zurück.

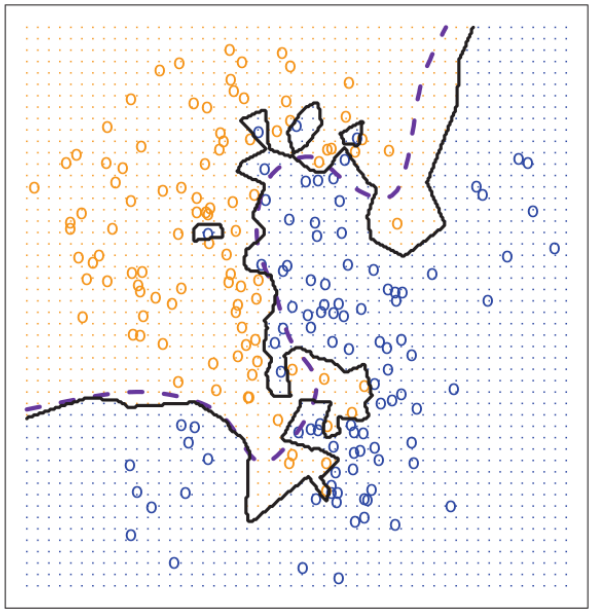
k-Nearest Neighbors (kNN)

Wahl von k

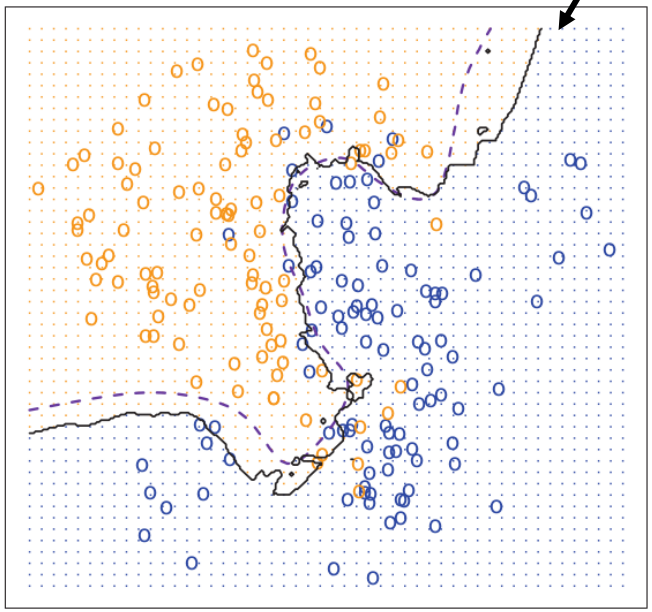
Beispiel:
Simulierte Daten, Klassenzugehörigkeit gemäß einer Wahrscheinlichkeitsverteilung gesampelt

Grenze des kNN Modells zwischen beiden Klassen

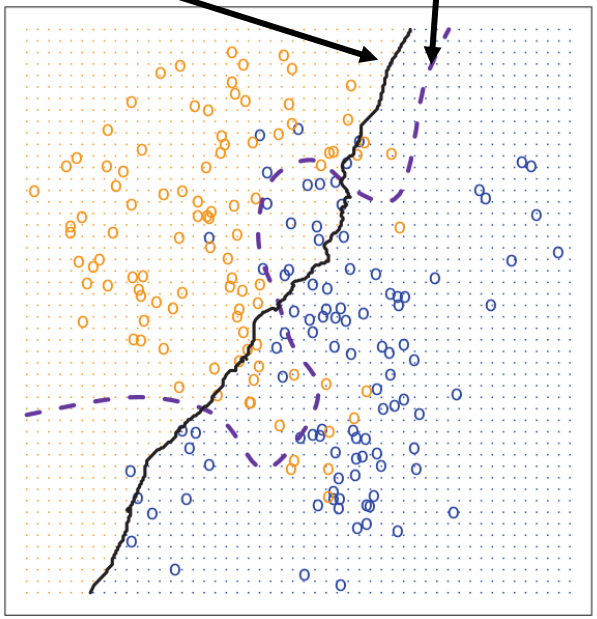
auf dieser Linie ist die Wahrscheinlichkeit zur blauen Klasse zu gehören genau 50%



