

Agenda

Data Science und Big Data - warum sollte das unterrichtet werden?

Das Projekt ProDaBi

Einstieg in Maschinelles Lernen - Unplugged

Umsetzung Maschinellen Lernens in Jupyter Notebook

Wie geht es weiter?

Data Science und Big Data in der Schule

- Data Science, Big Data und Künstliche Intelligenz sind keine integralen Bestandteile von Lehrplänen der verschiedenen Schularten
- **Aber:**
 - Bedeutung nimmt im Alltag immer mehr zu
 - Immer mehr Geräte, System und Anwendungen beinhalten KI-Anteile
 - Grundsätzliches Verständnis wird von **Allen** erwartet und benötigt
 - Ethische Fragestellungen zum Einsatz von KI in verschiedenen Bereichen verlangen Grundkompetenzen über KI
- **Zum Gestalten der eigenen, aktuellen und zukünftigen Lebenswelt ist die Fähigkeit zum Verständnis und zur Reflexion über die Funktion, Anwendung, die Chancen und Risiken von Systemen mit Elementen Künstlicher Intelligenz enorm wichtig!**
 - Vermittlung dieses Wissens ist nicht nur Aufgabe der Informatik
 - Diskussionen über KI und wie wir mit ihr leben wollen, ist eine Fragestellung für ALLE Fächer!



Bildquellen: vdi-nachrichten.de, amazon.de

Das Projekt ProDaBi - Data Science und Big Data in der Schule

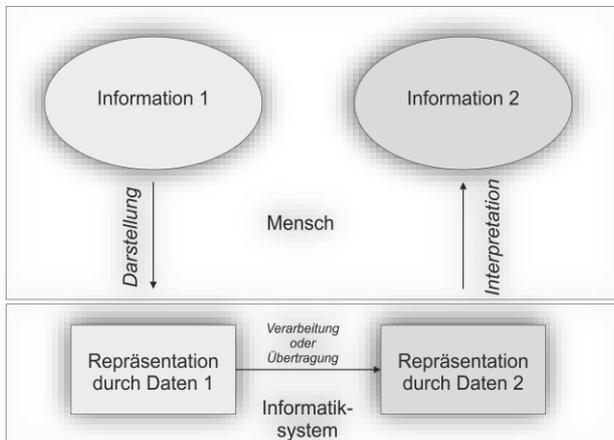


Rahmen der Erprobung

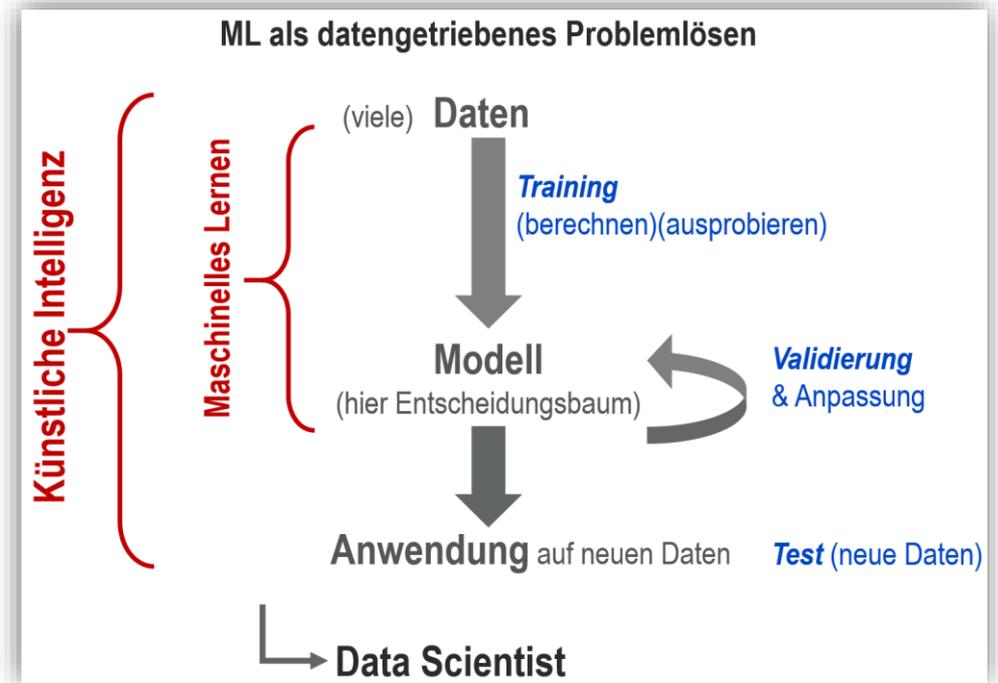
- Kooperation mit 2 paderborner Gymnasien
 - Projektkurs – Ersatz zu Facharbeit
 - 2 Schülerinnen, 17 Schüler der Q2
-
- Wöchentlich 3 Unterrichtsstunden (Montag 16:00 – 18:30)
 - 25 Wochen (September 2018 – April 2019)
-
- Lehr-Lern-Labor *PIN-Lab* der Informatikdidaktik der Uni-Paderborn
 - Aufzeichnung durch Raumkameras und Bildschirmaufzeichnung
 - Beobachtung des Unterrichts durch 2 Personen



