Algorithmus Schmiede

Data Science | Numerik | Physik

Data Science und Machine Learning für Softwareentwickler - eine Einführung

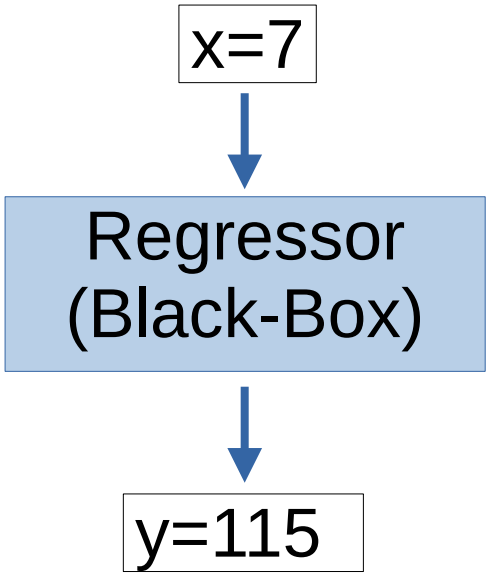
Dr. Markus Dutschke

Algorithmus Schmiede
Data Science | Numerik | Physik ©

Was ist ein Regressor?

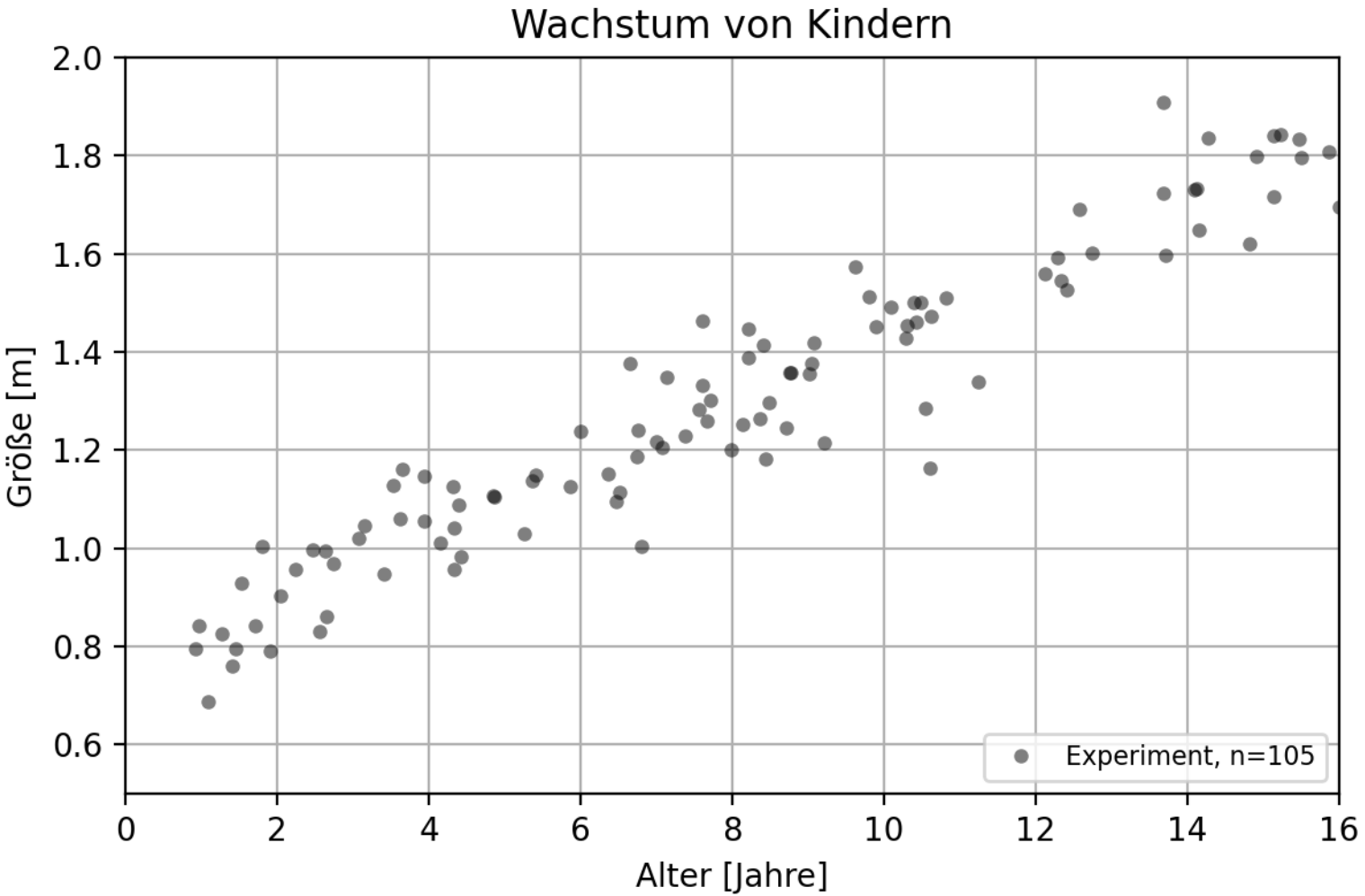
Einsatz:

- x-Wert hineinwerfen
  y-Wert herausbekommen
- Training mit x-y Paaren
  statt Implementierung von Formel



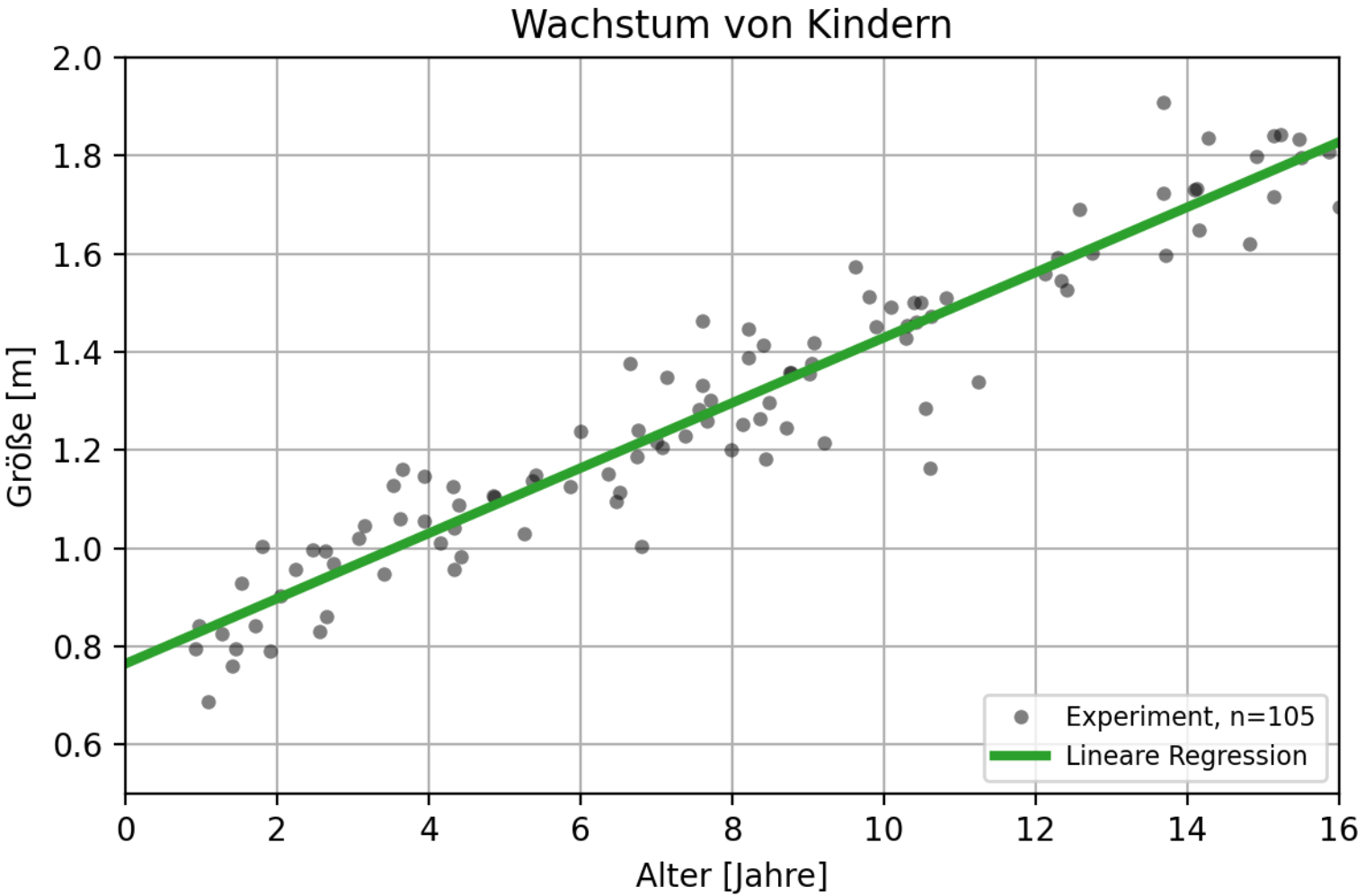
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Lineare Regression



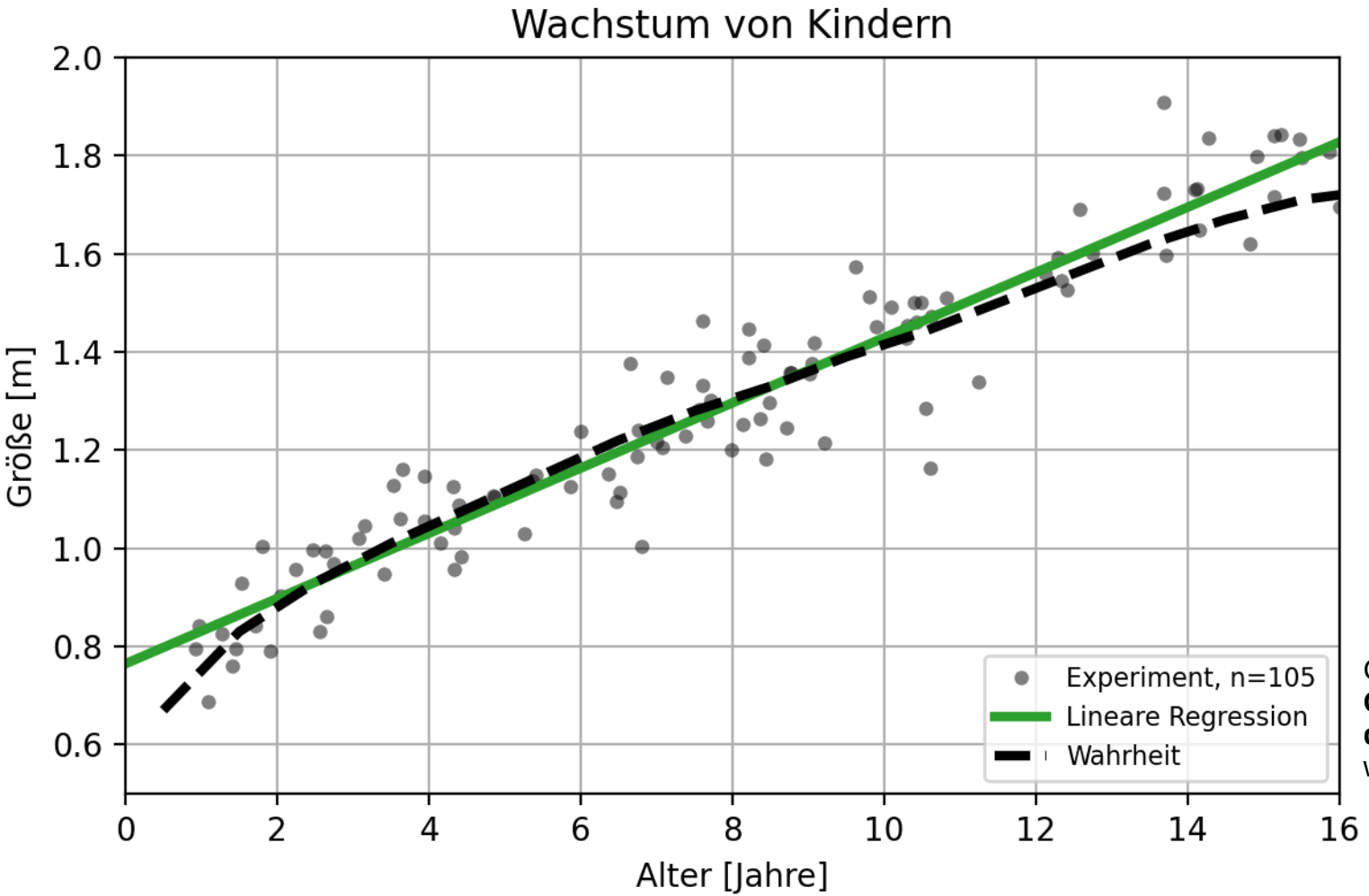
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Lineare Regression



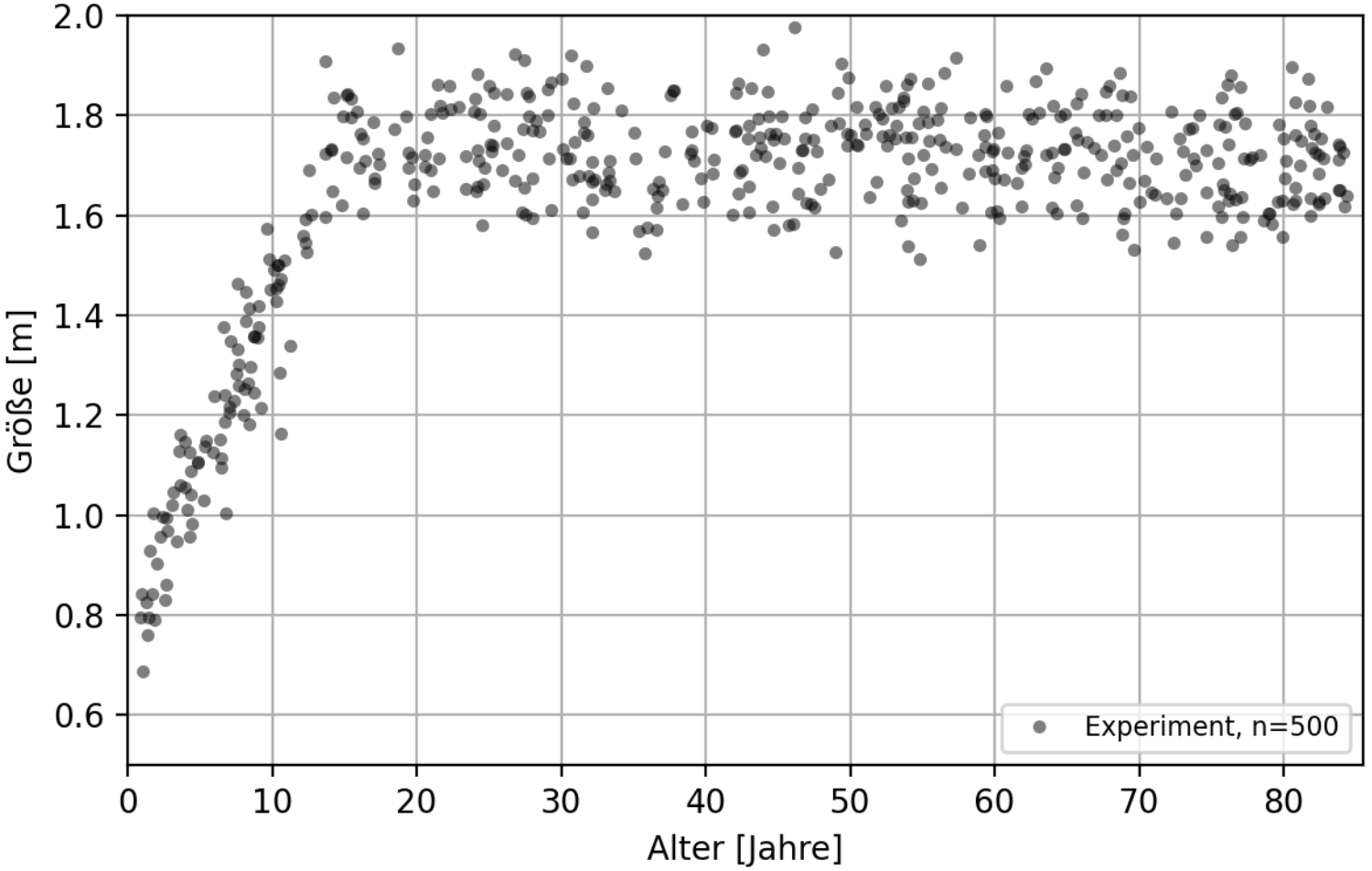
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Lineare Regression