$k = 1$



$k = 10$



$k = 100$

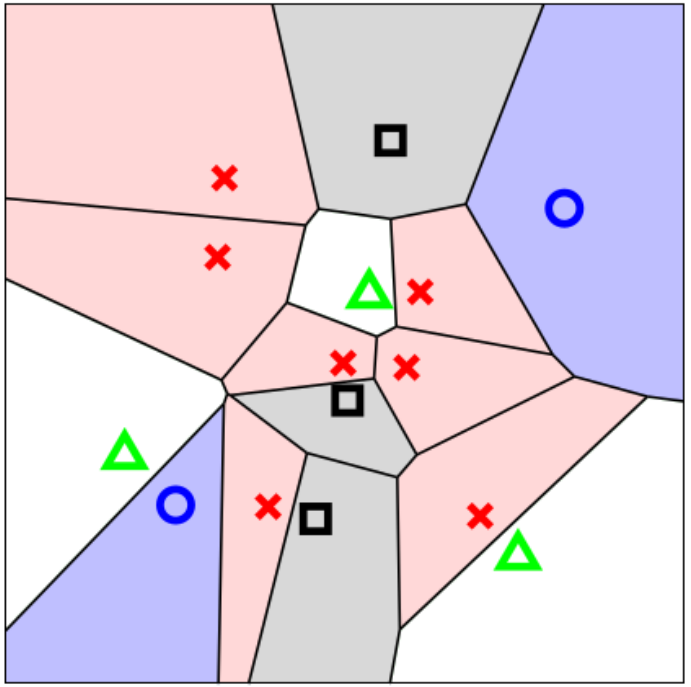
Frage: Wird die Entscheidungsgrenze einfacher oder komplizierter, wenn sich k erhöht?

k-Nearest Neighbors (kNN)

Multiklassen-Klassifikation mit kNN:

1. Selektiere die k nächsten Nachbarn für einen Punkt ***x***.

2. Die Klasse eines Punktes ist dann die Mehrheitsklasse aller nächsten Nachbarn.



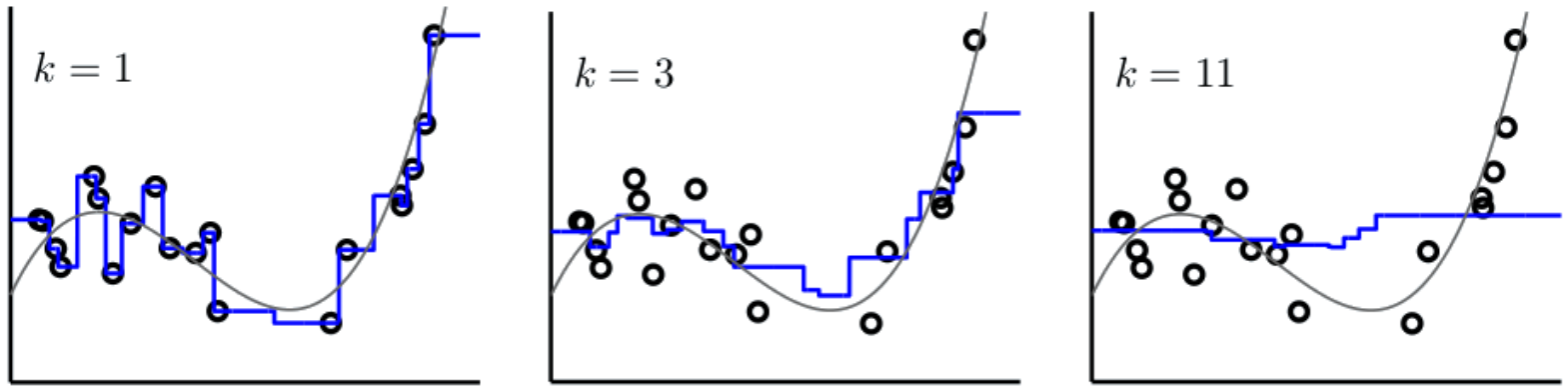
$k = 1$

k-Nearest Neighbors (kNN) Regression

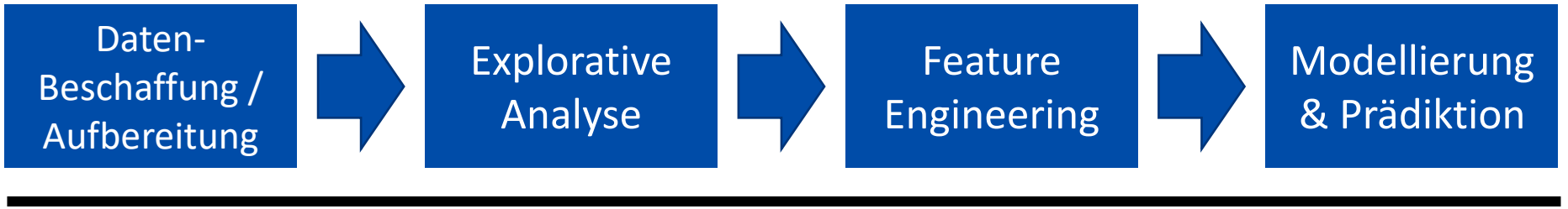
- Sie erinnern sich: Bei Regressionsprobleme sind die Labels y_i reellwertig.

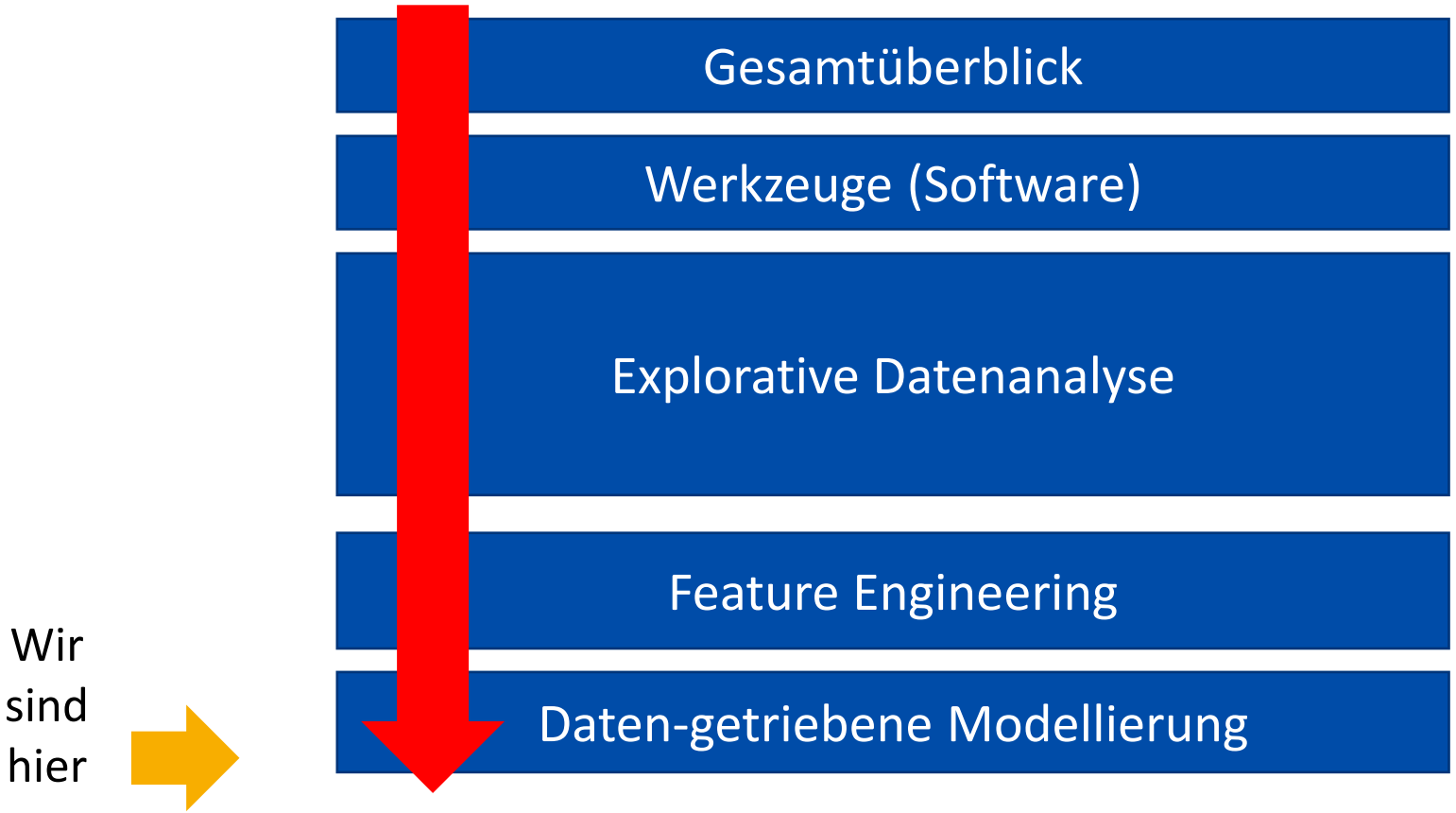
kNN-Regression:

- Mittelwert-Bildung über die Labels der k nächsten Nachbarn
- Finale Hypothese: $g(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{k} y_{[i]}(\mathbf{x})$



Data Science





Daten-aufbereitung (wird in den Übungen behandelt)

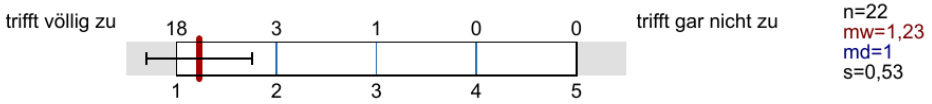
Lehrevaluation Aachen

22 Studierende haben sich an der Lehrevaluation beteiligt.

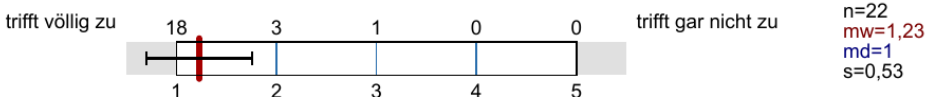
Was Sie besonders positiv angemerkt haben:

- Interaktion zwischen Ihnen und mit dem Dozenten, Lernatmosphäre
- Struktur und Art der Übungen, DataCamp, Kahoot
- Dass man der Vorlesung online folgen konnte

Der/die Lehrende ist im Umgang mit uns Studierenden freundlich und aufgeschlossen.



Der/die Lehrende schafft eine zur Mitarbeit anregende Lernatmosphäre.



- - der Einstieg mit DataCamp war genial. Danach noch eine handvoll weitere Kurse gemacht!
  - Das Praktikum mit notebooks ist ebenfalls top notch. Der rote Faden zieht sich durch die Aufgaben.
  - Es gibt bei den Aufgaben genug Tipps (Links zu Bibliothekendokumentation) um schnell die Befehle umzusetzen und sich auf Data Science konzentrieren zu können
  - Zum Ende das Online-Quiz ist die Krönung und sollte auch beibehalten werden

  Die Übungen in den Jupyter Notebooks sind super. Die Korrektur und das Feedback sind hilfreich bei der Nachbearbeitung.

- Das man die Vorlesung Online verfolgen konnte, mit Kopfhörern, sodass man durchs Umfeld weniger abgelenkt wird.
- Die Arbeit an echten Datensätzen
- Übungen und Korrekturen waren sehr hilfreich
  Kahoot Quizze

Lehrevaluation Aachen

Was Sie als verbesserungswürdig angemerkt haben:

- Dass die Vorlesung online/hybrid und nicht in Präsenz gehalten wurde
- Formulierung mancher Übungsaufgaben
- Eigenes Lernverhalten

- Bei manchen Themen könnte stärker auf interaktive Visualisierungen verwiesen werden.
  Machine Learning war von den Anforderungen her viel spannender, da einiges vorausgesetzt wurde und einem mehr abverlangt wurde. Danach DS zu hören und festzustellen, dass die Grundlagen hier erst Schritt für Schritt erarbeitet werden, war eher langweilig und ein wenig enttäuschend.
  Dass die Vorlesung leider nur per Zoom übertragen wurde und terminlich nicht in Präsenz gehalten werden konnte, war sehr schade.