Daten als Basis des Handelns im Kurs



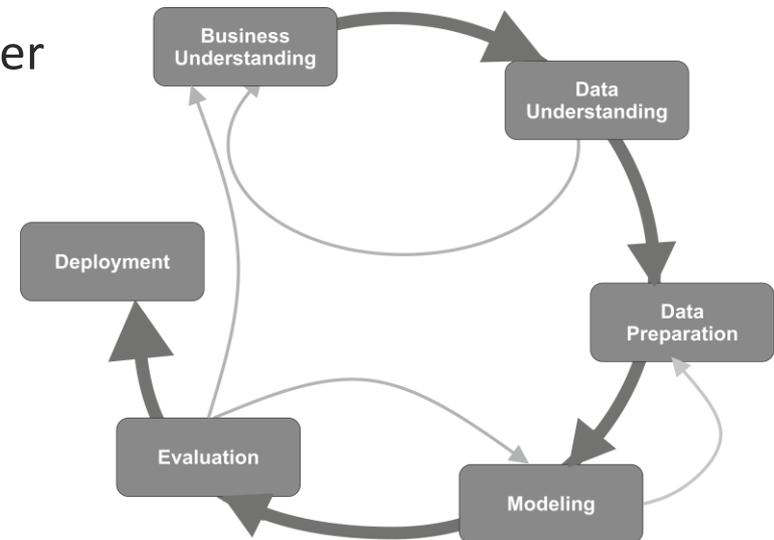
Information und Daten nach GI



Ergebnis einer Arbeitsphase während des Projektkurses

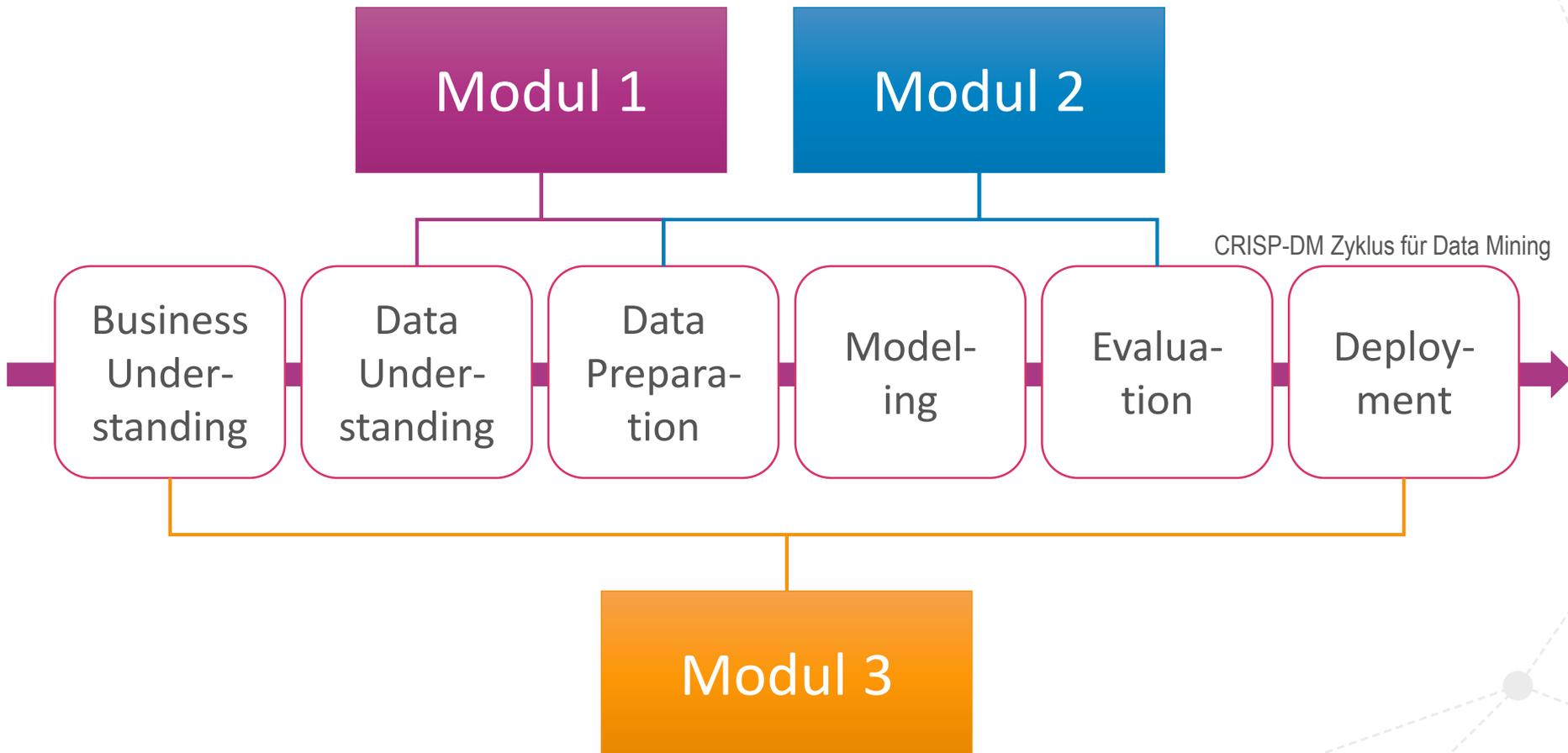
Der Projektkurs „Data Science“ – Planung

- Basis des Kurses ist der **CRISP-DM Zyklus**
- Entwicklung von einer algorithmen- zu einer datengetriebenen Sicht
- **Modularer Aufbau:**
 - **Modul 1:**
Daten verstehen, aufbereiten und präsentieren – Datendetektive
 - **Modul 2:**
Modelle erstellen und evaluieren – Maschinelles Lernen
 - **Modul 3:**
Anwenden des Wissens auf eine konkrete Fragestellung - Projektmodul



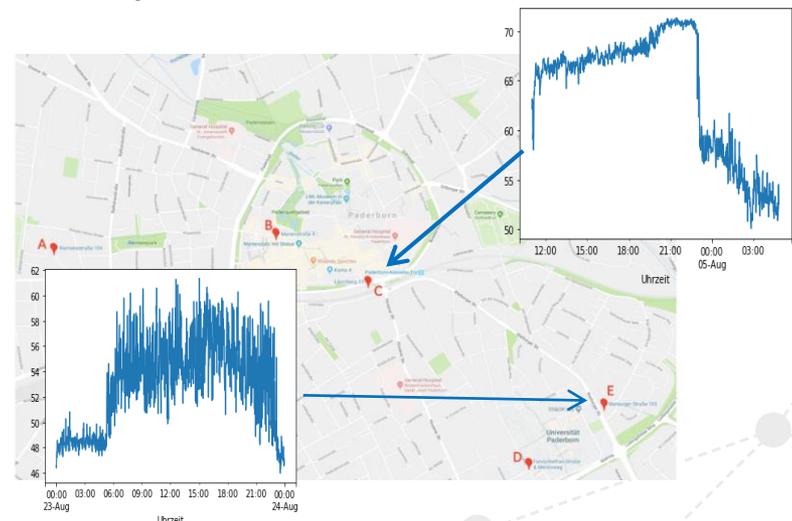
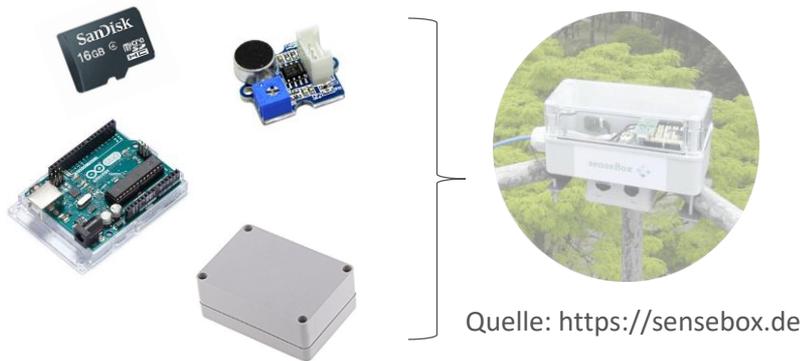
CRISP-DM Zyklus für Data Mining
nach Berthold et al. (2010)

Der Projektkurs „Data Science“ im Datenzyklus



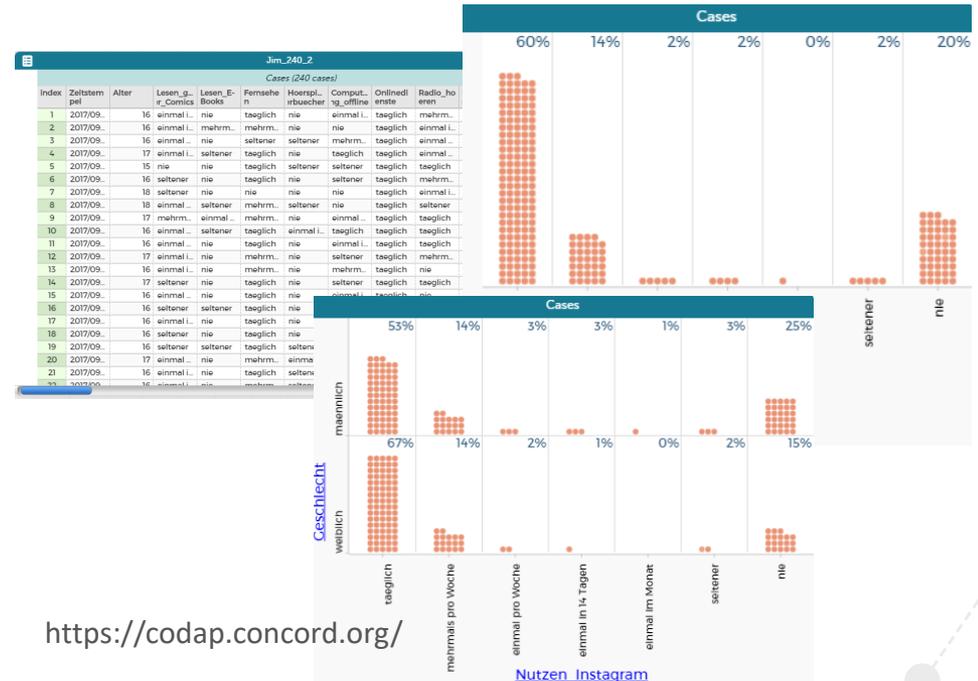
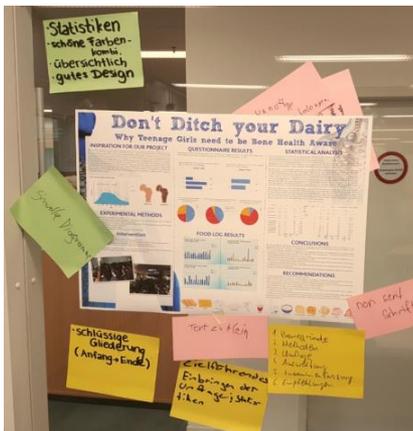
Modul 1: Datendetektive