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

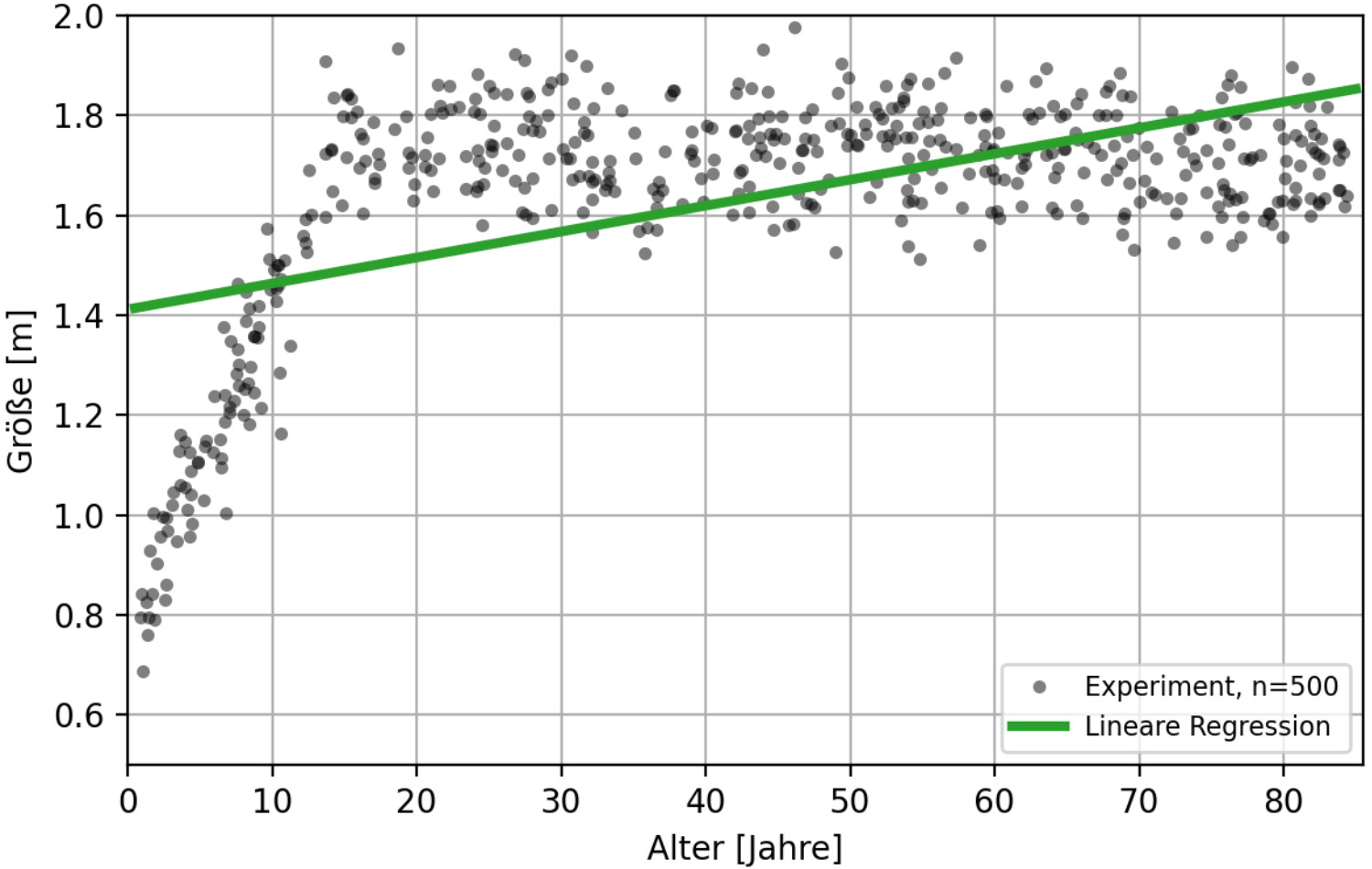
Überblick Regressoren



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Algorithmus Schmiede
Data Science | Numerik | Physik ©

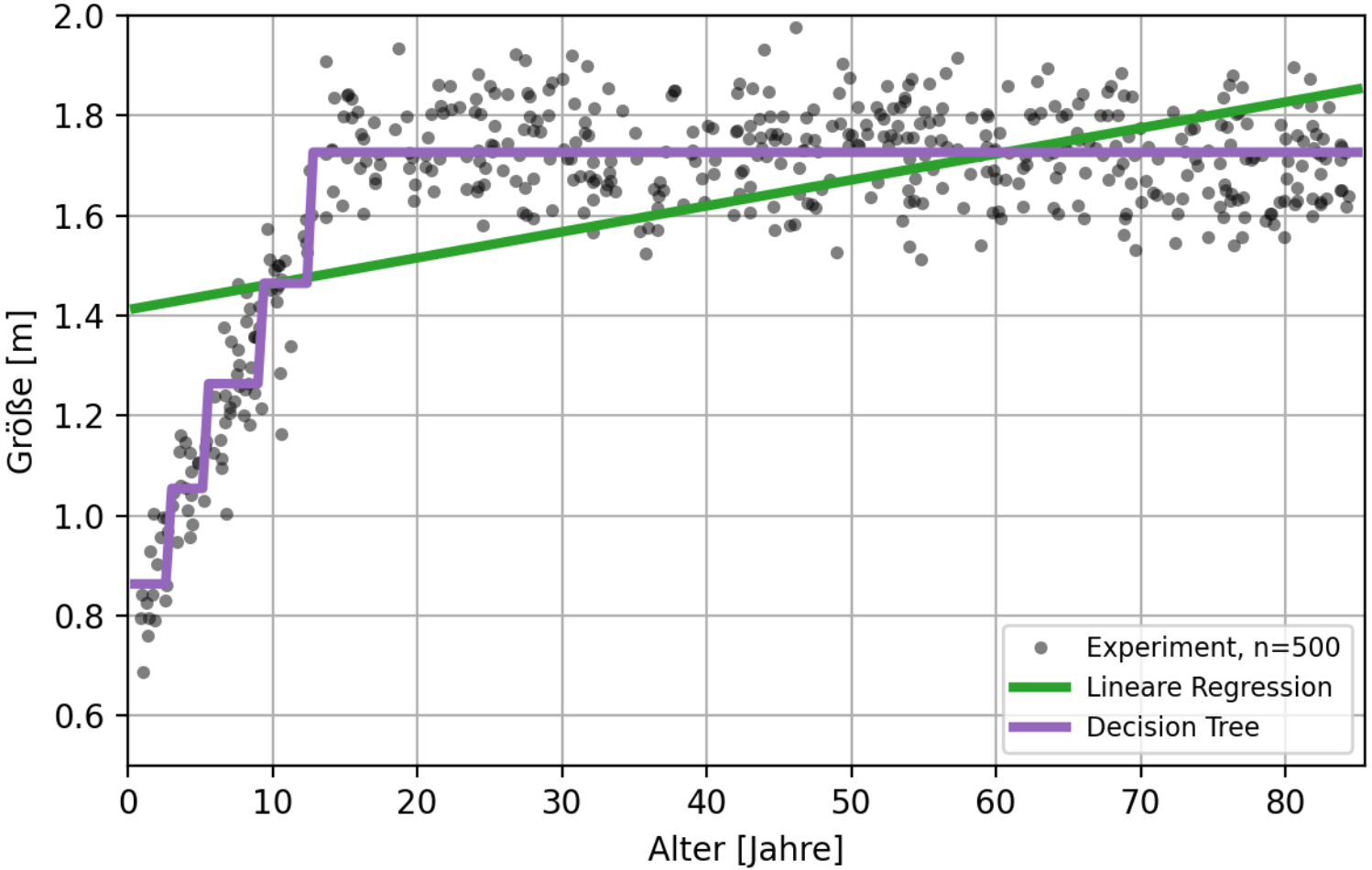
Überblick Regressoren



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Algorithmus Schmiede
Data Science | Numerik | Physik ©

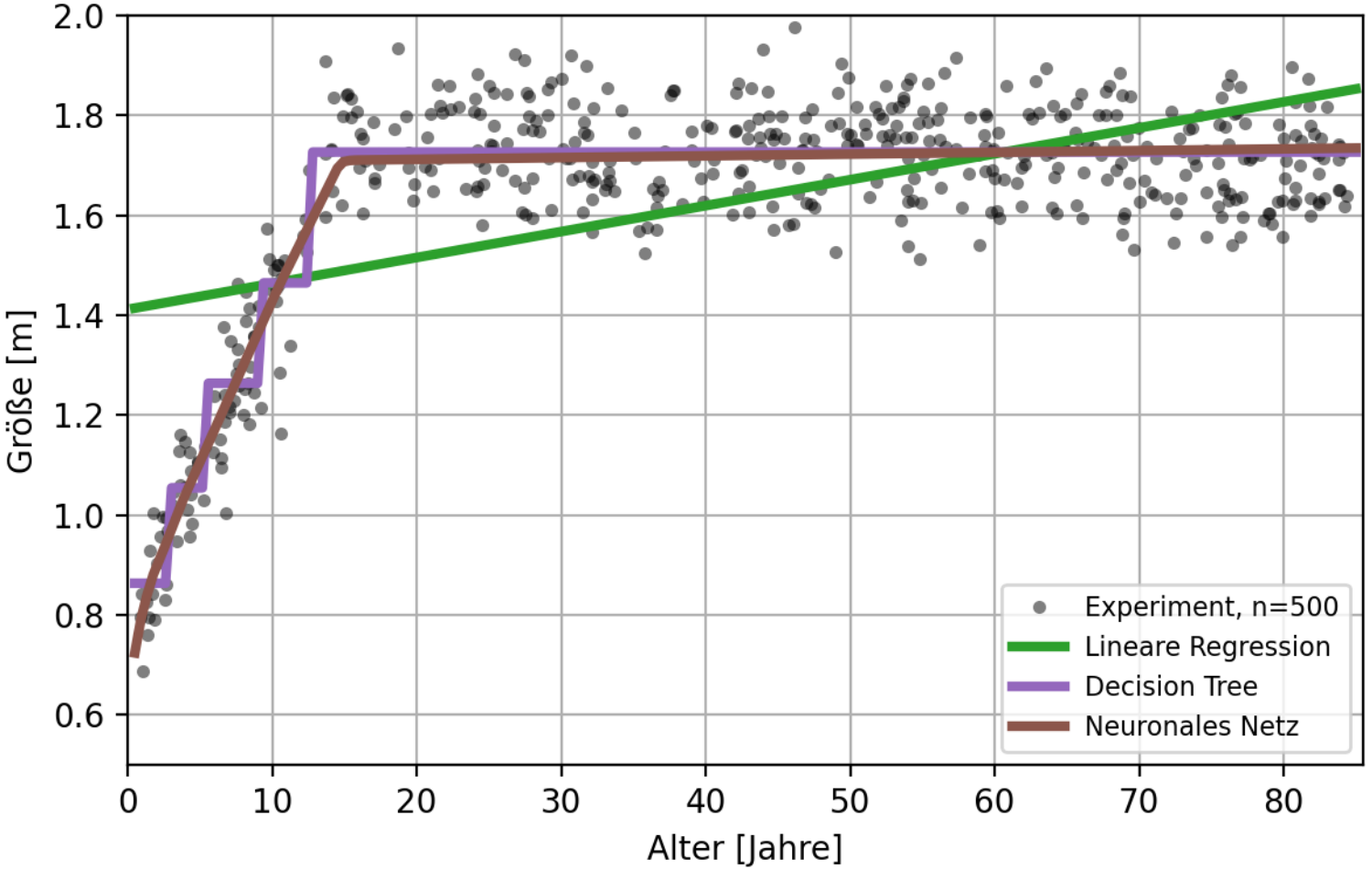
Überblick Regressoren



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

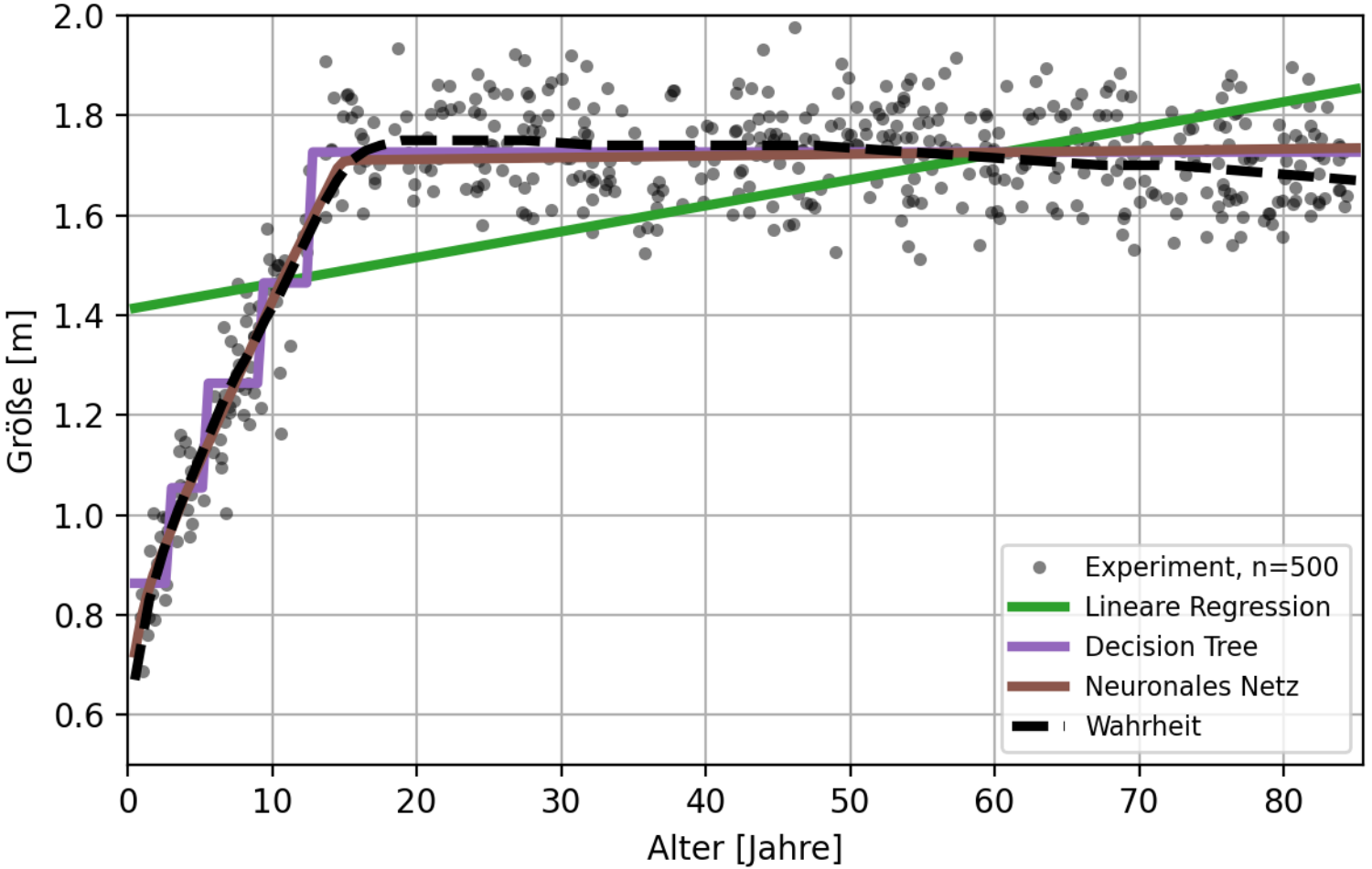
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Überblick Regressoren



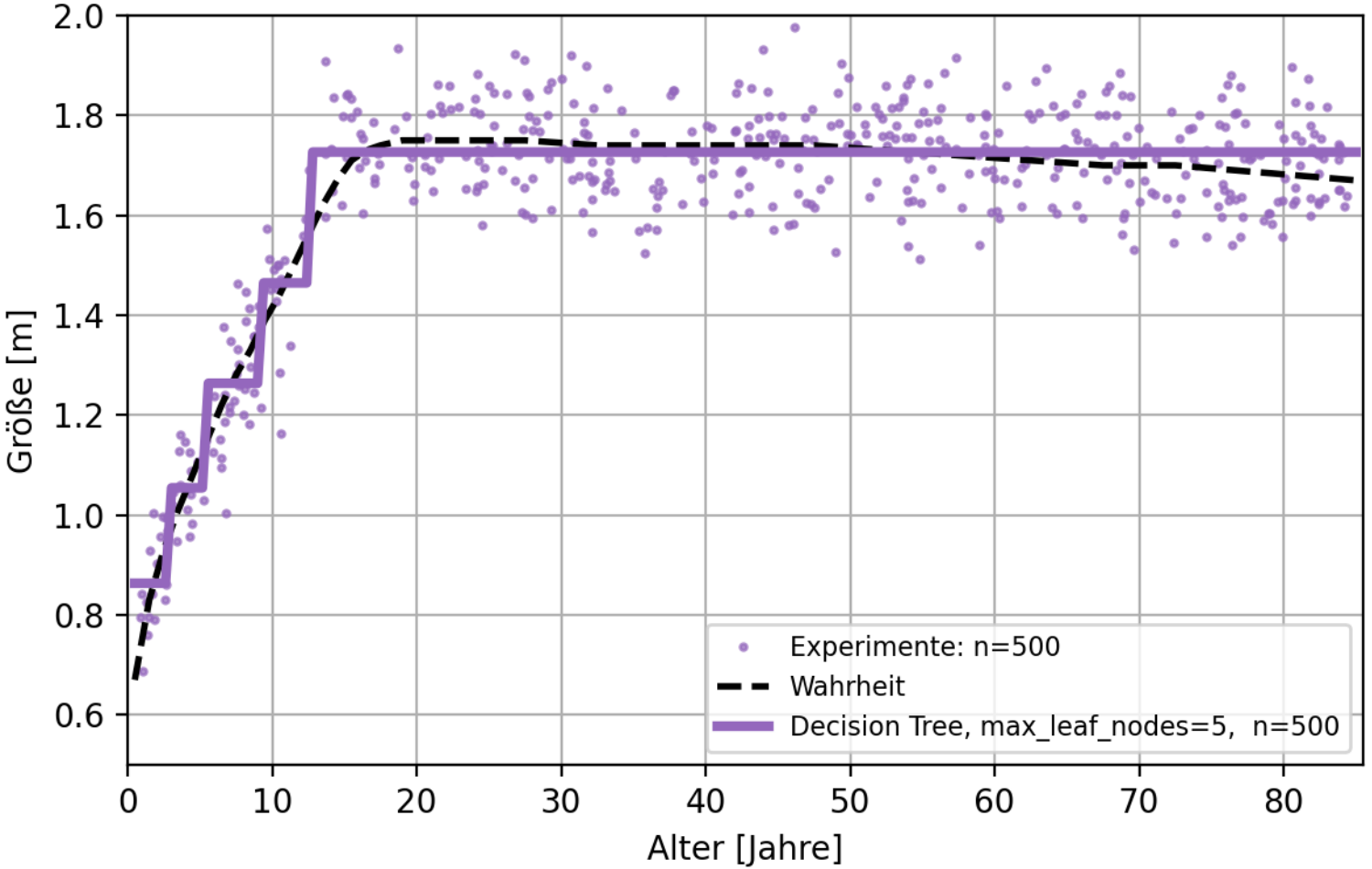
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Überblick Regressoren