- Es besteht die Möglichkeit, dass man innerhalb der Übungen an ein Dead-End gelangt
  Vllt wäre es gut einige Hilfestellungen (in Form von Tipps oder Beispielen) für die Aufgaben bereit zu stellen
- Formulierung der Übungsaufgaben überarbeiten, da nicht immer klar ist, was gefordert ist
- Frei entscheidbar für den Studenten, ob er für die Vorlesung anwesend sein muss.
- Mehr Erklärung von Syntax, mehr Zusammenfassungen von technischen Themen mündlich und nicht nur schriftlich
- Präzensunterricht

Mein eigenes Lernverhalten (Mitarbeit, systematische Vor-/Nacharbeit etc.) beurteile ich als:

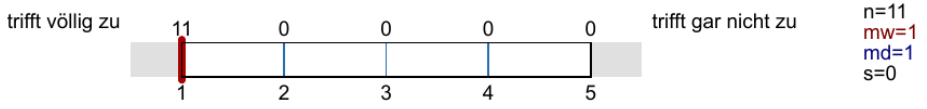
Lehrevaluation Jülich

11 Studierende haben sich an der Lehrevaluation beteiligt.

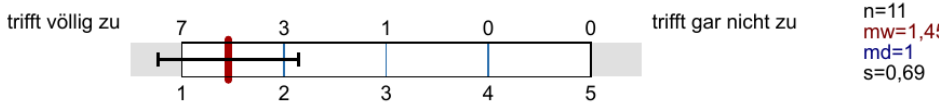
Was Sie besonders positiv angemerkt haben:

- Interaktion zwischen Ihnen und mit dem Dozenten, Lernatmosphäre
- Struktur und Art der Übungen, Kahoot
- Analyse echter Datensätze

Der/die Lehrende ist im Umgang mit uns Studierenden freundlich und aufgeschlossen.



Der/die Lehrende schafft eine zur Mitarbeit anregende Lernatmosphäre.



- - Aufgabenstellungen als Jupyter-Notebooks
  - Vorlesungsinhalte direkt in der Praxis mit Datensätzen ausprobiert
  - echte Datensätze (Tesla) verwendet
- Am anfang haben die Anwendungsbeispiele aus der Realen welt für sehr viel interesse gesort jedoch ist natürlich geringer geworden da theorie benötigt wurde
- Der Umfang der Übungen ist sehr gut und die Bearbeitung in Gruppen hilft beim Verständnis.
- Die Übungen waren sehr gut konzipiert und die Vorlesung gut aufgebaut. Insbesondere die aktive Mitarbeit, also das einbinden von Studenten durch Fragen vom Lehrenden und das eingehen auf Fragen von Student*Innen, war hilfreich.
- Kahoot = gut
- Kahoot und beste HA-Bearbeitung sehr motivierend
  Übungen klar strukturiert