- **Ziel:** Entwicklung von Datenkompetenz
- **Baustein 1:** Einführung in Data Science
 - Analyse von Akustikdaten
 - Erfassung mittels „Sensebox“
 - Verwendung von Jupyter Notebook und Python



Modul 1: Datendetektive

- **Baustein 2:** Einführung in die explorative Datenanalyse
 - Analyse des JIM-Datensatzes
 - Verwendung von CODAP
 - Multivariate Datenanalysen
 - Anwenden statistischer Konzepte
 - Analyse von Datenplakaten

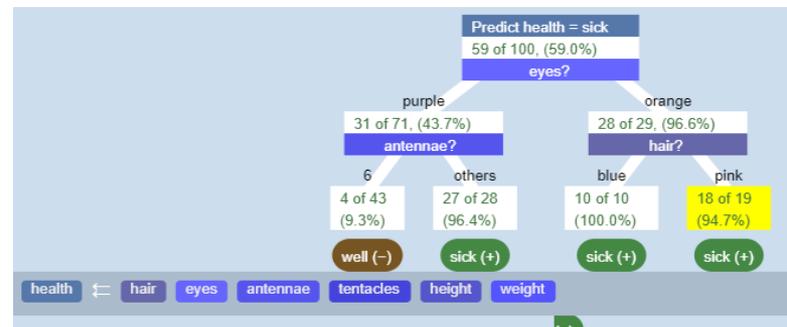
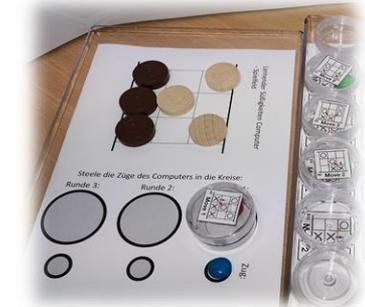


<https://codap.concord.org/>

[Nutzen Instagram](#)

Modul 2: Einstieg in Maschinelles Lernen

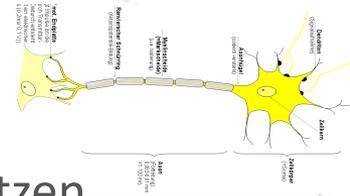
- **Ziel:** Kennen von Grundkonzepten Künstlicher Intelligenz und Maschinellen Lernens
- **Baustein 1:** Einführung in Maschinelles Lernen mit Entscheidungsbäumen
 - **LAC-H** (Lernender Analoger Computer für Hexapawn)
 - Decision Trees (Entscheidungsbäume) mit CODAP (Treetool) und Jupyter Notebook u.a. mit JIM-Datensatz



TP = 55, TN = 39, FP = 2, FN = 4

<https://tinyurl.com/ydhn7fqm>

Modul 2: Einstieg in Maschinelles Lernen

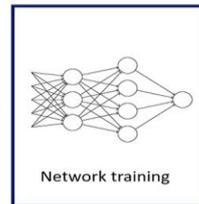


Baustein 2: Maschinelles Lernen mit Künstlichen Neuronalen Netzen

- Einstieg durch Unplugged-Aktivität „Brain in the Bag“
- Erkennen der Parameter und Kenngrößen durch Training von Netzen mit „Playground Tensorflow“
- Modellierung eigener Netze



- **MNIST-Datenbank** zur Erkennung handschriftlicher Ziffern
- Python und Jupyter Notebook
- Eigene Ziffern zur Validierung

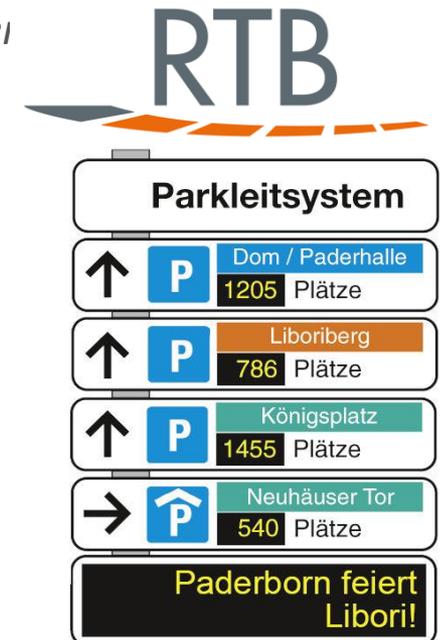


- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



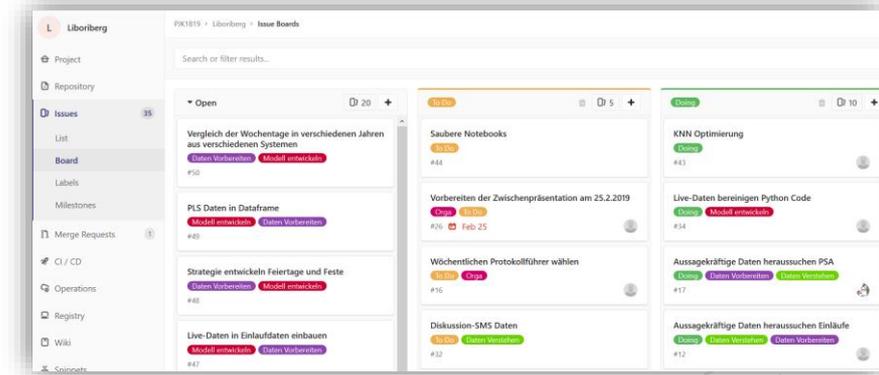
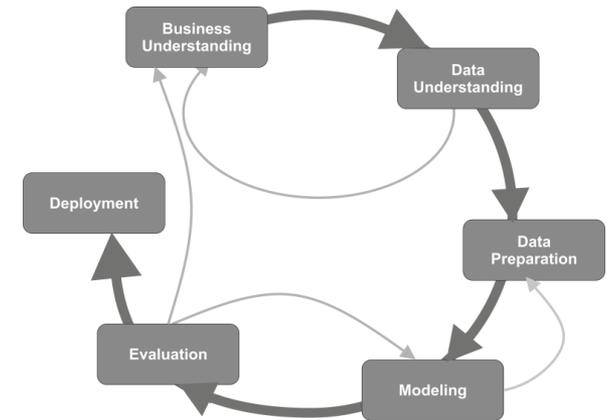
Modul 3: Wissen anwenden – Projektmodul

- Kooperation mit: Abfall- und Entsorgungsbetriebe Paderborn (ASP, Parkraumbewirtschaftung) und Fa. RTB
- **Aufgabenstellung:**
 - Erstellung einer **Prognose** über die Belegung verschiedener bewirtschafteter Parkmöglichkeiten zu einem zukünftigen Zeitpunkt
 - Können die jeweiligen Belegungen – abhängig von verschiedenen Parametern – für einen zukünftigen Zeitpunkt mit einer hinreichend guten Genauigkeit **vorhergesagt** werden?
- **Verfügbare Daten:**
 - Daten des Parkleitsystems (Induktionsschleifen)
 - Daten aus Parkautomaten und Kassensystemen