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

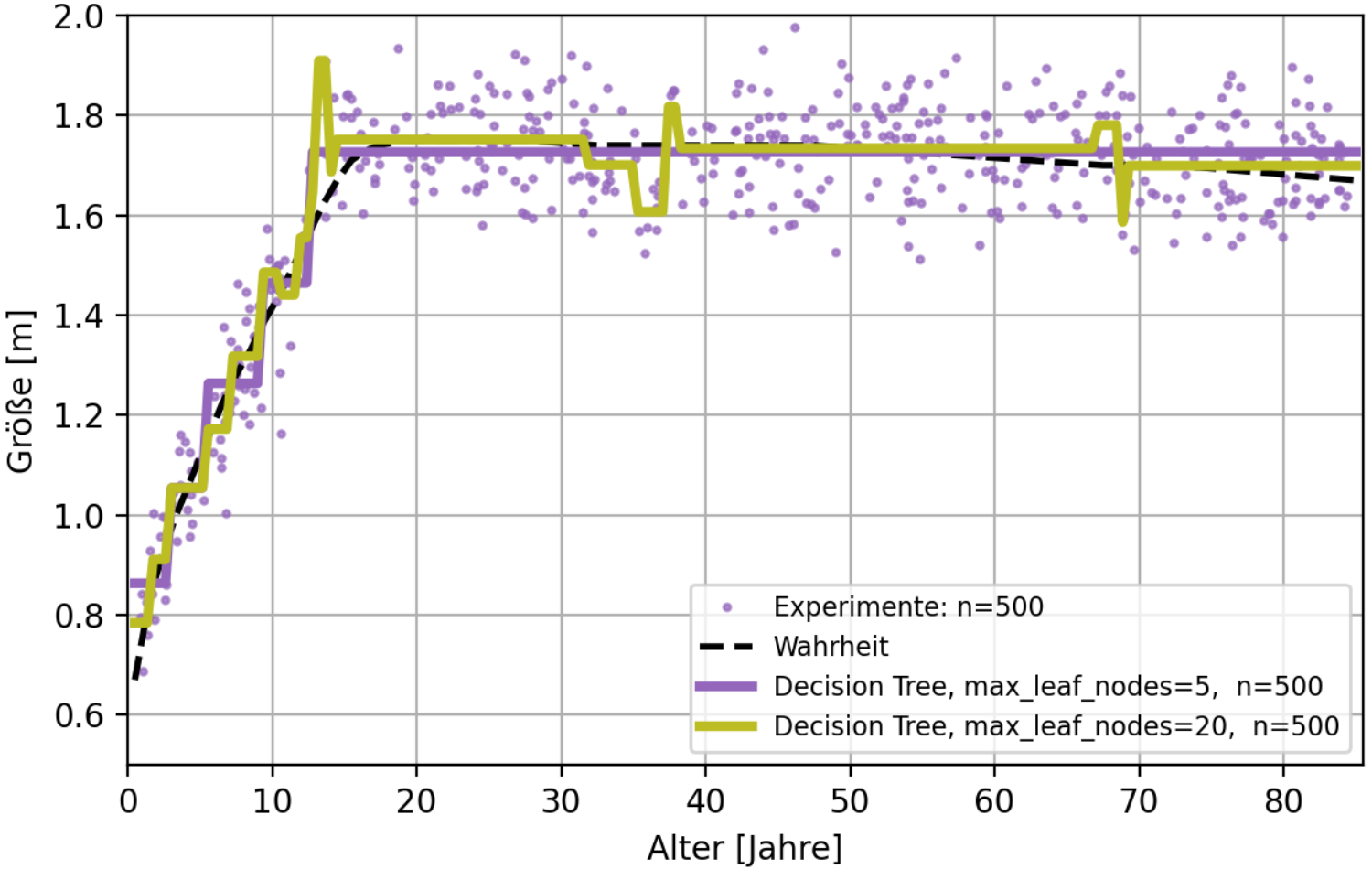
Decision Tree



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

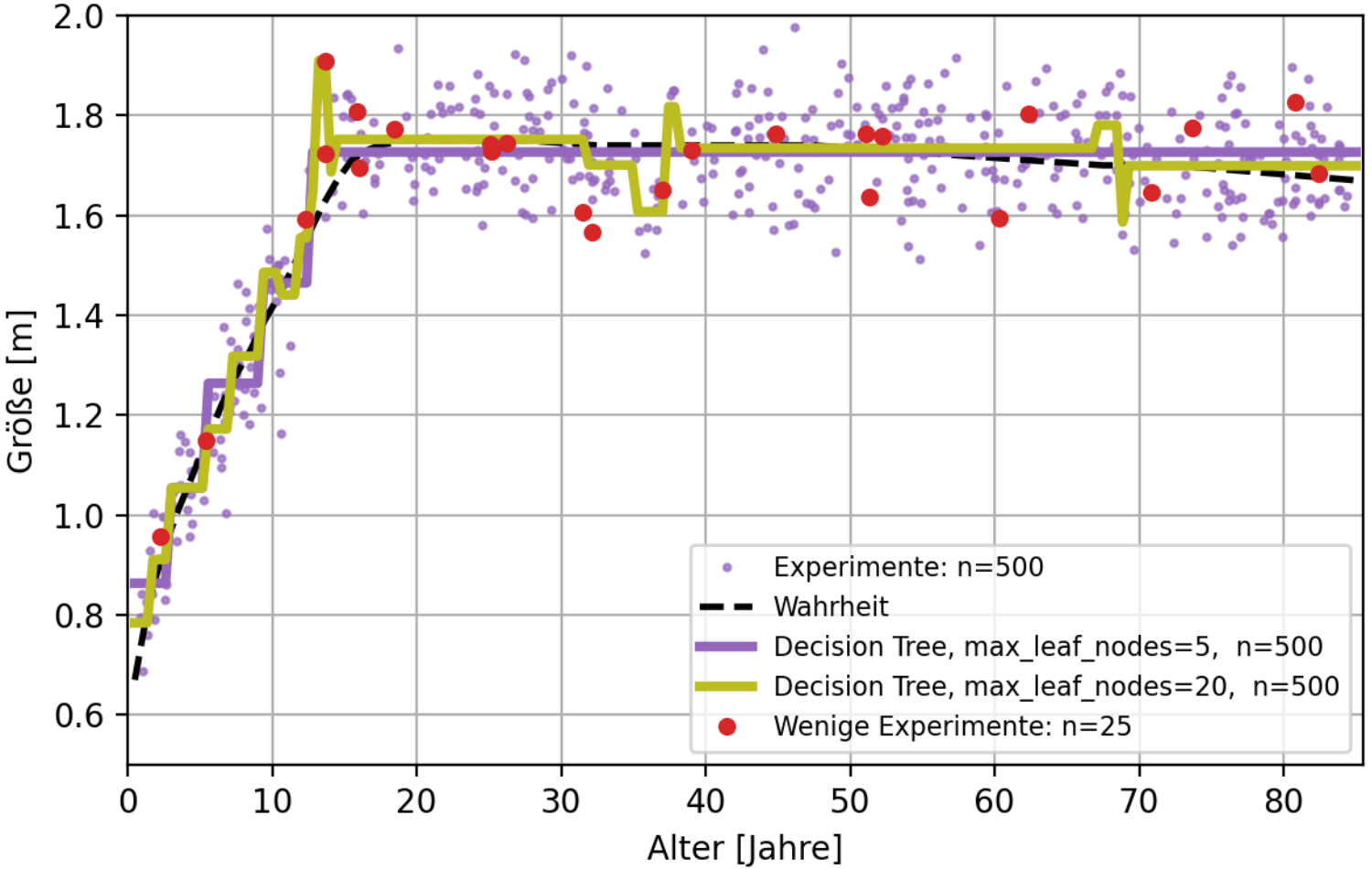
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Decision Tree



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

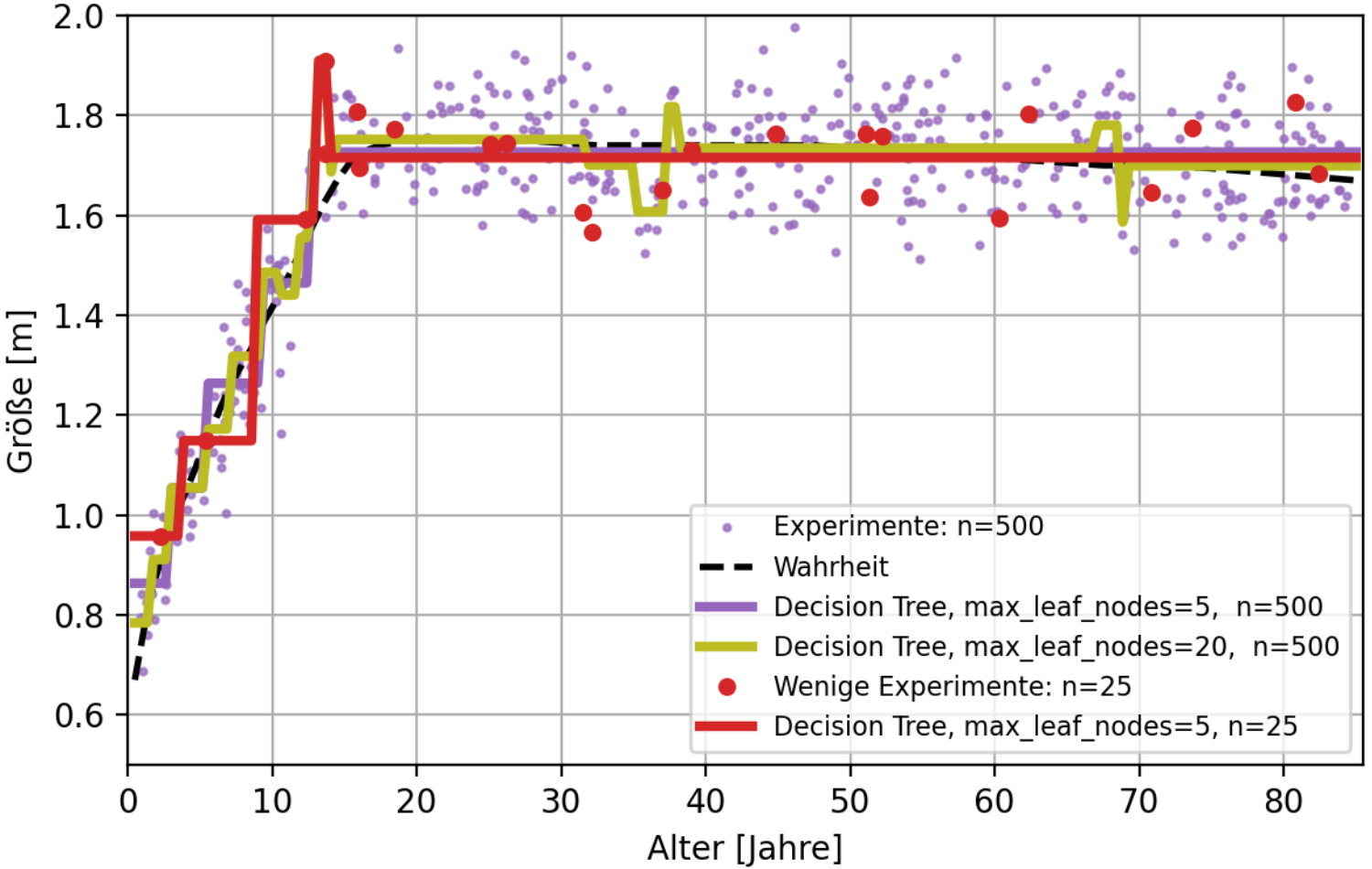
Decision Tree



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

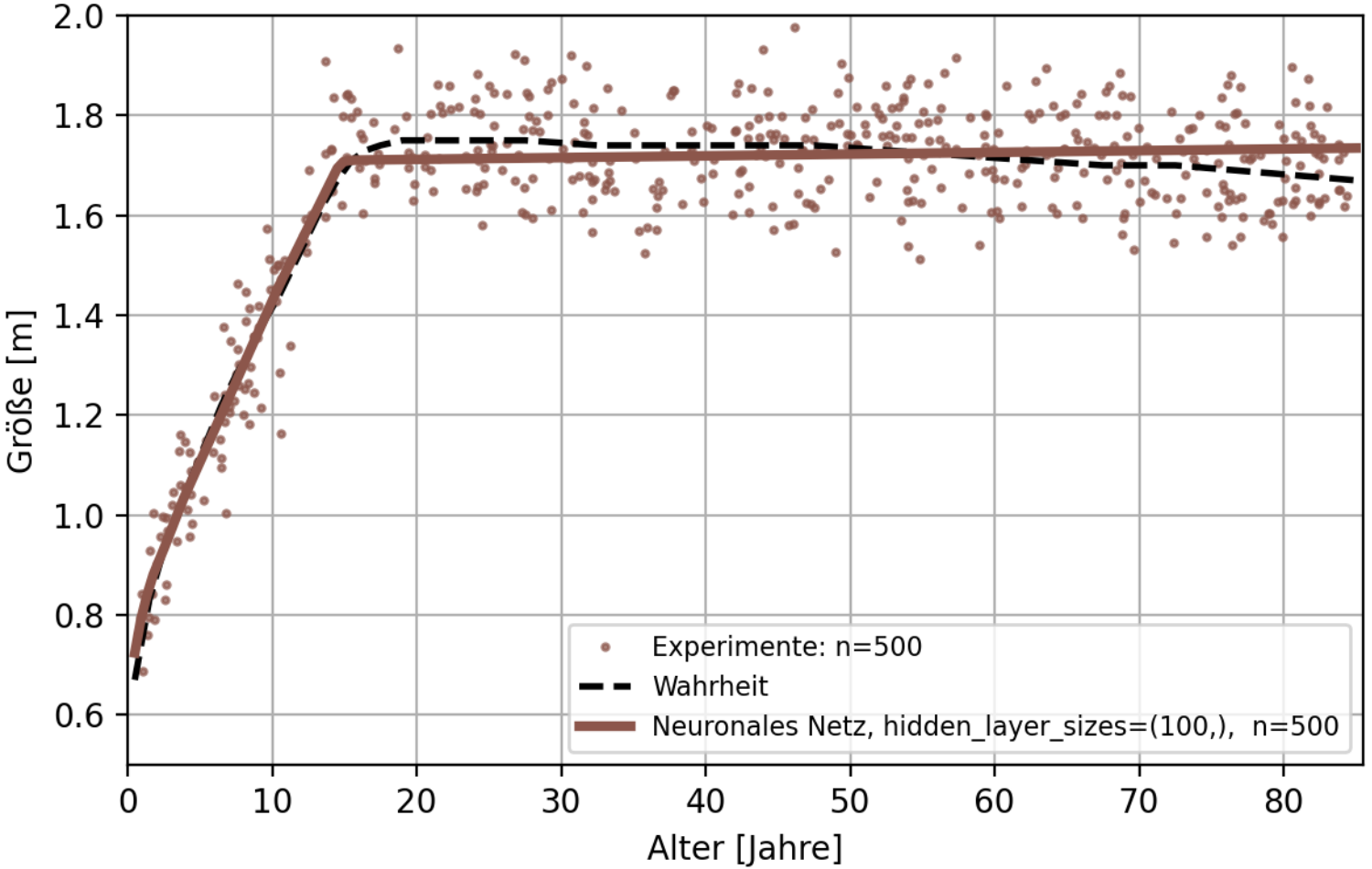
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Decision Tree