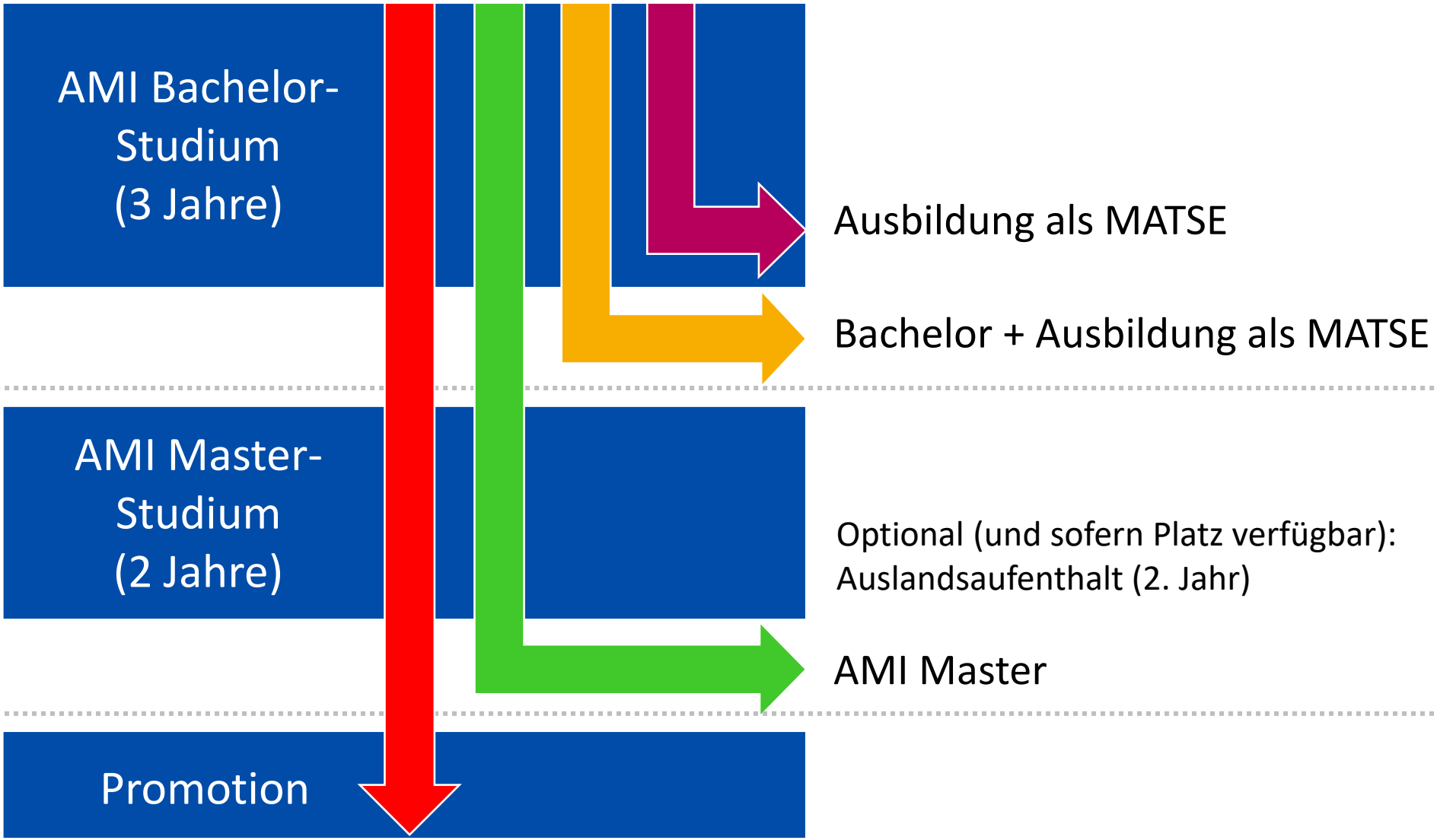
Lehrevaluation Jülich

Was Sie als verbesserungswürdig angemerkt haben:

- Dass die Vorlesung online/hybrid und nicht in Präsenz gehalten wurde
- Ausführlichere Hausaufgabenbewertung
- Strukturierte Anordnung der Folien (ILIAS)

- - Benamung der Foliensätze im ILIAS mit Thema der Vorlesung versehen
- Bewertung der Hausaufgaben teilweise sehr knapp, gerne ausführlicher/detaillierter
- Das Hybride Konzept zwischen Remote und Präsenz ist schlecht. Entweder hört man alles doppelt oder alles sehr leise. Lieber komplett Remote oder komplett Präsenz.
- Die Folien könnten etwas strukturierter hochgeladen werden, z.B. indem die Nummerierung an den Anfang des Dateinamens gesetzt wird oder getrennte Ordner für die beiden Folienarten angelegt werden.
- Die hybride Veranstaltung sollte meiner Meinung nach kein Dauerzustand werden.
  Es ist deutlich schwieriger zu folgen und die Konzentration zu behalten, da der Dozent (notwendigerweise) eher Leise in den Laptop redet.
  Die Vorlesung ist gut gemacht und inhaltlich interessant, aber ich weiß nicht ob ich sie in diesem Format nochmal belegen würde.