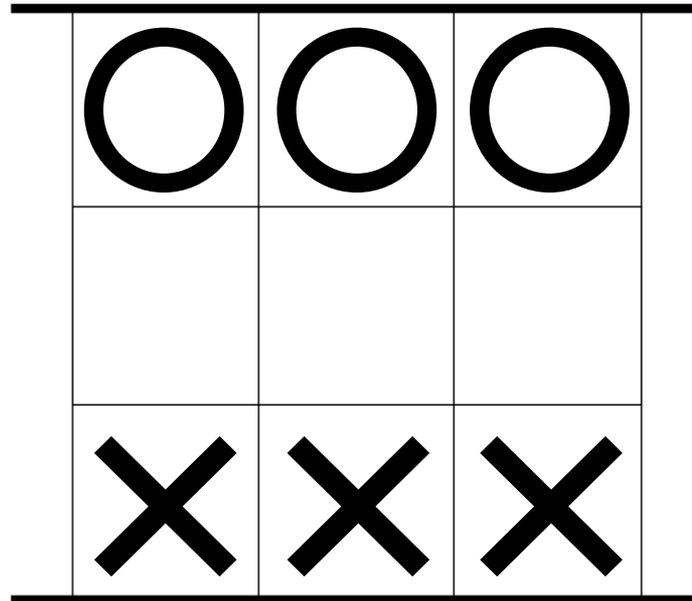
Modul 3: Wissen anwenden - Projektmodul

- **Projektgruppe 1:** Parkhausbelegung – Erfassung Parkdauer bei Bezahlung
- **Projektgruppe 2:** Parkplatzbelegung – Erfassung der geplanten Parkdauer
- **Vorgehen:**
 - Orientierung am **CRISP-DM Zyklus**
 - Arbeitsorganisation angelehnt an agilen Methoden
- **Werkzeuge:**
 - Jupyter Notebook mit Python
 - Git als Repository und Issue-Tool
- Abschlusspräsentation 1. April 2019



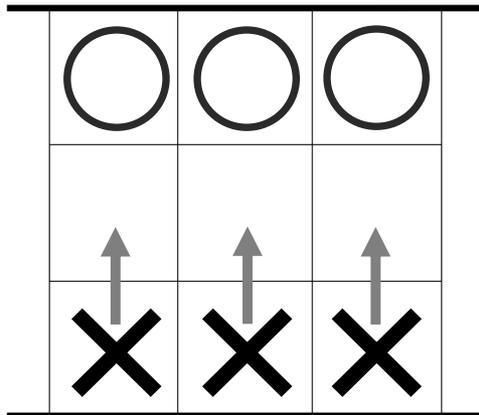
Einstieg in Maschinelles Lernen – Unplugged

Das Spiel Hexapawn - Bauernschach

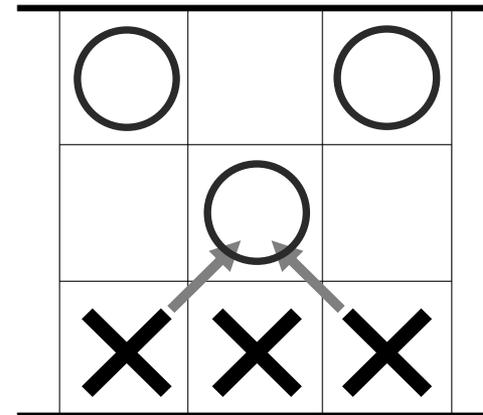


Wie spielt man?

Zugmöglichkeiten:

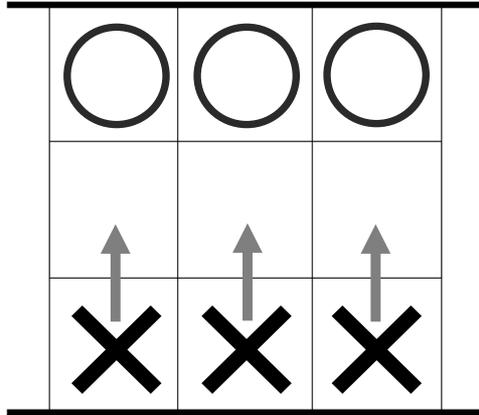


Geradeaus ziehen

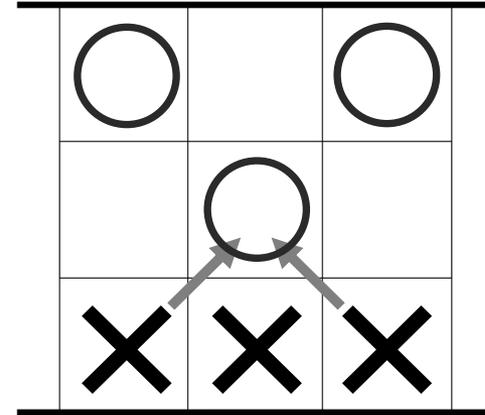


Schräg schlagen

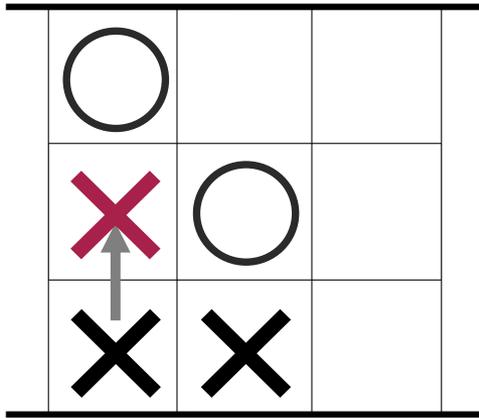
Wie gewinnt man?



Geradeaus ziehen

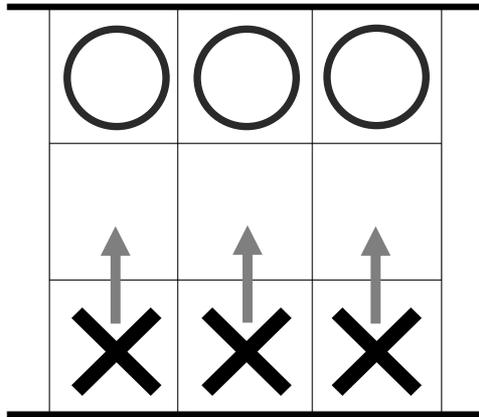


Schräg schlagen

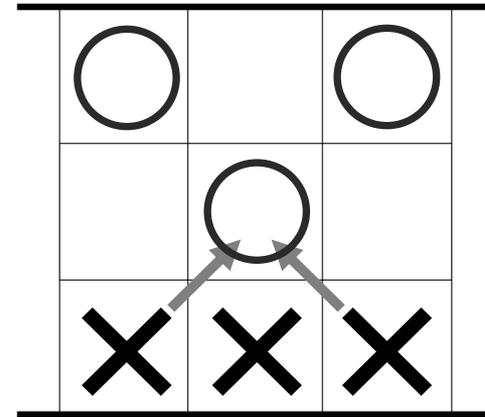


Gegner blockiert

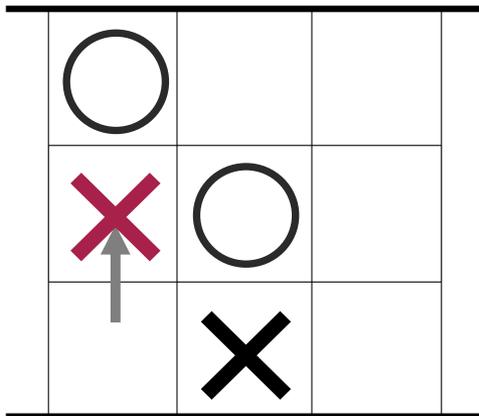
Wie gewinnt man?