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

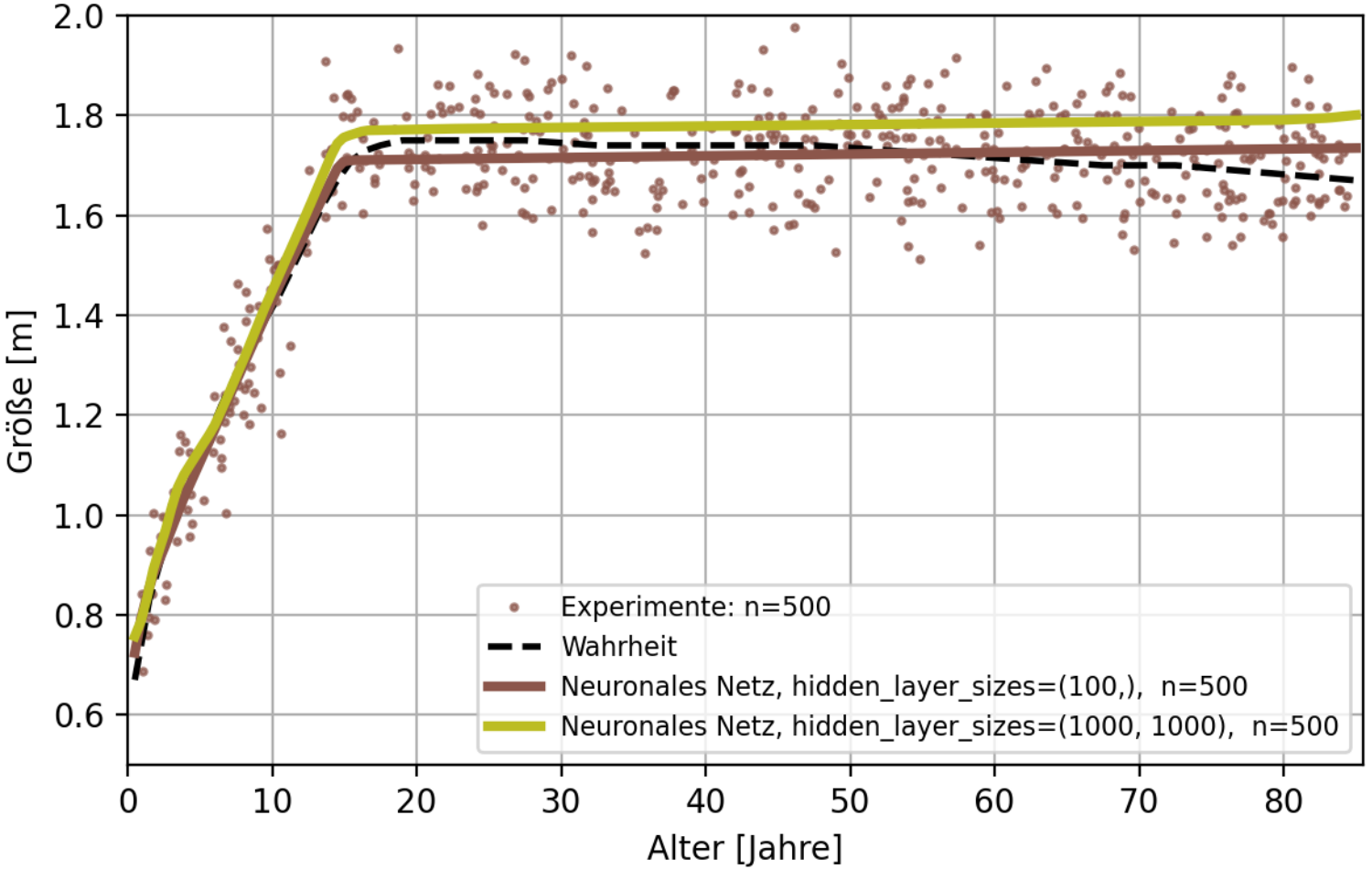
Neuronales Netz



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Algorithmus Schmiede
Data Science | Numerik | Physik ©

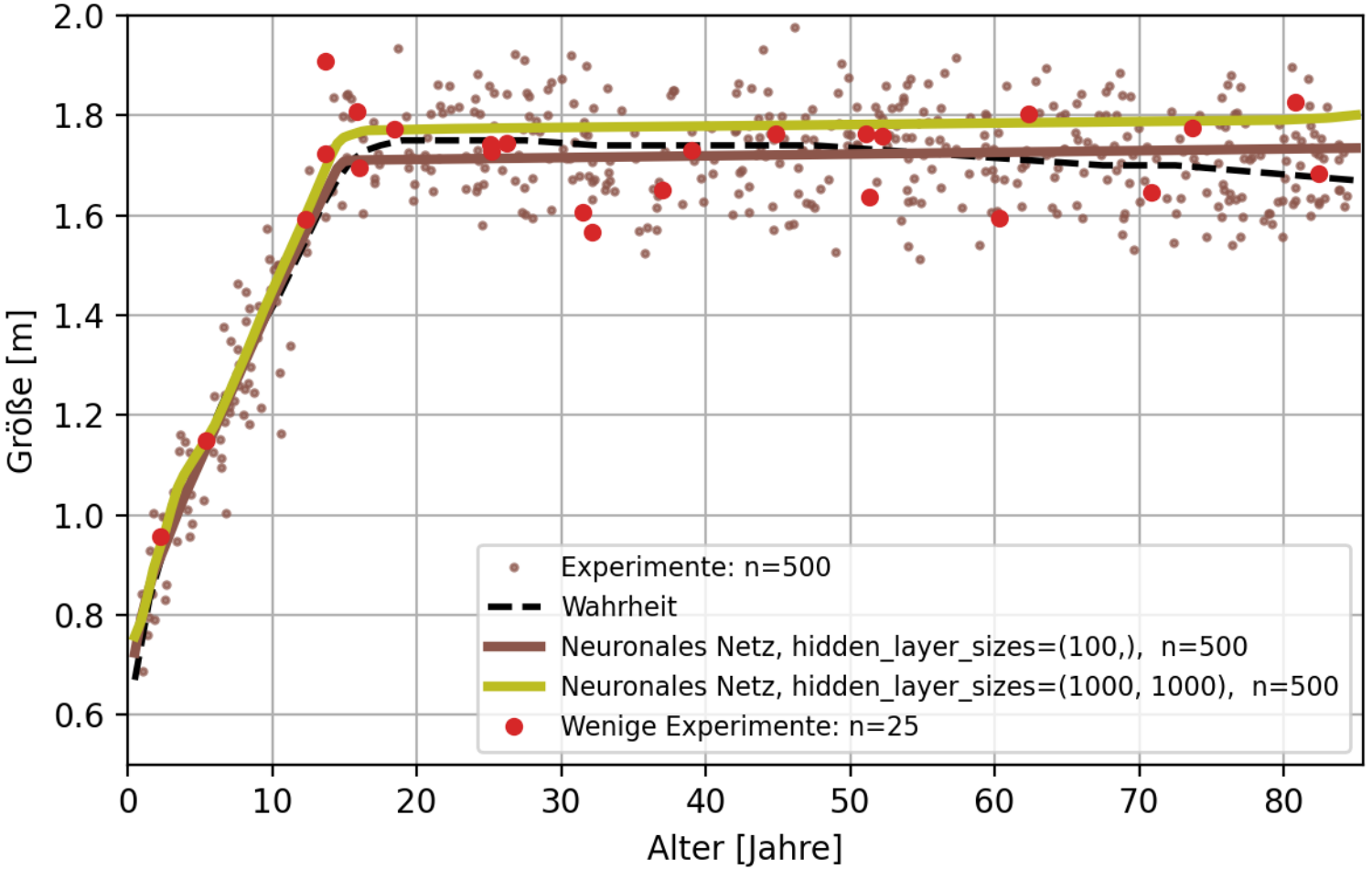
Neuronales Netz



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Algorithmus Schmiede
Data Science | Numerik | Physik ©

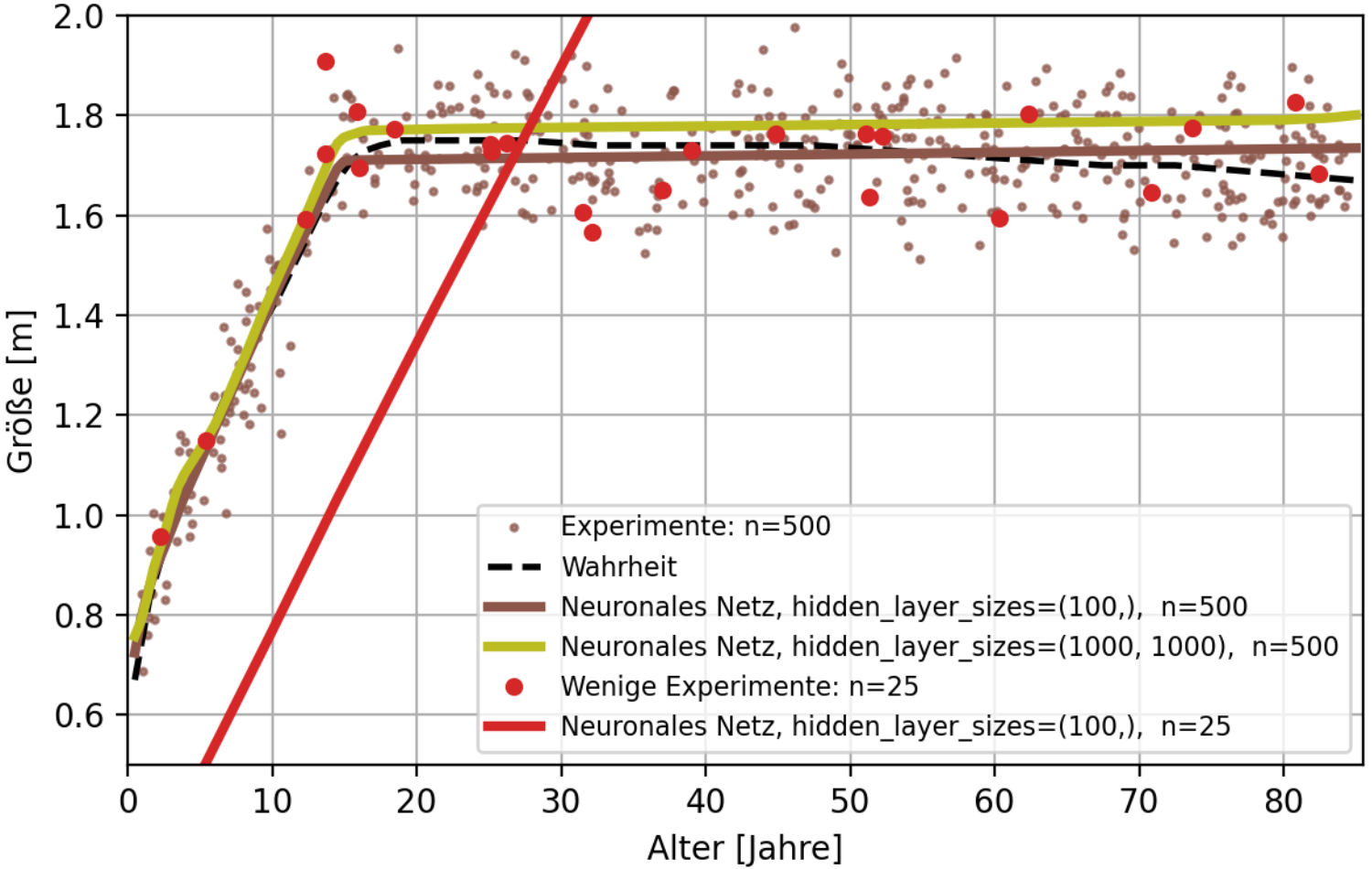
Neuronales Netz



Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

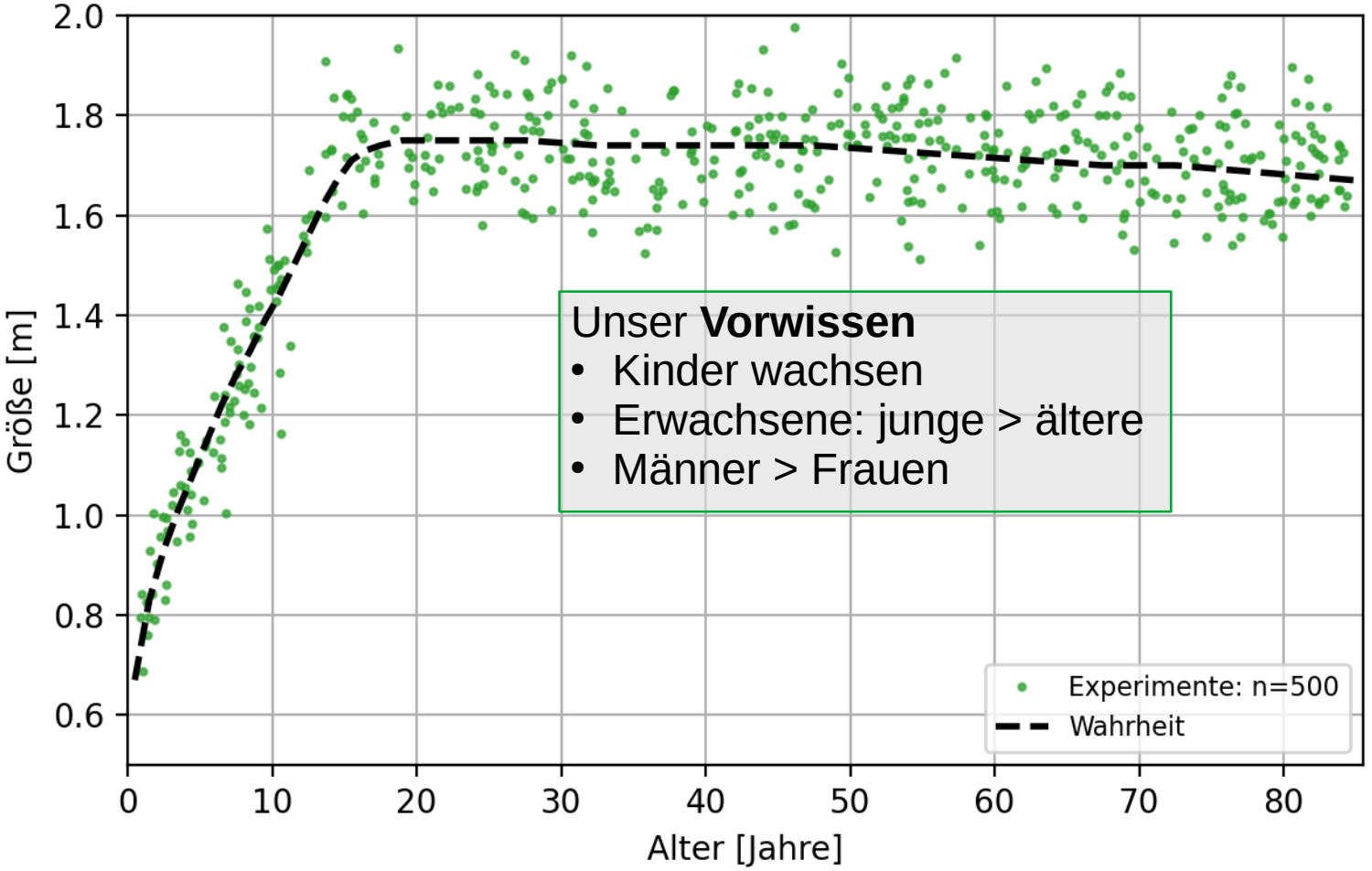
Algorithmus Schmiede
Data Science | Numerik | Physik ©

Neuronales Netz



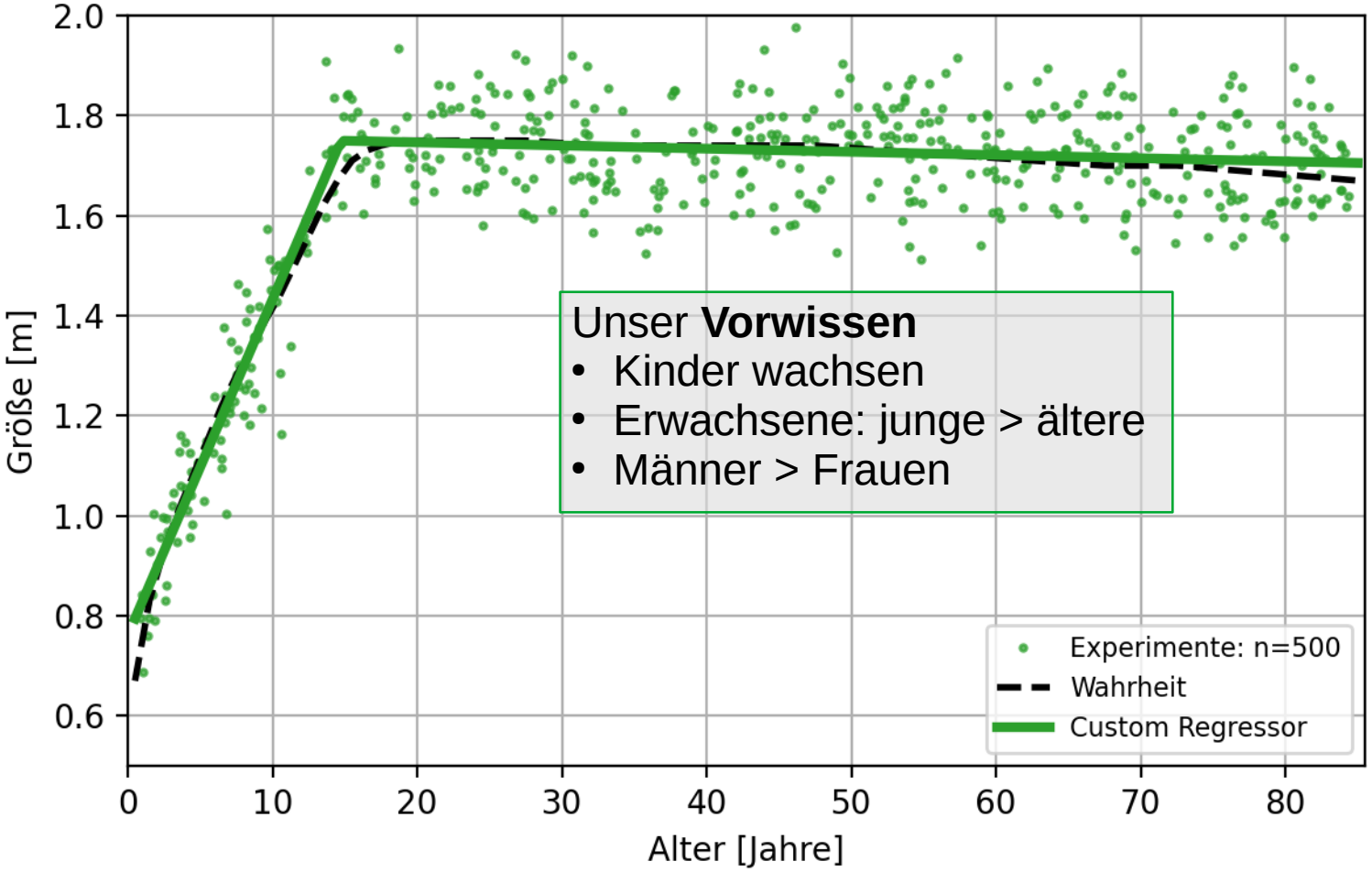
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Individuallösung