AMI Studiengänge



Überblick – Data Science

Einführung Data Science

Machine Learning

Weitere Wahlpflichtmodule (z.B. Data Analysis)

AMI Bachelor

AMI Master

Methoden-spezifische Vorlesungen

Machine Learning 1: Grundlagen und statistische Lerntheorie

Machine Learning 2: Deep Learning

Viele weitere Wahlpflichtmodule
(z.B.: numerische Algorithmen, Datenanalytik, statistische Modellierung, …)

Datendomänen-spezifische Vorlesungen

Zeitreihen

Zeitreihen und Prognoseverfahren

Bilder

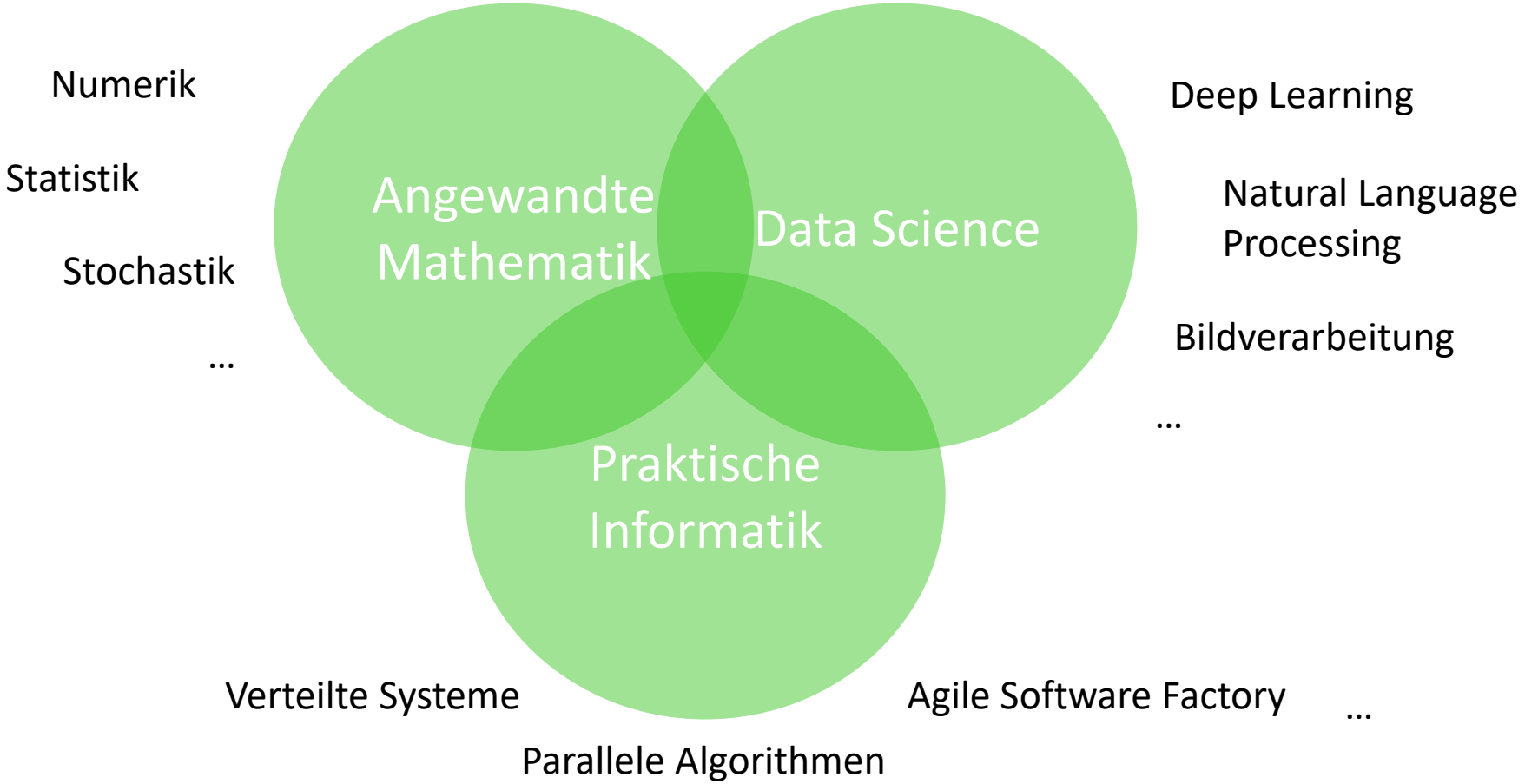
Digitale Bildverarbeitung

Sprache

Natural Language Processing

M. Sc. Angewandte Mathematik und Informatik

2 Jahre, 3 wählbare Schwerpunkte:



M. Sc. Angewandte Mathematik und Informatik

Beispiele für Berufsfelder in Industrie oder Forschung

- Simulation und Optimierung
- Data Science & Machine Learning Engineering
- Softwareentwicklung
- Beratung
- Produktentwicklung
- Statistische Qualitätskontrolle
- …