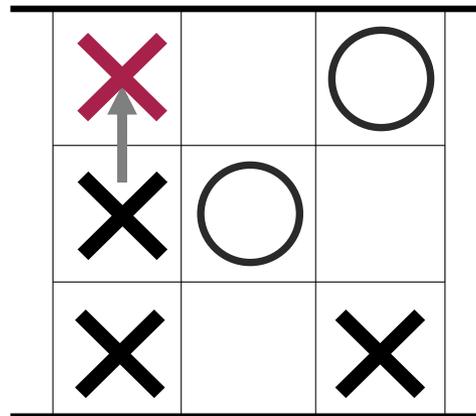
Geradeaus ziehen



Schräg schlagen

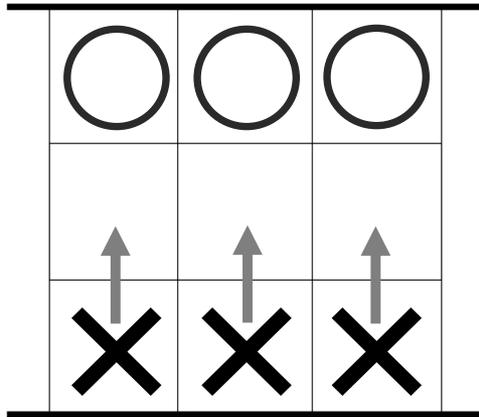


Gegner blockiert

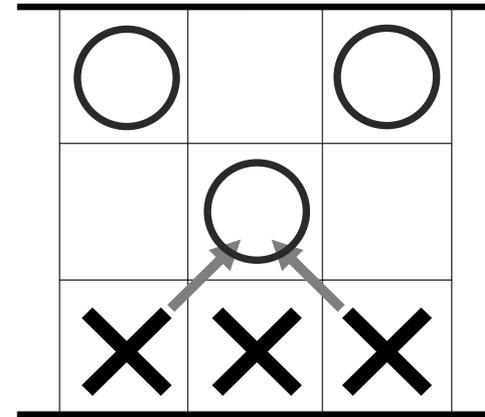


Stein erreicht gegnerische Basis

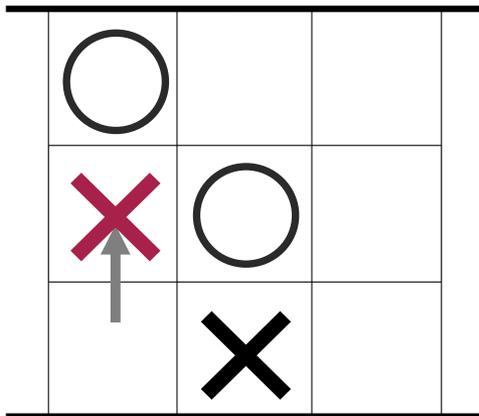
Wie gewinnt man?



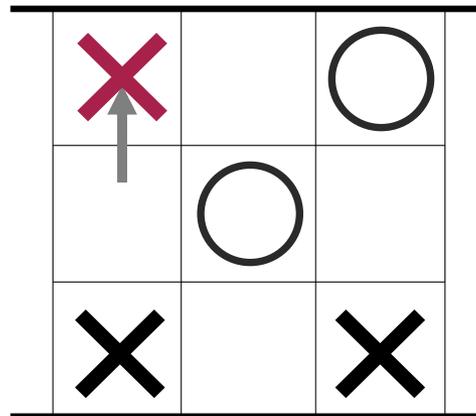
Geradeaus ziehen



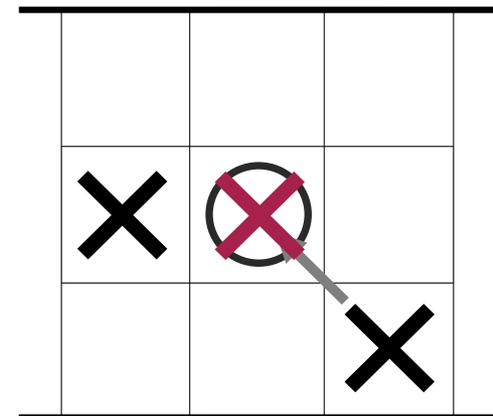
Schräg schlagen



Gegner blockiert



Stein erreicht gegnerische Basis



Gegner verliert alle Steine

Spielen mit dem LAC-H

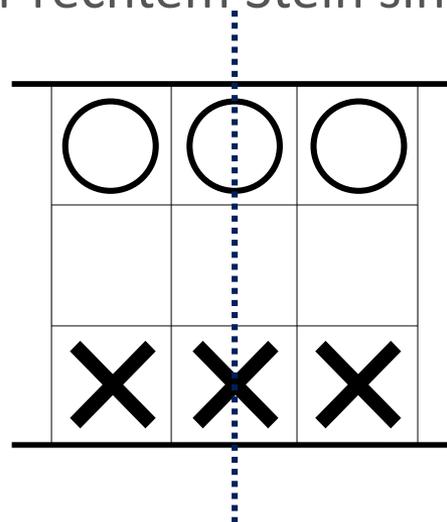
Jedes Spiel dauert bis zu drei Runden

Der LAC-H beginnt jedes Spiel als Zweites

Der Spieler zieht beim ersten Zug nur seinen linken oder mittleren Stein

- Das Spielfeld ist Achsensymmetrisch

→ Zug mit linkem oder rechtem Stein sind gleichbedeutend



Fazit – LAC-H

- Der Computer lernt nur aus Fehlern
 - Der Computer lernt schneller von besseren Gegenspielern
 - Das Lernen ist spezifisch für Bauernschach
 - Das Lernen kann optimiert werden durch Reduzierung des Regelsatzes
-
- „Lernen“ bedeutet hier nicht verstehen
 - „Lernen“ ist hier Löschen von Zügen, die nicht erfolgreich waren
-
- **Bedeutet Lernen beim Menschen auch, dass er vergisst?**

Maschinelles Lernen

LAC-H

- Wodurch?* Lernen durch
... Verlieren / Fehler
... Spielen
- Wie?* Ungünstige Züge werden entfernt
- Was?*
- Hexapawn / Bauernschach ~~spielen~~ gewinnen
 - nur diese Aufgabe → Lernen ist an diese Aufgabe angepasst

Agent / Lerner

- Bestärkendes Lernen
 - Daten
 - Training
- Anpassen der Gewichte auf „0“
(Wahrscheinlichkeit einer Farbe)
Analogie: Backpropagation bei KNN*
- Strategie lernen