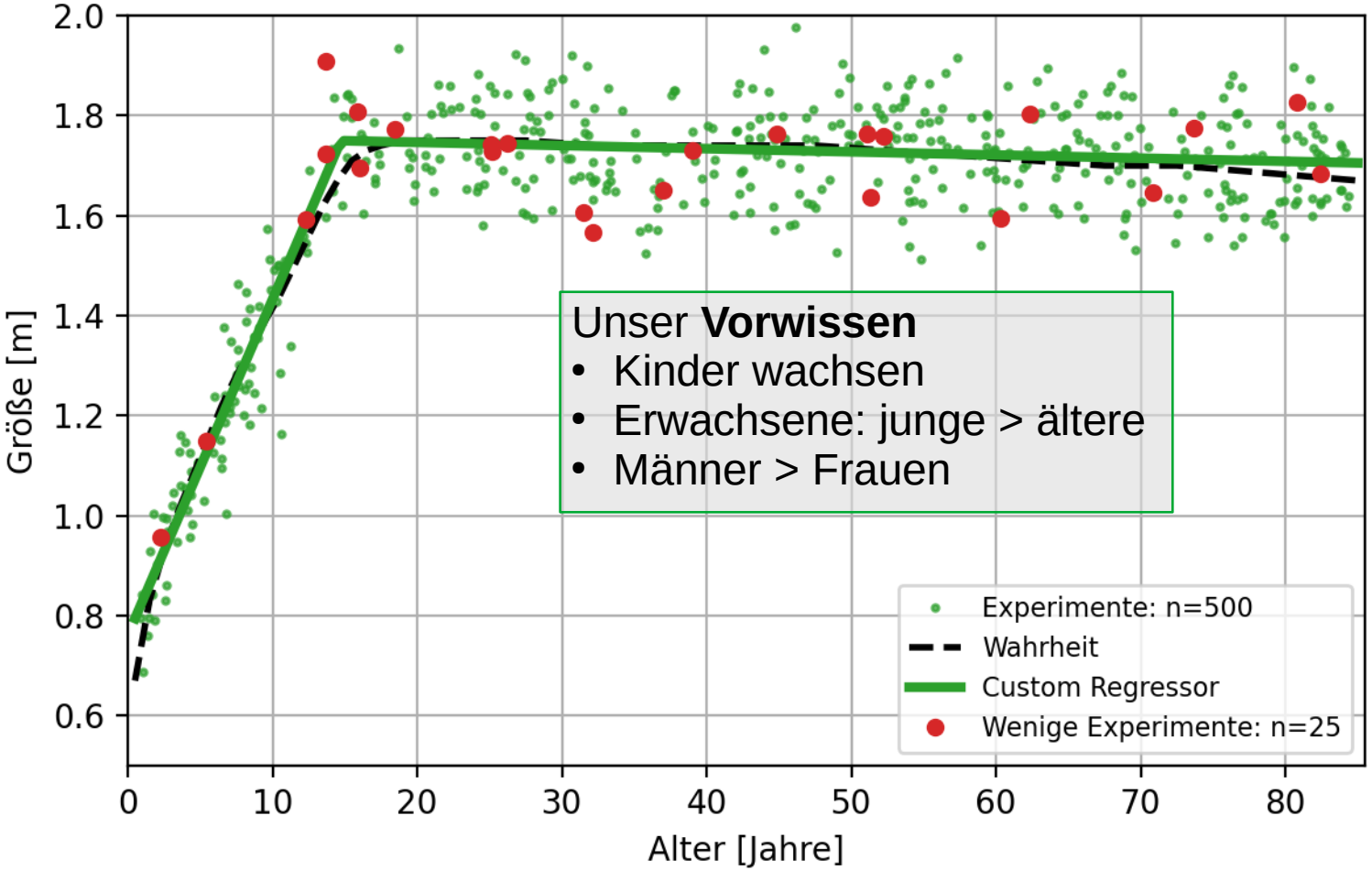
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Individuallösung



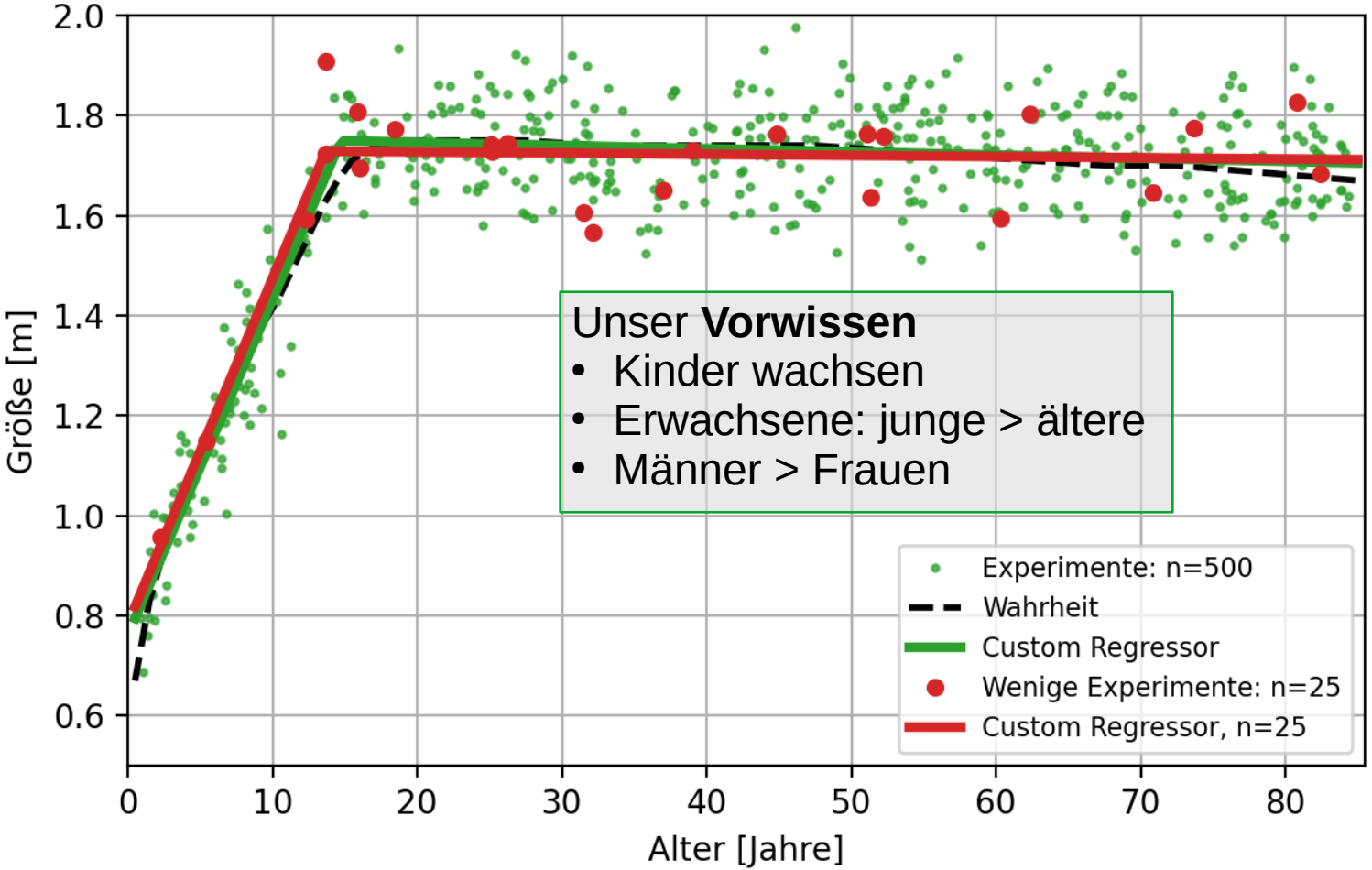
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Individuallösung



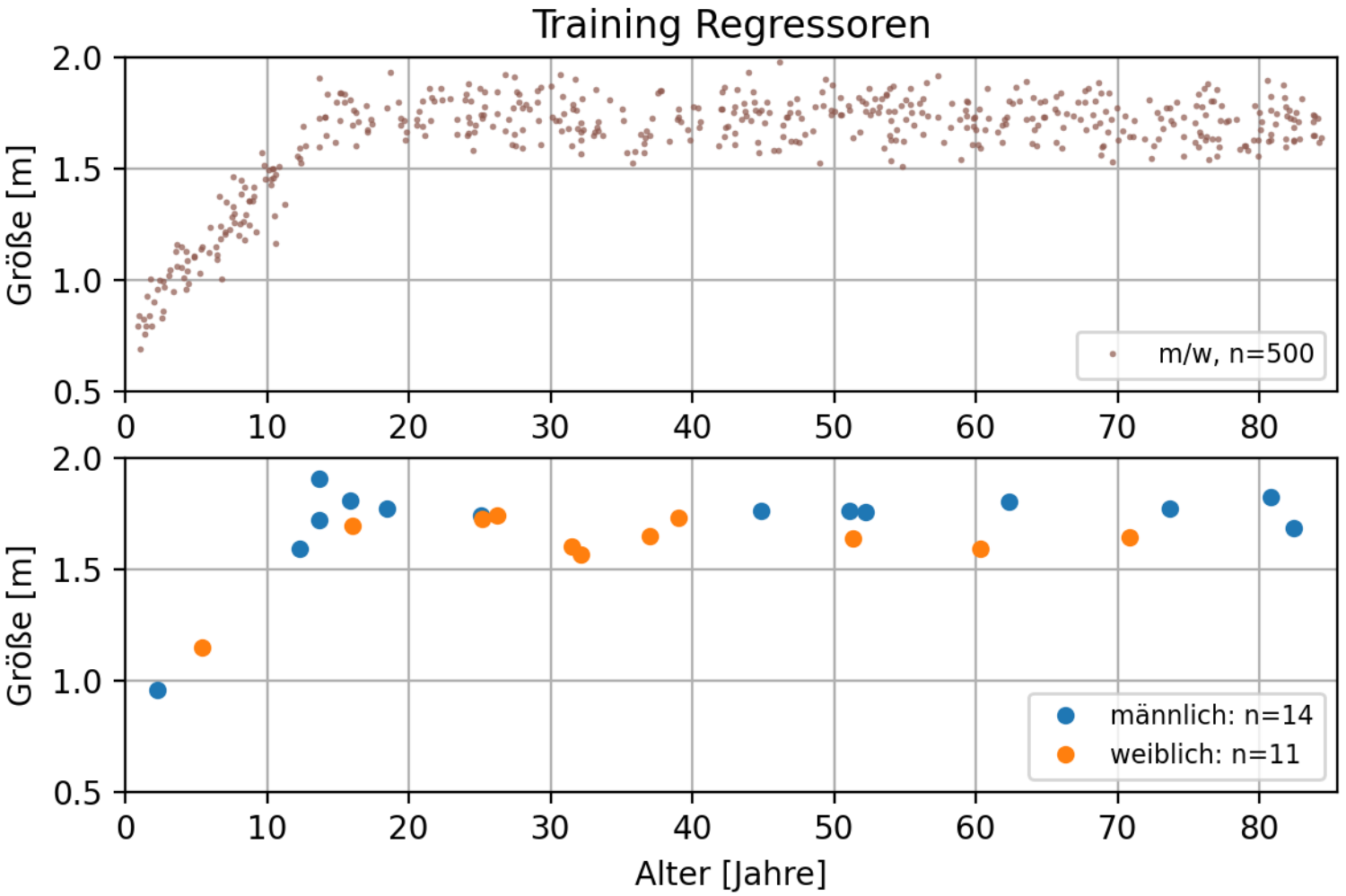
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Individuallösung



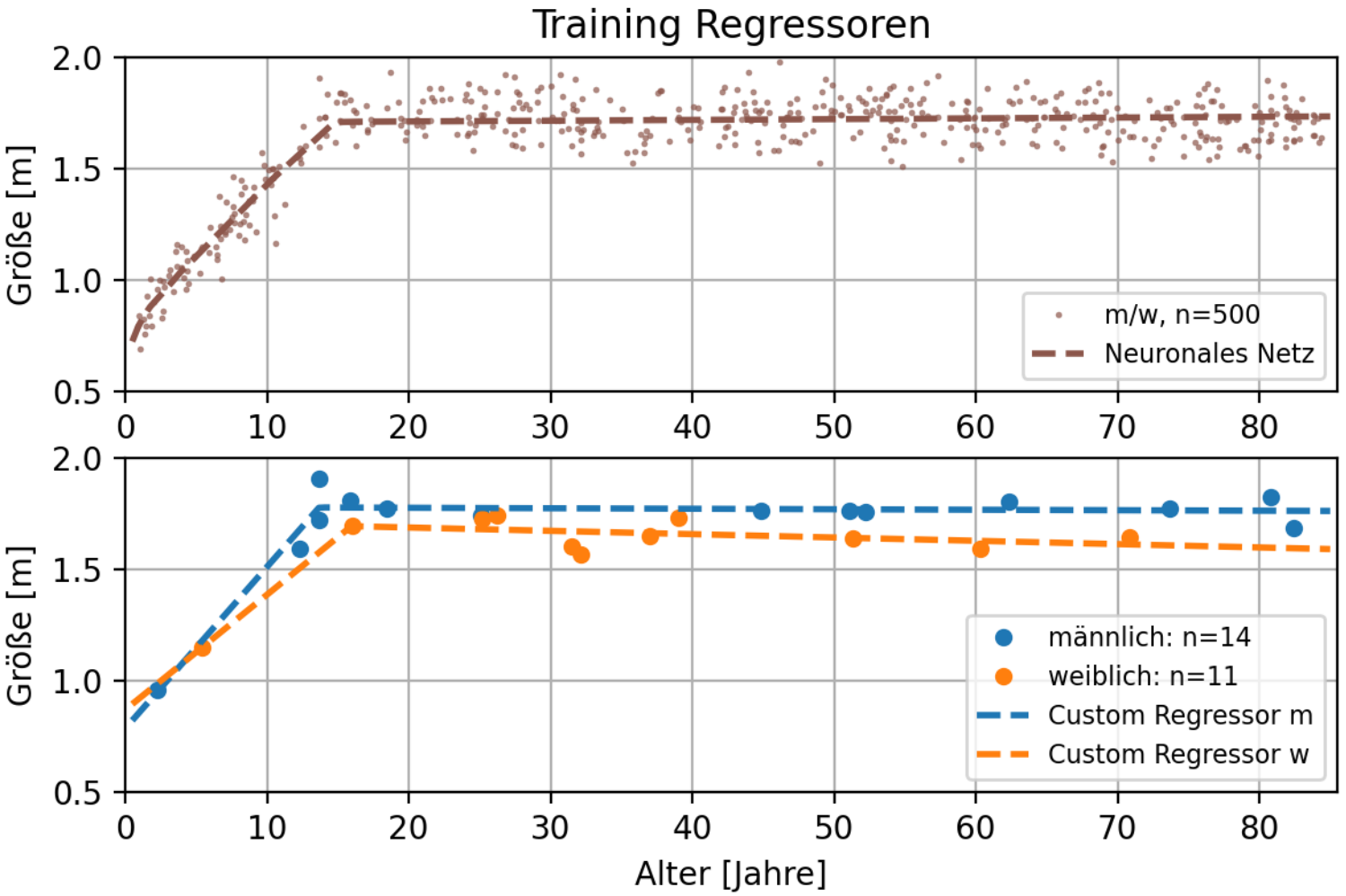
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Benchmark