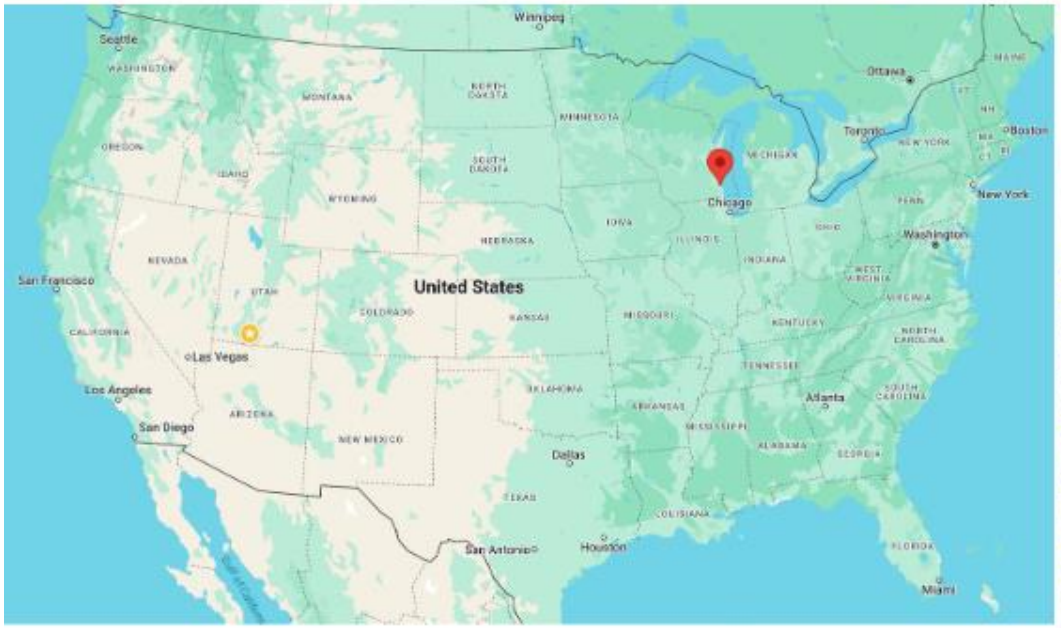
M. Sc. Angewandte Mathematik und Informatik

Weitere Informationen zum Masterstudiengang finden Sie auf der Webseite der Hochschule:

Antworten zu vielen praktischen Studiumsfragen finden Sie auch in den „Häufig gestellten Fragen“ (FAQ):

M. Sc. Angewandte Mathematik und Informatik

Auf Wunsch Möglichkeit eines Graduate Dual Degree
mit Auslandsaufenthalt (2. Studienjahr)
in den USA (University of Wisconsin-Milwaukee)
2 Abschlüsse: M. Sc. AMI und M. Sc. in Mathematics



direkt am Lake Michigan
150km nördlich von Chicago

ca. 600 000 Einwohner
tiefe Winter, warme Sommer

M. Sc. Angewandte Mathematik und Informatik

Graduate Dual Degree Ablauf

1. Beendigung des Bachelors
2. Immatrikulation im Master im September
3. Stundenplanbesprechung Ende September
4. Informationen zum Programm für Interessierte
5. Englisch-Test und Empfehlungsschreiben
6. Bewerbung an der UWM bis Ende Januar
7. Rückmeldung von der UWM im März/April

Weitere Informationen zum Dual Degree erhalten Sie bei:
Gitte Kremling, kremling@fh-aachen.de
Prof. Dr. Gerhard Dikta, dikta@fh-aachen.de