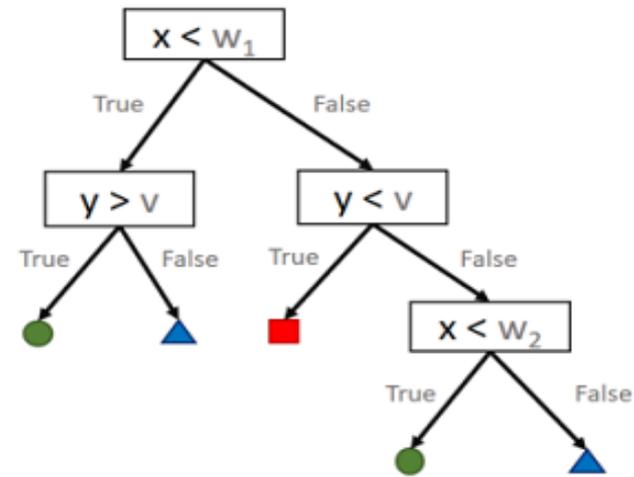
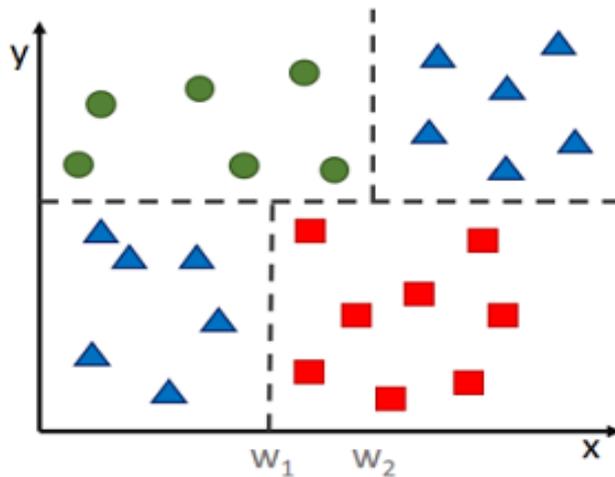
Oberstufe / nur Lehrkraft

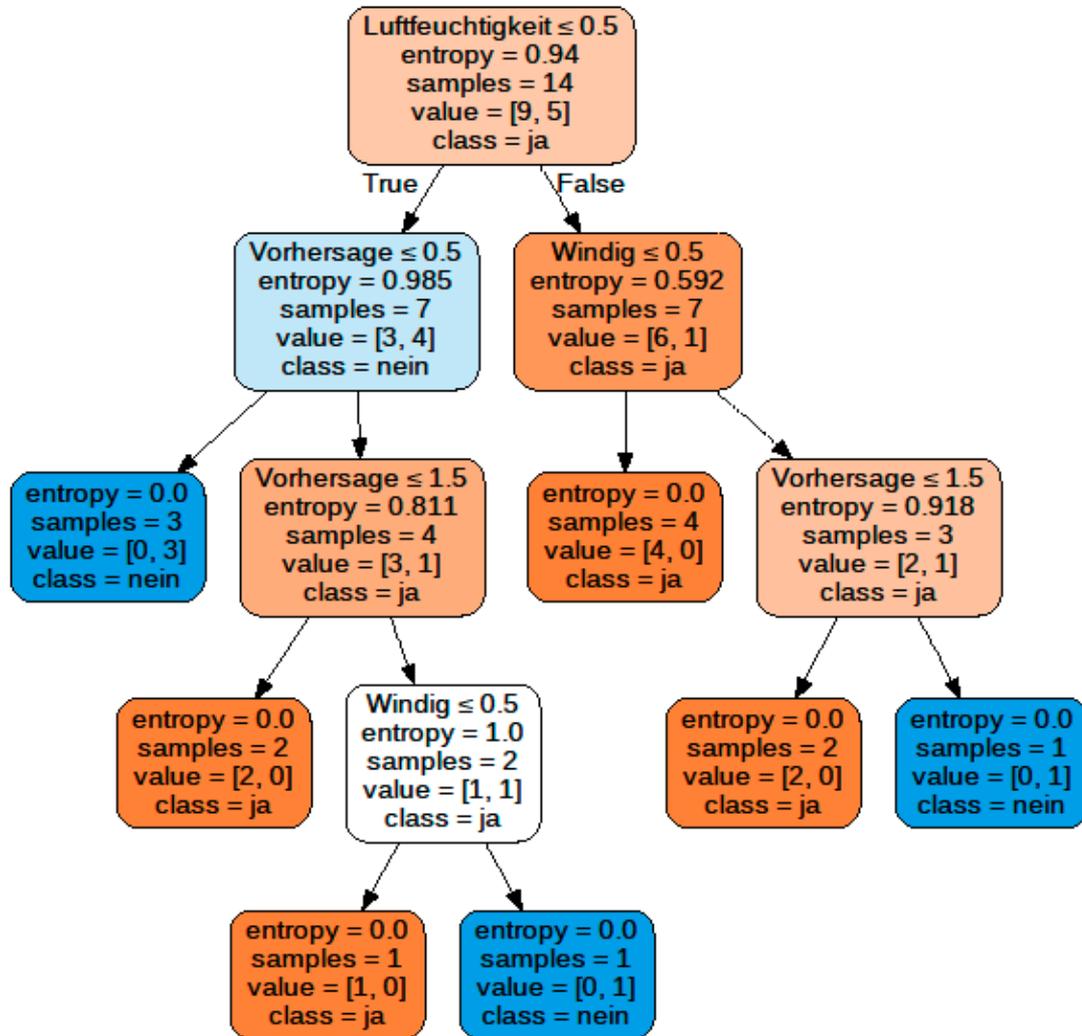
Umsetzung Maschinellen Lernens mit Jupyter Notebook

- Fokus auf zwei Lernverfahren:
 - Entscheidungsbäume
 - Künstliche Neuronale Netze

Entscheidungsbäume

- Leicht zu interpretieren
- Liefern Klassifikation mit Begründung
- Können aus Beispielen "gelernt" werden – einfache Algorithmen möglich, effizient und skalierbar





Einführung in Künstliche Neuronale Netze (KNN)

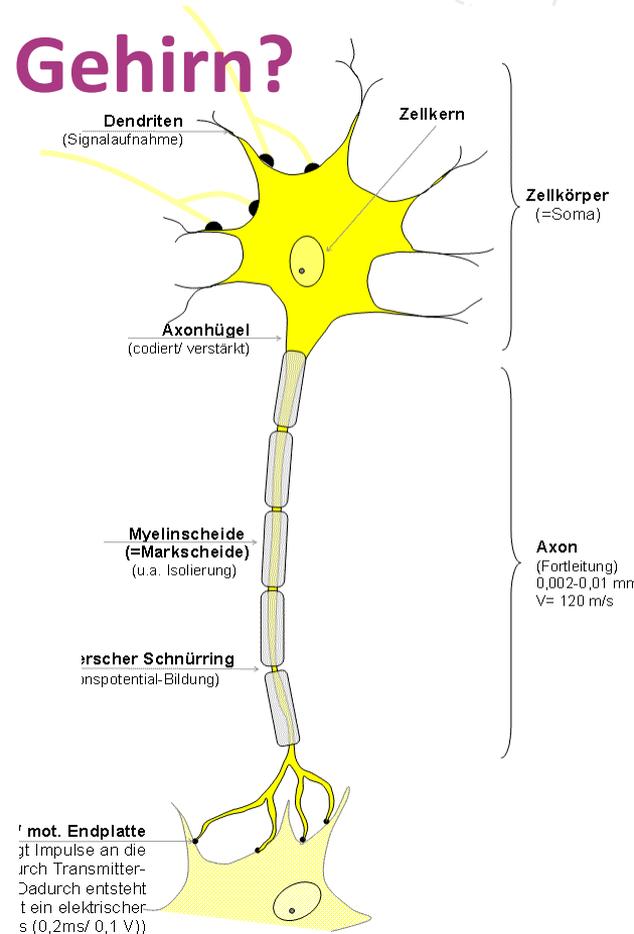
- Drei verschiedene Arten von Neuronen kennengelernt, die in unterschiedlichen Bereichen des KNN eingesetzt werden.
 - Eingabeneuronen -> Eingabeschicht
 - Ausgabeneuronen -> Ausgabeschicht
 - Zwischenneuronen -> Verdeckte Schicht(en)
- Die Grundidee eines Neurons ist aber bei allen drei Arten gleich. Dies wollen wir näher betrachten.

Grafiken und Folien für KNN adaptiert nach:

Rashid, T. (2016). *Make your own neural network*. CreateSpace Independent Publishing Platform.