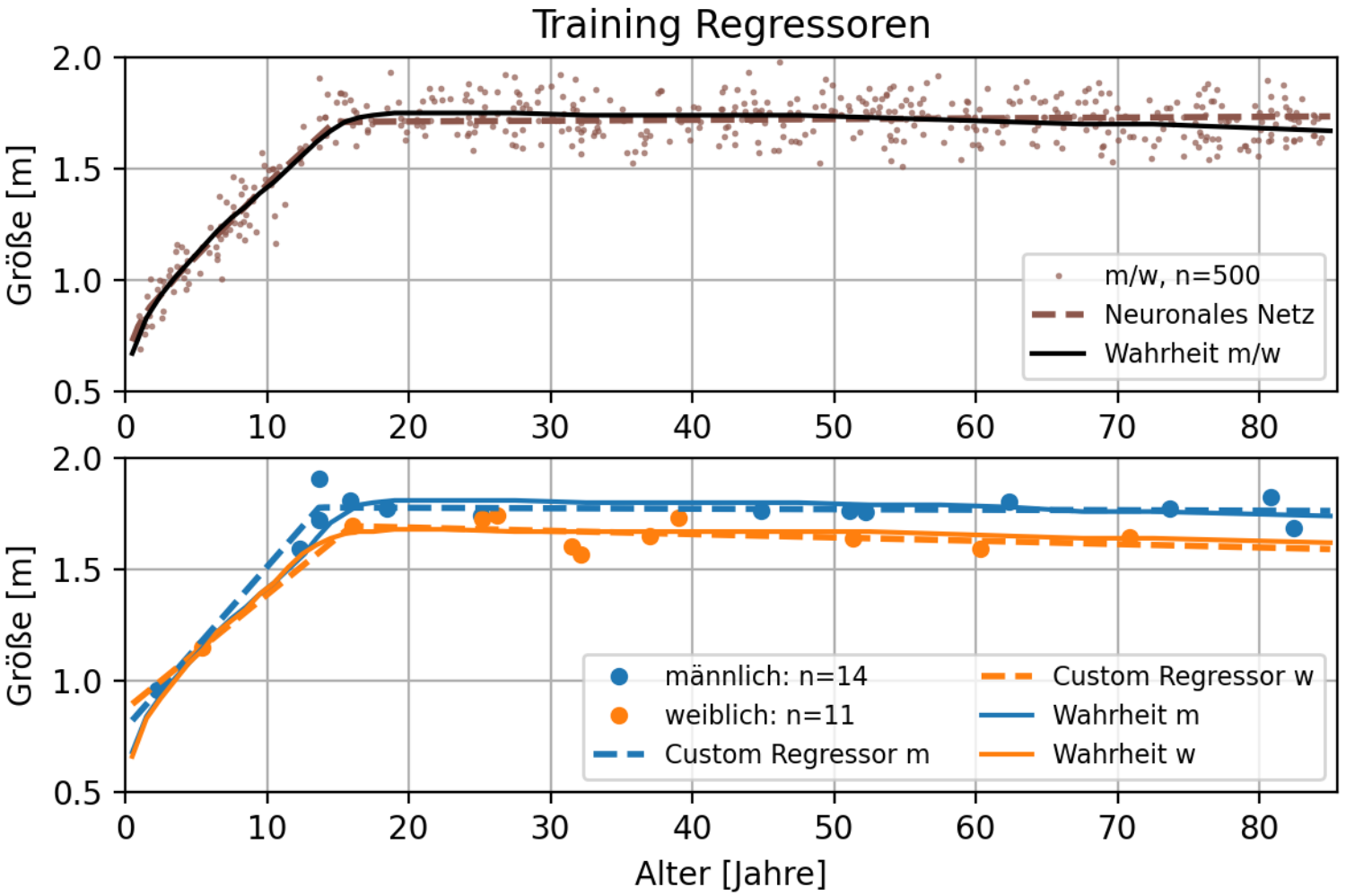
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Benchmark



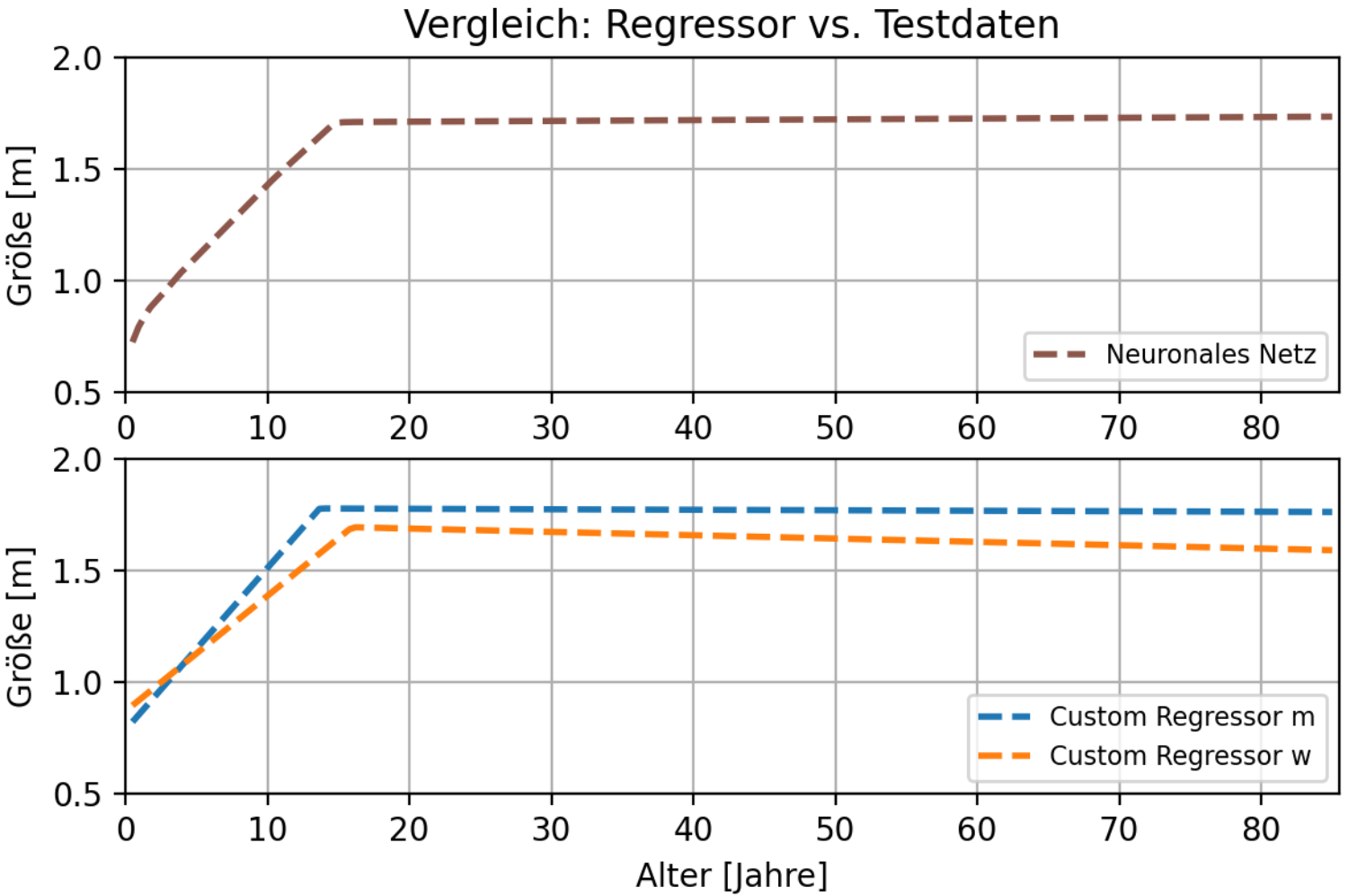
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Benchmark



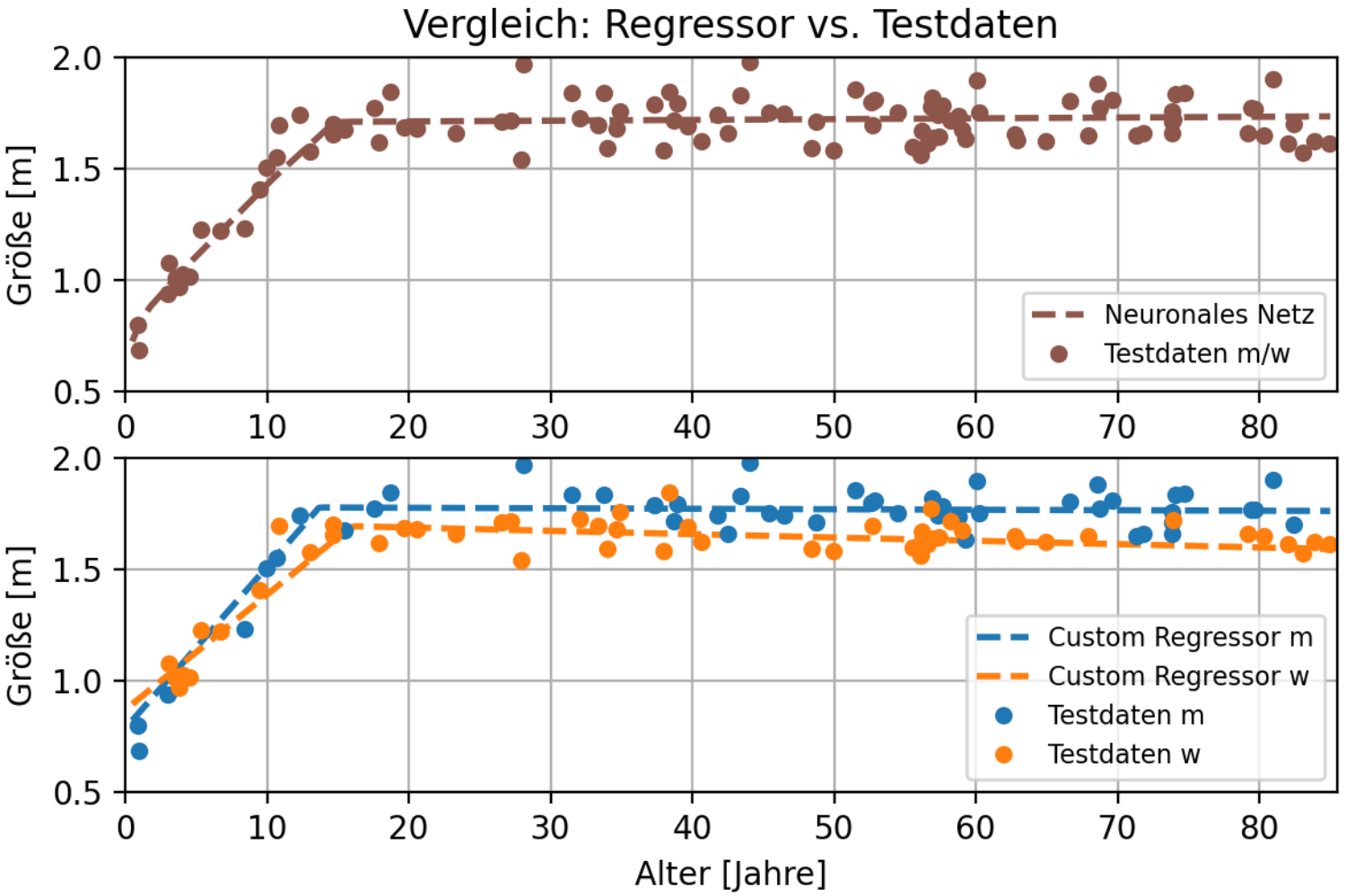
Quelle:
Gesundheitsberichterstattung der Bundes
www.gbe-bund.de

Benchmark



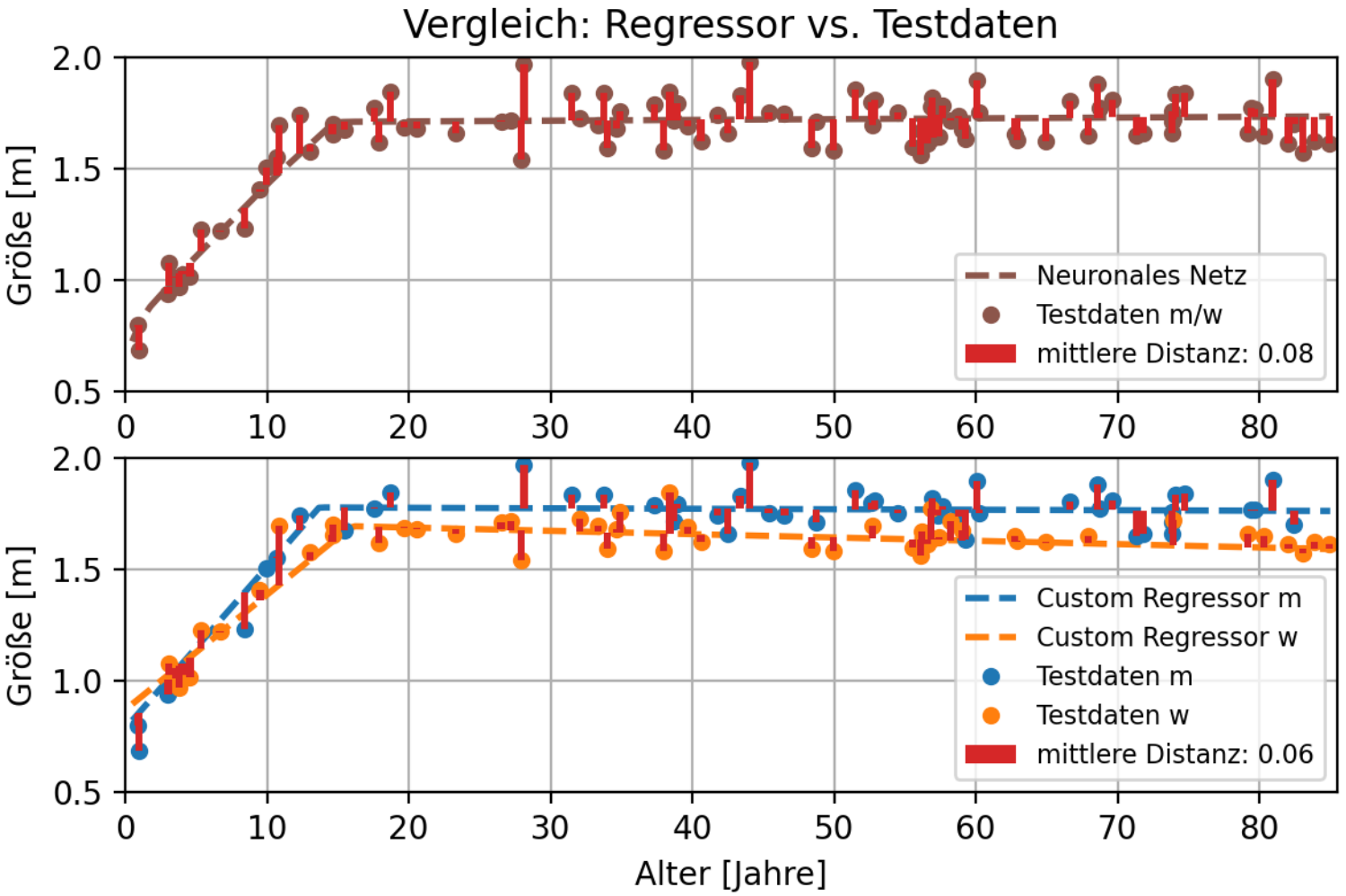
Data Science
Lösungen
via Wettbewerb:
www.kaggle.com

Benchmark



Data Science
Lösungen
via Wettbewerb:
www.kaggle.com

Benchmark



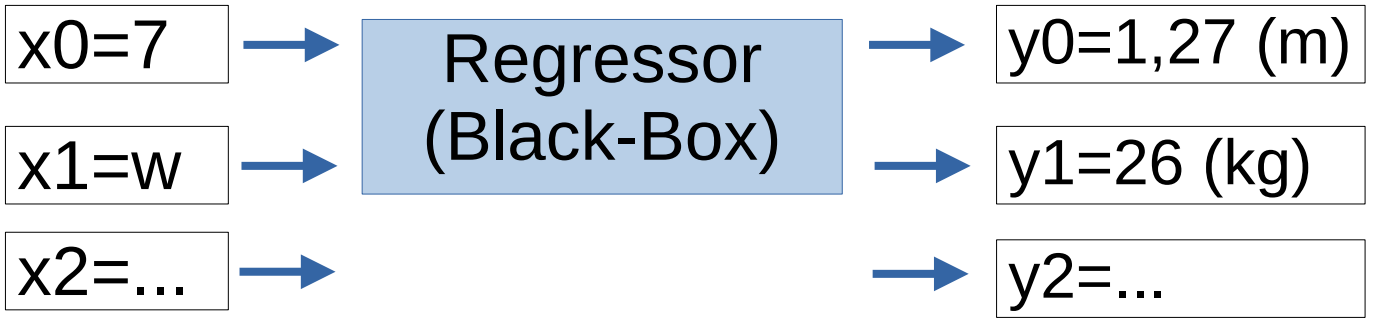
Data Science
Lösungen
via Wettbewerb:
www.kaggle.com

Zwischenfazit

Neuronale Netze sind nicht die Universallösung

	komplexes Machine Learning Model	Lösung basierend auf Systemverständnis
Entwicklungsaufwand	niedrig	hoch
Bedarf an Trainingsdaten	hoch	niedrig
Wiederverwertbarkeit	niedrig	hoch
Vorhersageverlässlichkeit	allgemein: gut selten: Totalversagen	sehr gut
Unschlagbar bei	Bild, Ton, Video, Ästhetik	Extreme Verlässlichkeit