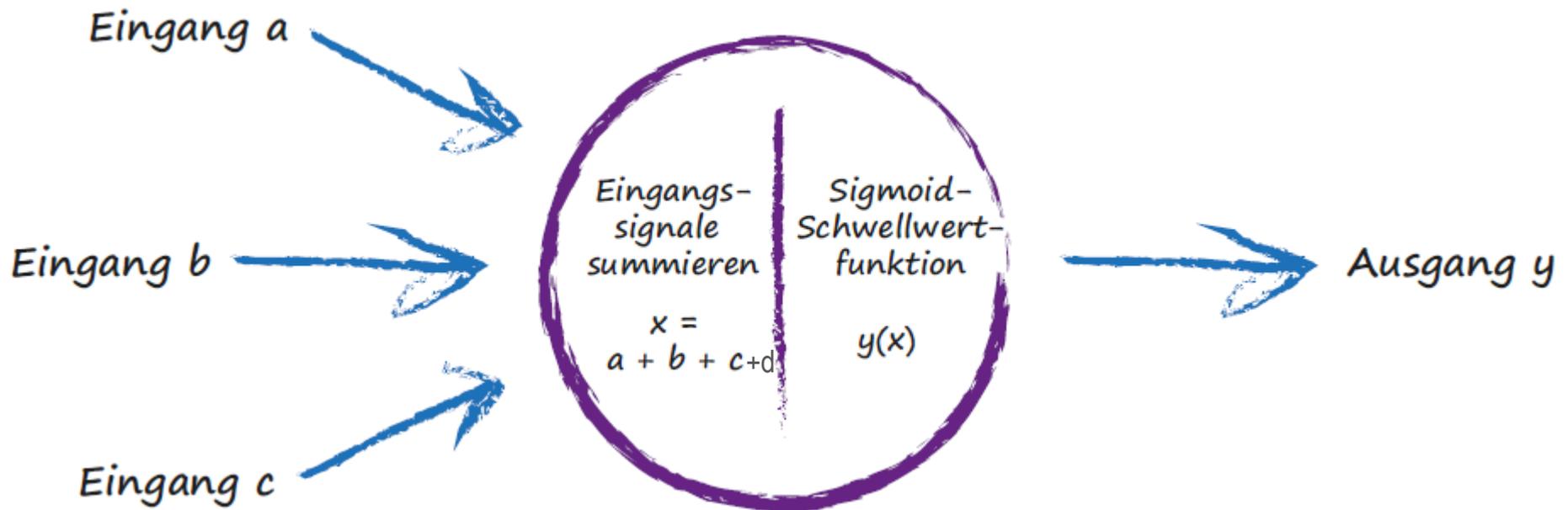
Wie funktioniert das menschliche Gehirn?

- Ein menschliche Gehirn besteht aus ca. 90 Milliarden miteinander verbundenen Nervenzellen, den sogenannten Neuronen.
- Jedes einzelne Neuron hat vier wesentliche Bestandteile:
 - Dendriten
 - Zellkern
 - Axon

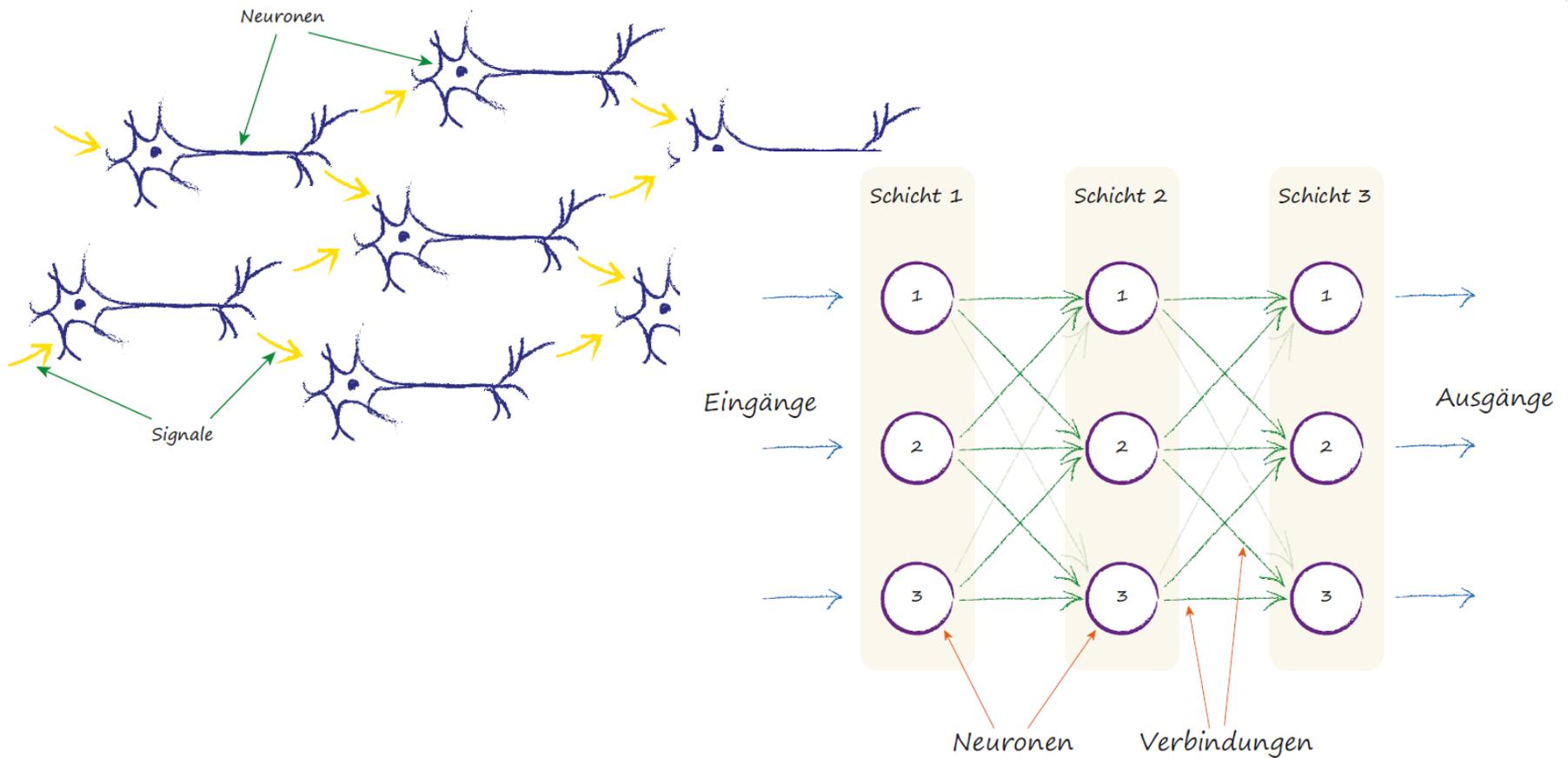


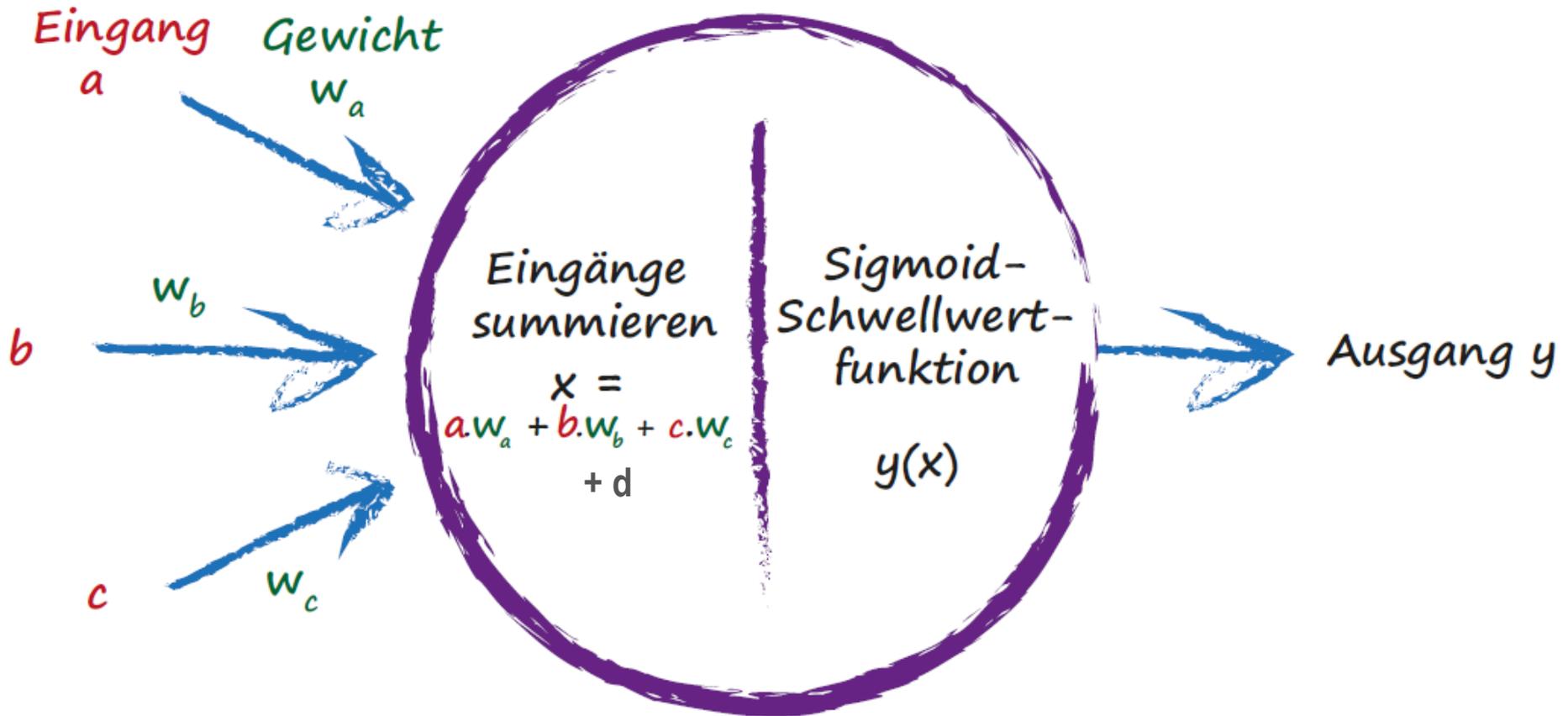
<http://ausmalbilderfuerkinder.club/nervenzelle-aufbau-arbeitsblatt/arbeitsblatt-7-motorische-endplatte-neu-aufbau-von-nervenzellen-pertaining-to-nervenzelle-aufbau-arbeitsblatt/>

Modell eines künstlichen Neurons



Netz von Neuronen





Keras: The Python Deep Learning Library (1)

- Wie man ein KNN mit Keras (<https://keras.io/>) definiert:
- `model = Sequential()` # KNN initiieren
- `model.add()` # Layer hinzufügen
 - `Dense` # Art des Layers
 - `Activation` # Aktivierungsfunktion: sigmoid, linear, tanh, relu,
softmax, ... <https://keras.io/activations/>

Keras: The Python Deep Learning Library (2)

- Wie man ein KNN mit Keras (<https://keras.io/>) definiert:
- `model.compile()` # KNN wird eingerichtet
 - `Optimizer` # Parameter des Gradientenverfahrens: <https://keras.io/optimizers/>
 - `Loss` # Zu optimierende Verlustfunktion: <https://keras.io/losses/>
- `model.fit()` # Modell wird trainiert
 - `Epochs` # Anzahl der Wiederholungen
 - `batch_size` # Anzahl Trainingsdaten bis Anpassung der Gewichte

Jupyter Notebook

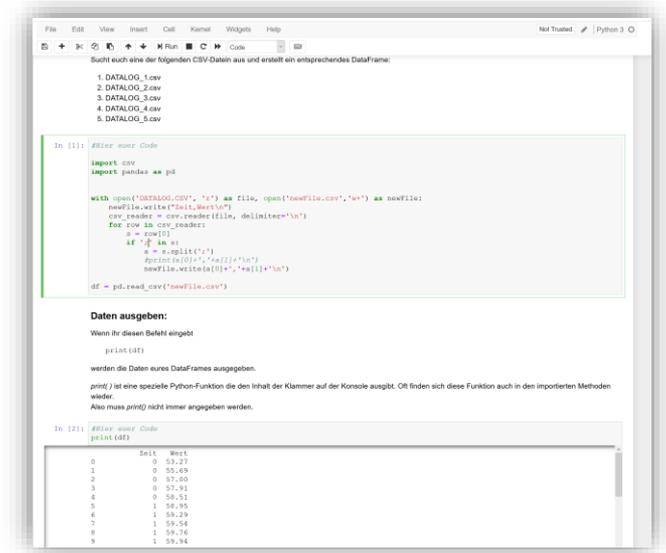
<https://jupyter.org/>

Tree-tool in CODAP:

<https://tinyurl.com/ydhn7fqm>

Playground Tensorflow:

<https://bit.ly/2BxldKN>



```
File Edit View Insert Cell Kernel Widgets Help Not Trained Python 3.0
Sucht euch eine der folgenden CSV-Daten aus und erstellt ein entsprechendes DataFrame:
1. DATALOG_1.csv
2. DATALOG_2.csv
3. DATALOG_3.csv
4. DATALOG_4.csv
5. DATALOG_5.csv

In [1]: #Hier mehr Code
import csv
import pandas as pd

with open('DATALOG_CSV', 'r') as file, open('newFile.csv', 'w') as newFile:
    newFile.write("Zeit,Wert\n")
    csv_reader = csv.reader(file, delimiter='\n')
    for row in csv_reader:
        a = row[0]
        if '0' in a:
            a = a.rsplit(':')
            #print(a[0]+' '+a[1]+' \n')
            newFile.write(a[0]+' '+a[1]+' \n')
df = pd.read_csv('newFile.csv')

Daten ausgeben:
Wenn ihr dessen Befehl eingibt
print(df)
werden die Daten eures DataFrames ausgegeben.
print() ist eine spezielle Python-Funktion die den Inhalt der Klammer auf der Konsole ausgibt. Ob finden sich diese Funktion auch in den importierten Methoden wieder.
Also muss print() nicht immer angegeben werden.

In [2]: #Hier mehr Code
print(df)
```

	Zeit	Wert
0	0	53.27
1	0	55.69
2	0	57.00
3	0	57.91
4	0	58.51
5	1	58.95
6	1	59.29
7	1	59.54
8	1	59.74
9	1	59.94

Weiterentwicklung des Projekts

Curriculum

Entwicklung für Sek II
allgemein

Design Based Research-
Ansatz wird vertieft

Konzentration auf
gesellschaftliche Aspekte

Projektkurs

Stärkere Verzahnung des
Projekts mit
Theoriemodulen

Neues Modul zu
Gesellschaftlichen
Auswirkungen von Big
Data und Künstlicher
Intelligenz

Unterrichtsmaterial

... Bausteine als gesamter
Projektkurs

... Stand-Alone Bausteine
mit minimalem
Programmieranteil

Evaluation und
Weiterentwicklung in
enger Verzahnung mit
Lehrkräften

Geplanter Ablauf im Folgekurs