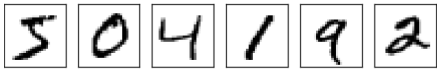
Verallgemeinerung



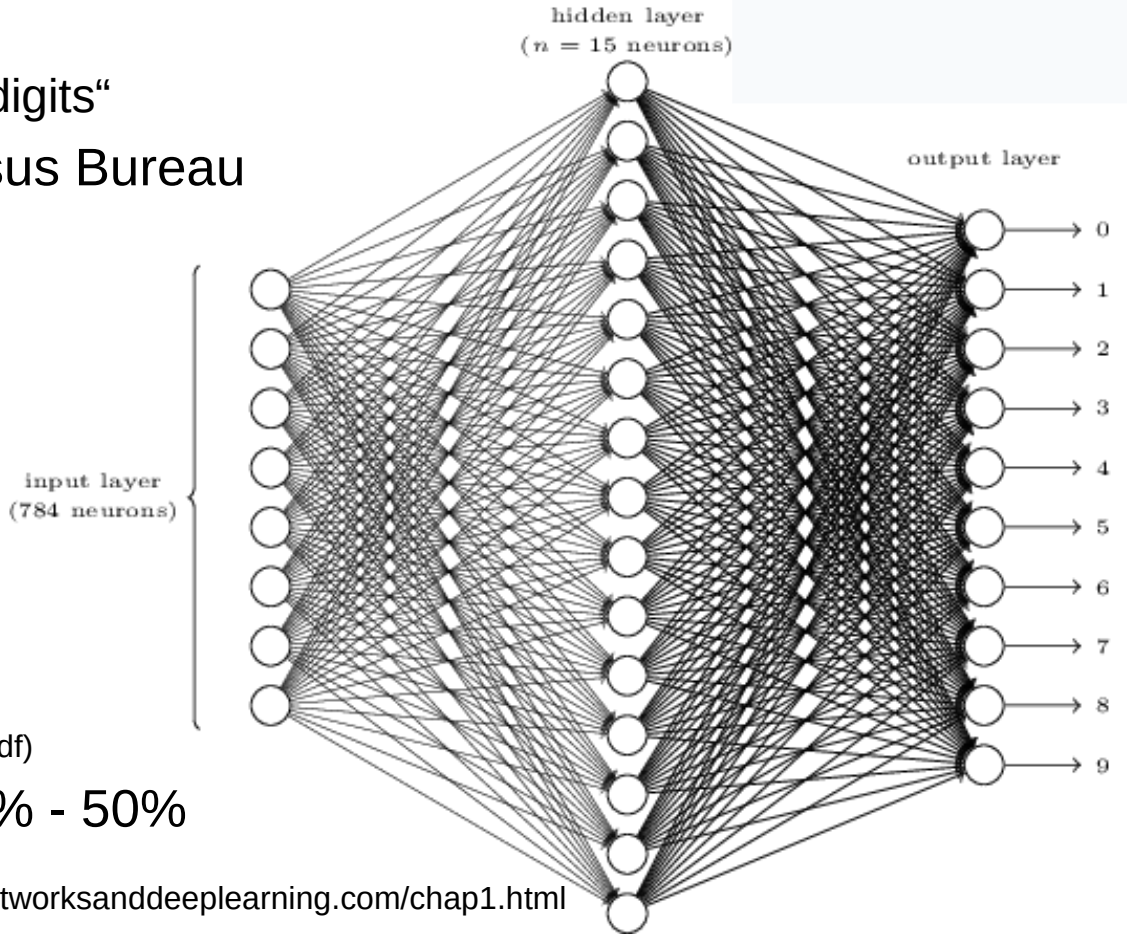
Beipiele:

- Aus Gewicht (x0), aktuellem Ort (x1, x2, x3), Geschwindigkeit (x4, x5, x6) und Windstärke (x7, x8, x9) von brennenden Spänen
  ... sage Auftreffpunkt (y0, y1, y2) und Streuung (y3, y4, y5) vorher.
- Aus Temperatur (x0), Lackart (x1), Untergrundmaterial (x2) und Zeit (x3)
  ... sage Trocknungsgrad (y0) vorher.
- Aus den Ergebnissen der Produktprüfung (x0, ...)
  ... sage Lebensdauer (y0) vorher.

NN @ Handschrifterkennung



- NIST Data-Set „handwritten digits“
- 250 Autoren aus US Census Bureau
- Training: **60 000 Ziffern**
- Test-Daten: 10 000 Ziffern

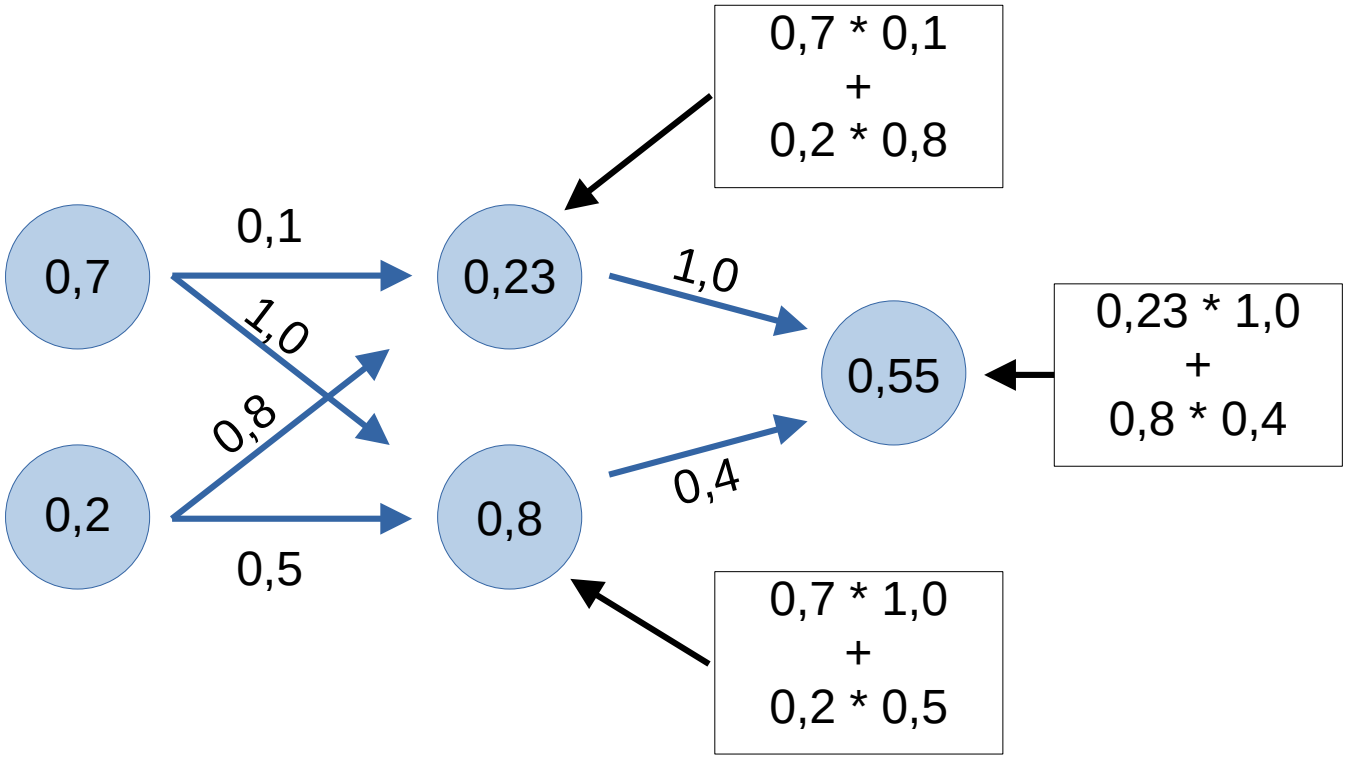


- Genauigkeit: 99,79%
(https://proceedings.mlr.press/v28/wan13.pdf)
- Typische Genauigkeit: 20% - 50%

http://neuralnetworksanddeeplearning.com/chap1.html

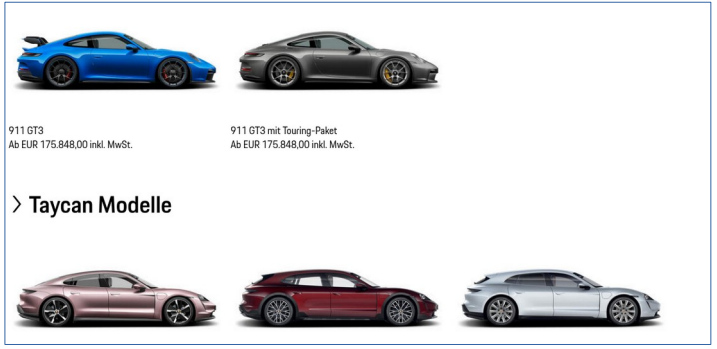
Neuronales Netz

Funktionsweise (vereinfacht):

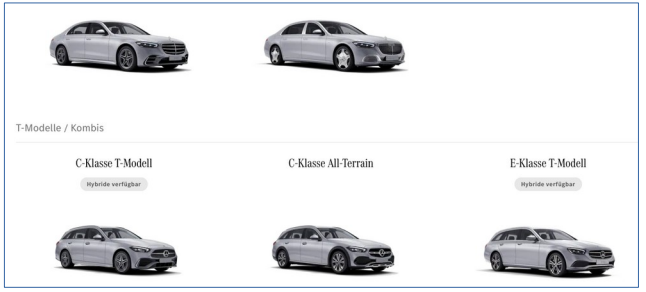


Aufgabe: Finde Automarke

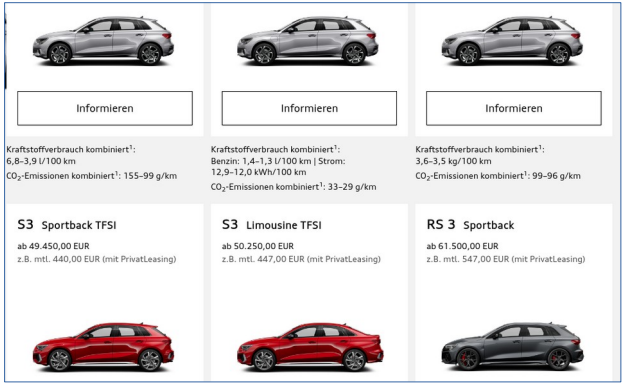
https://www.porsche.com/germany/models/



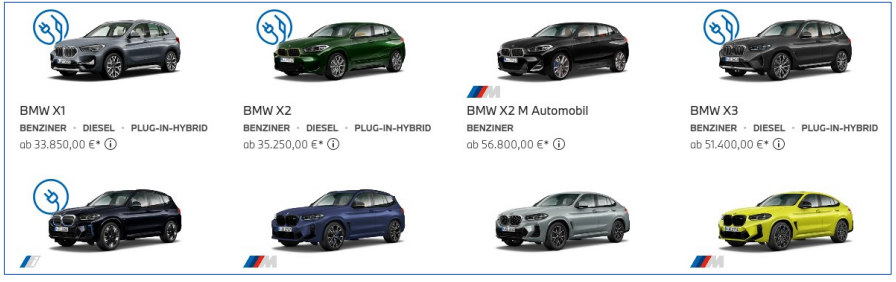
https://www.mercedes-benz.de/passengercars/models.html



https://www.audi.de/de/brand/de/neuwagen.html



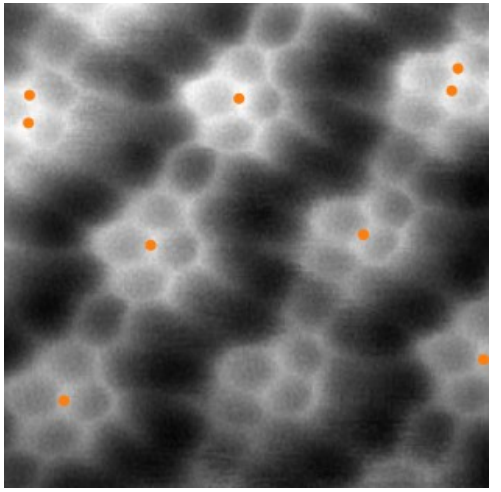
https://www.bmw.de/de/neufahrzeuge.html



Algorithmus Schmiede
Data Science | Numerik | Physik ©

Praxisbeispiel: Neuronale Netze in der Produktion

- Messverfahren:
  kann nur Höhe erfassen, Messen der Farbe ist „physikalisch unmöglich“
- 2-tes Messsystem zum erfassen von Farbe ist angedacht
- Verlauf des Höhenprofils und Rauschen ermöglicht
  an einzelnen Punkten die Errechnung der Farbe via NN
- Farbinformation muss an diesen Punkte durch irgendeine Konstellation von Messartfakten vorliegen
  z.B. Lichtstreuung, Erwärmung, Ausdehnung, ...
- Farbverlauf im Restlichen Bild wird durch Interpolation rekonstruiert
- Funktionsweise ist bis heute nicht klar
- Verlässlichkeit von 80% deutlich höher als notwendig



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4050271/

Praktisches Vorgehen

Projektbausteine

- math. Modelle
- Experimente
- Softwaremodule

Beispiel: Optische Inspektion

- Objekterkennung via NN
- Objektprüfung via Algorithmus

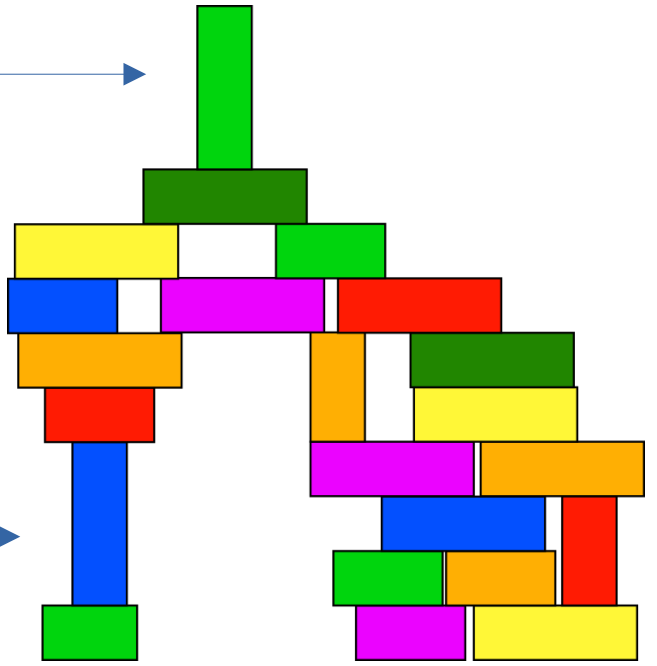
statt

- Objekterkennung + Prüfung via NN

Komplexes Machine Learning Model?

EVTL. OK

NICHT OK



Weitere Machine Learning Techniken

Natural Language Processing

- Letzte 100 Buchstaben → nächster Buchstabe

Reinforcement Learning:

- aktuelle Situation, Steuersignal → finaler Punktestand

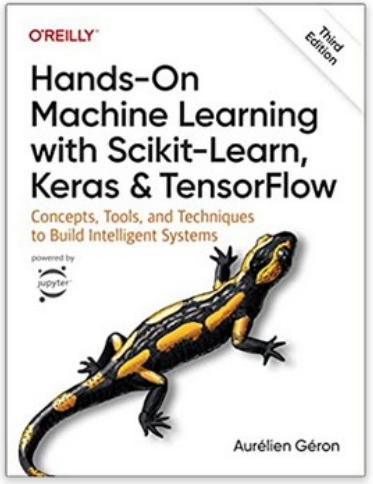
Clustering:

- Anomalie Detektion: Weit weg von jedem Cluster

Code Snippet

```python
if __name__ == "__main__":

    df_data = read_data()
    df_samples = create_sample_data(df_data, n=N_SAMPLES)

    # data
    X = df_samples[[AGE]].to_numpy()  # X.shape=(500, 1)
    Y = df_samples[[HEIGHT]].to_numpy()  # Y.ravel().shape=(500,)

    # create ML model
    from sklearn.neural_network import MLPRegressor
    hidden_layer_sizes = (100,)
    rgr_mlp = MLPRegressor(random_state=1, max_iter=2000, hidden_layer_sizes=hidden_layer_sizes).fit(X, Y.ravel())

    # forecast
    xs_forecast = np.array([[7], [18], [45]])  # xs_forecast=[[7], [18], [45]]
    ys_forecast = rgr_mlp.predict(xs_forecast)  # ys_forecast=[1.23440654 1.71132157 1.72072799]

    print(f"X.shape={X.shape}")
    print(f"Y.ravel().shape={Y.ravel().shape}")
    print(f"xs_forecast={xs_forecast}")
    print(f"ys_forecast={ys_forecast}")
```

Buchempfehlung



Dieses Bild anzeigen

Dem Autor folgen

Aurélien Gé-ron

Folgen

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems Taschenbuch – 8. November 2022

Englisch Ausgabe | von Aurélien Géron (Autor)

4,7 ★★★★½ 31 Sternebewertungen

Alle Formate und Editionen anzeigen

Kindle	Taschenbuch
42,66 €	**60,89 €**
Lies mit **kostenfreier App**	3 Gebraucht ab 68,40 € 29 Neu ab 60,84 €

Möchten Sie Ihre Elektro- und Elektronikgeräte kostenlos recyceln? Mehr erfahren

Through a recent series of breakthroughs, deep learning has boosted the entire field of machine learning. Now, even programmers who know close to nothing about this technology can use simple, efficient tools to implement programs capable of learning from data. This bestselling book uses concrete examples, minimal theory, and production-ready Python frameworks (Scikit-Learn, Keras, and TensorFlow) to help you gain an intuitive understanding of the concepts and tools for building intelligent systems.
With this updated third edition, author Aurélien Géron explores a range of techniques, starting with simple linear regression and progressing to deep neural networks. Numerous code examples and exercises throughout the book help you apply what you've

Mehr lesen

Seitenzahl der Print-Ausgabe	Sprache	Herausgeber	Erscheinungstermin	Abmessungen	ISBN-10
834 Seiten	**Englisch**	**O'Reilly Media**	**8. November 2022**	**18.42 x 5.08 x 24.13 cm**	**1098125975**

Spektakuläre Videos

- Pilco, Card-Pole: Balancieren Lernen
  https://www.youtube.com/watch?v=XiigTGKZfks
- Boston Dynamics, BigDog: Roboterhund
  https://www.youtube.com/watch?v=cNZPRsrwumQ
- Rotationskräfte: Einrad fahrender Roboter
  https://www.youtube.com/watch?v=RqOYLWgq2QA
- YOLOv2: Personenerkennung umgehen
  https://www.youtube.com/watch?v=MIbFvK2S9g8
  https://www.youtube.com/watch?v=6SXmOhTSh8Q

Für Techies:

- DeepMind, Agent57: KI spielt auf Atari Konsole
  https://www.youtube.com/watch?v=dJ4rWhpAGFI

Vielen Dank

Algorithmus Schmiede
Data Science | Numerik | Physik

Dr. Markus Dutschke
Inhaber
+49 178 148 32 64
Algorithmus Schmiede
impact@algorithmus-schmiede.de
www.algorithmus-schmiede.de

Backup Slides

Es folgen einige ergänzende Slides zum Vortrag.
Sollten diese Themen Ihr Interesse Wecken, wenden Sie sich gerne an uns.

Was soll schon schief gehen?
Bias in Trainingsdaten

iPhone X - Face ID kann chinesische Frauen nicht voneinander unterscheiden

Aus China kommt ein Bericht, laut dem die Gesichtserkennung Face ID zwei chinesische Arbeitskolleginnen nicht voneinander unterscheiden kann.

gamestar.de

Microsoft

Twitter-Nutzer machen Chatbot zur Rassistin

Tay, ein Chatbot von Microsoft mit künstlicher Intelligenz, sollte im Netz lernen, wie junge Menschen reden. Nach wenigen Stunden musste der Versuch abgebrochen werden.

Von **Patrick Beuth**

zeit.de

24. März 2016, 15:13 Uhr / 51 Kommentare /

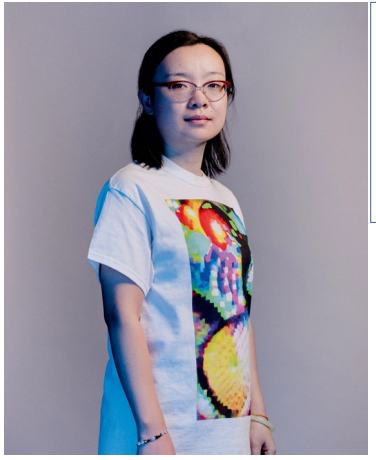
Video-Vorschläge von Facebook

Algorithmus verwechselt Schwarze mit Affen

04.09.2021 09:34 Uhr

zdf.de

Was soll schon schief gehen?
Adverserial Examples



ALEX LEE SECURITY 11.05.2020 06:00 AM wired.co.uk

This ugly t-shirt makes you invisible to facial recognition tech

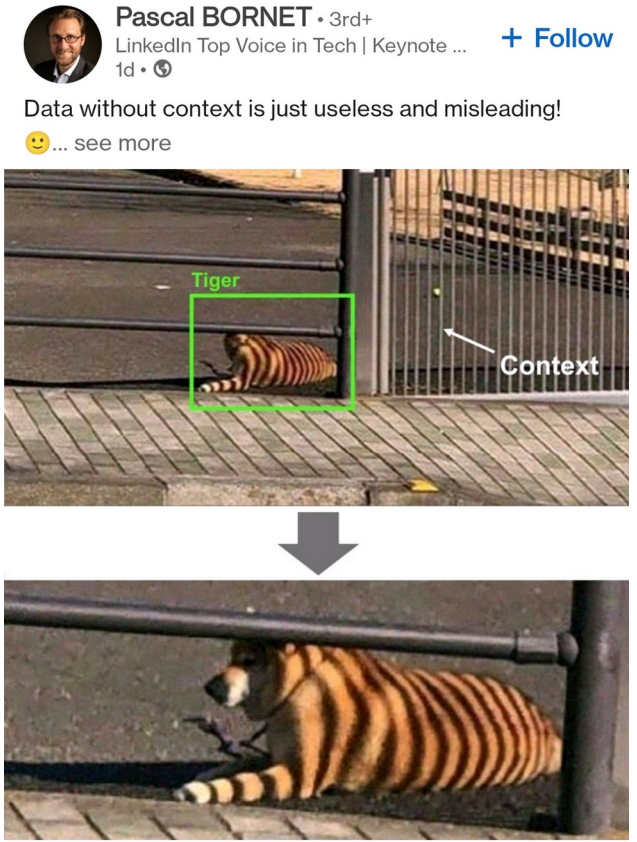
Researchers at Northeastern University have developed an adversarial example that works even when printed onto a moving fabric

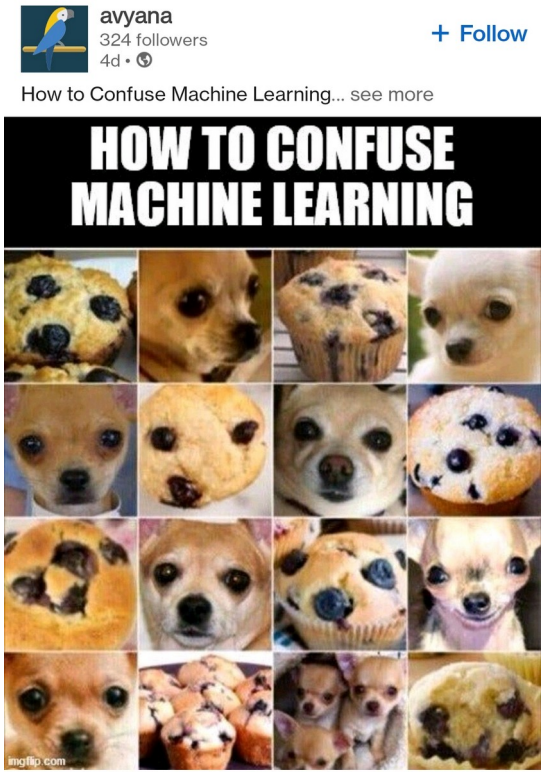
Fooling a Real Car with Adversarial Traffic Signs

Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, Yuval Weisglass

Harman International, Automotive Security Business Unit arxiv.org

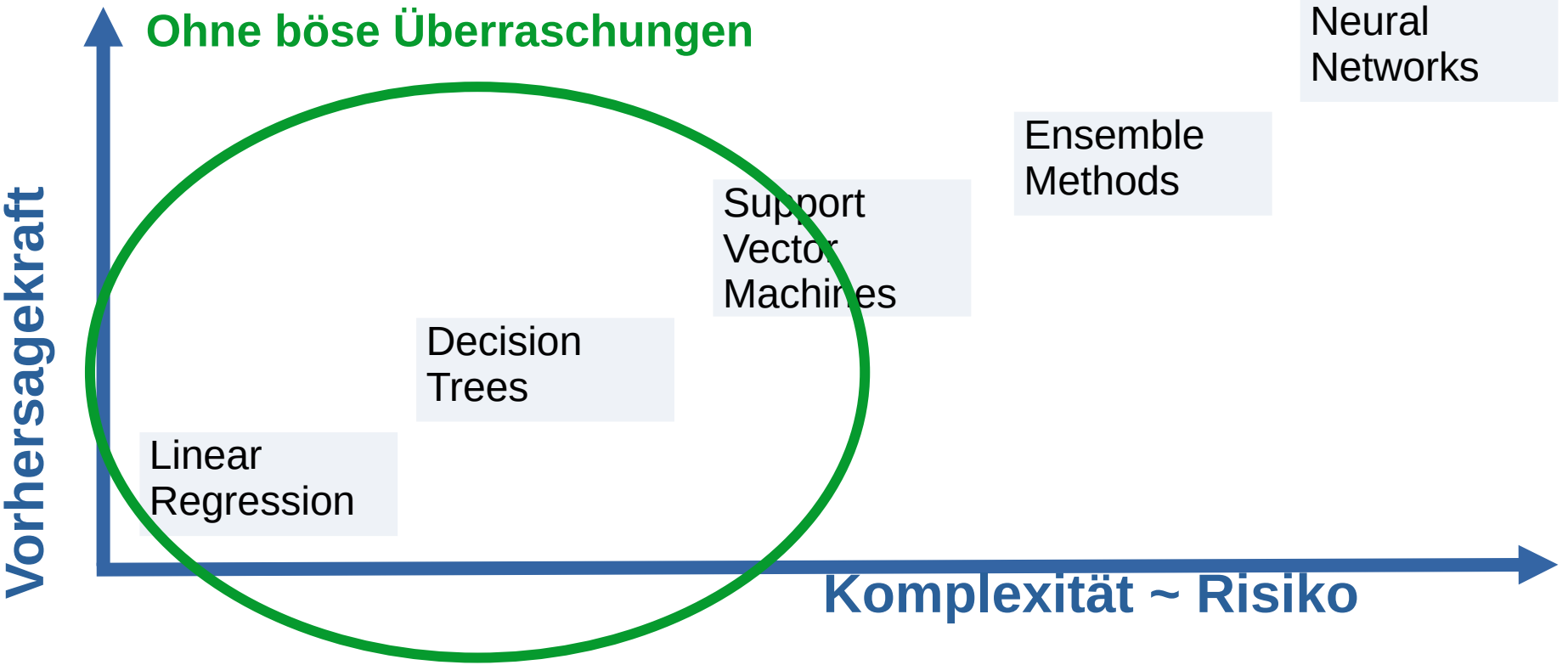
Was soll schon schief gehen?
Plausibilität, Kontext, Spezialfälle





Quelle: LinkedIn

Machine Learning Modelle



Algorithmus Schmiede
Data Science | Numerik | Physik ©

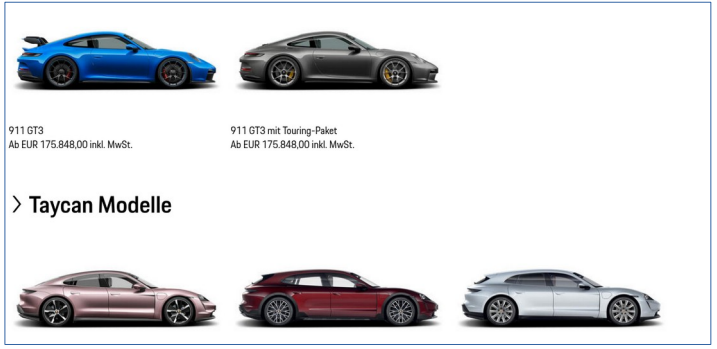
Idealfall für die Anwendungen von Neuronalen Netzen

- Daten werden automatisch erzeugt
  (z.B. Sensor in Produktion)
- Trainingsdaten sind repräsentativ
  Temperatur: Sommer/Winter
  Beleuchtung: Tag/Nach
  Maschine: selbe / nicht gleiche
- Eine Lösung ist mit keiner anderen bekannten Methodik realistisch
- Ein Ausfall führt zu keinem großen Schaden oder lässt sich durch Notfallalgorithmen verhindern

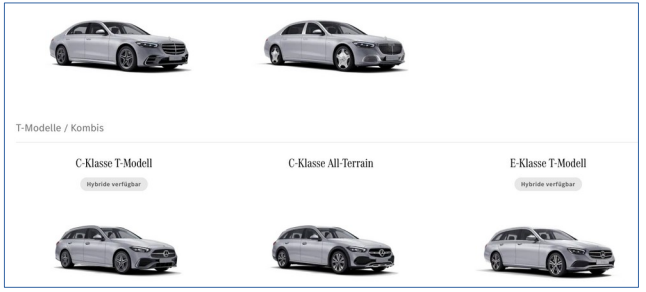
Unter diesen Umständen lässt sich der Arbeitsaufwand gut abschätzen und die Risiken halten sich in Grenzen. Je weniger der Voraussetzungen erfüllt sind, desto herausfordernder wird das Entwicklungsprojekt.

Aufgabe: Finde Automarke

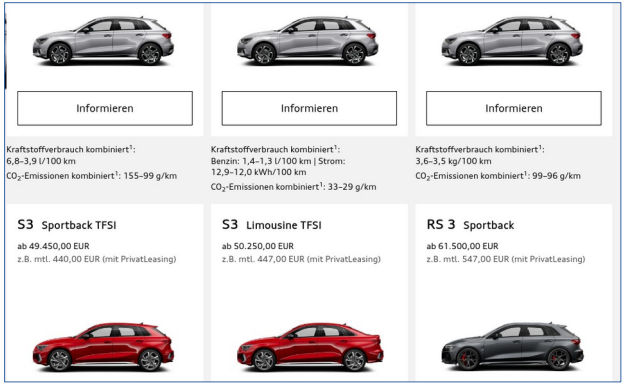
https://www.porsche.com/germany/models/



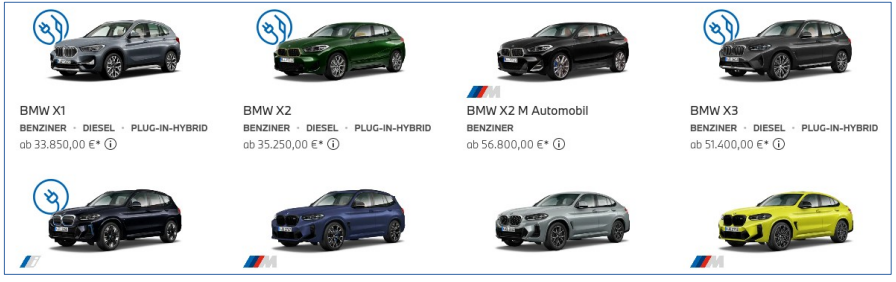
https://www.mercedes-benz.de/passengercars/models.html



https://www.audi.de/de/brand/de/neuwagen.html



https://www.bmw.de/de/neufahrzeuge.html



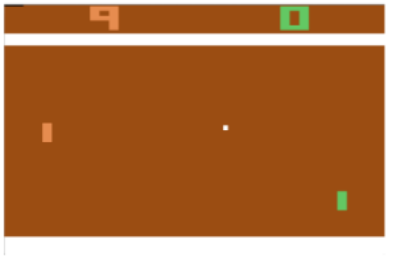
Aufgabe: Finde Automarke

Erwartete Entwicklungsstadien eines Neuronalen Netz
bis zur gewollten Funktionsweise

- Ausrichtung des Autos, evtl. nur untere Kante des Schattens
- Verhältnis von weißen Pixeln im Bild
- Farbe
- Metasymbole
- Winkel der Beleuchtung / Schattenverläufe entlang der Karosserie
- ...
- Nebensächlichkeiten wie: Anhängerkupplung, Blinker im Seitenspiegel, Felgen, ...
- ...
- Modellreihen zu jeder Marke + relevante Merkmale wie Herstelleremblem

Reinforcement Learning: Anwendungen

- Komplexe Aufgaben in der Regelungstechnik
- Alle Probleme ähnlich zu Gesellschafts- oder Compute
- Atari 2600 Spiele



Beispiele:

- Objekte greifen und stapeln
- Spurhalte-Assistent
- Balance Aufgaben
- Notfallsteuerung, um Produkt (z.B. Lebensmittel) nach Systemfehler noch zu Retten
- Routenplanung in dynamisch veränderlicher Umgebung
- Dynamische Preisgestaltung

Reinforcement Learning

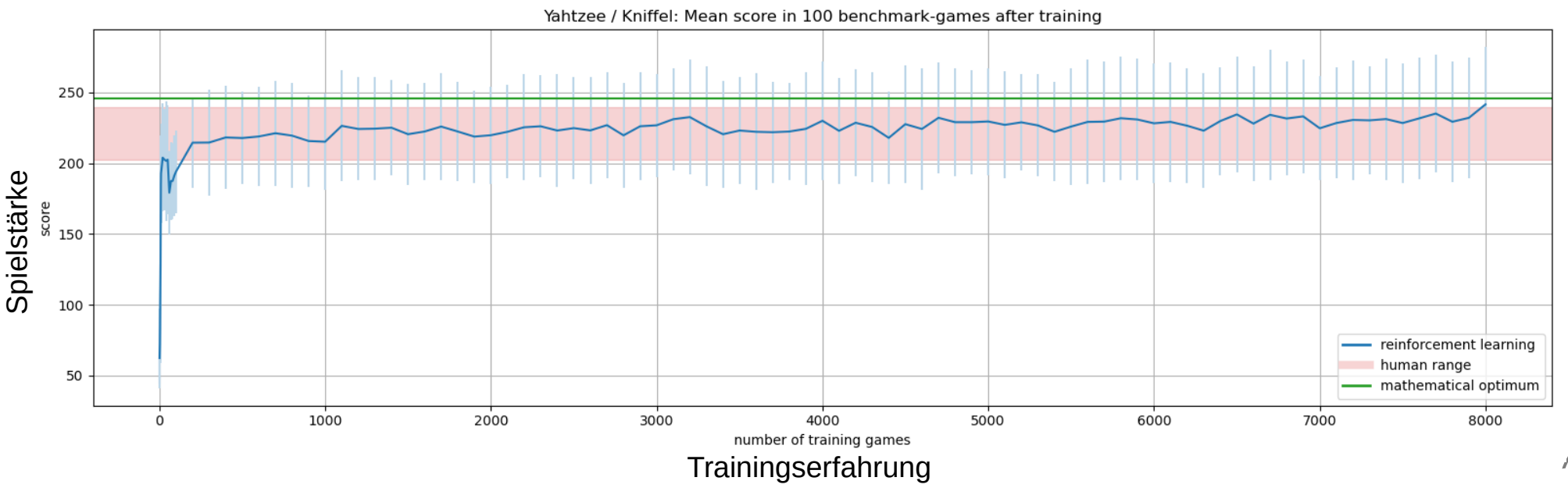
Funktionsweise:

- Regressor: (aktuelle Situation, Steuersignal) -> finaler Punktestand
- Training des Regressors etwas trickreich

Sobald der Regressor fertig ist, ist das Problem gelöst, denn:

- Berechne den finalen Punktestand für jedes mögliche Steuersignal in der aktuellen Situation.
- Wähle das Steuersignal, welches in der aktuellen Situation den höchsten finalen Punktestand erzielt.

Reinforcement Learning

Anwendung: Selbstlernender Yahtzee / Kniffel Bot

- Erreicht nach 8000 Spielen übermenschliche Spielstärke