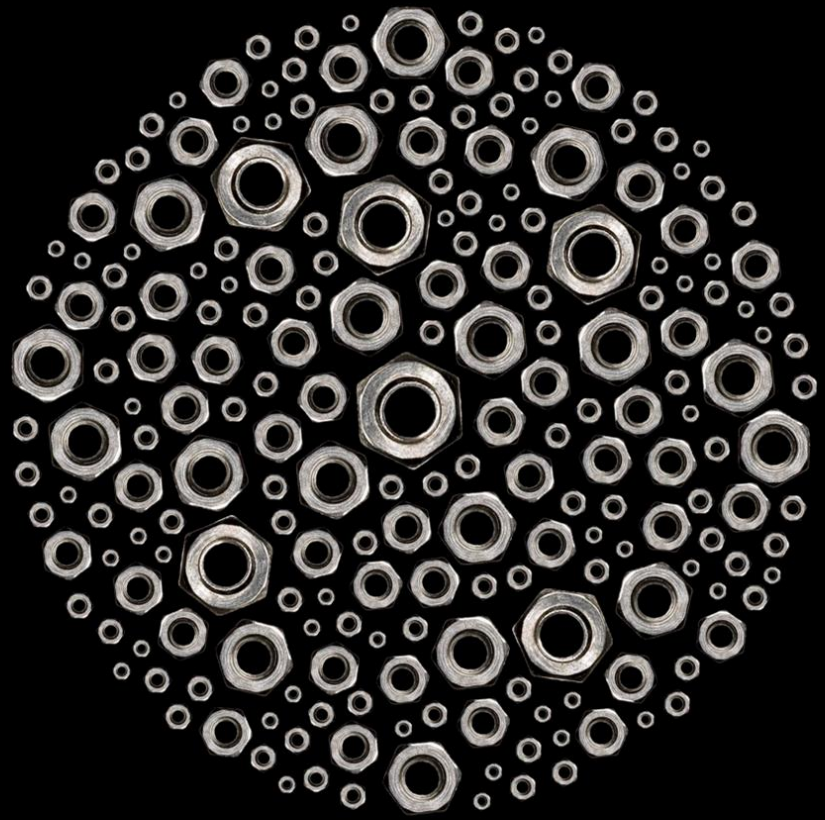
Deloitte.



Schadenvorhersage in der Krankenversicherung mithilfe von Random Forests

Zoran Nikolić

MAKING AN IMPACT THAT MATTERS since 1845

Vortragender *

Zoran Nikolić

Deloitte.

Universität zu Köln

* Die Umsetzung in Zusammenarbeit mit Vinothan Sriharan

Agenda

01 **Einführung**

02 **Neuronale Netze**

03 **Entscheidungsbäume**

- Einleitung Entscheidungsbäume
- Regressionsbäume
- Klassifikationsbäume
- Random Forests

04 **Leistungsschätzung in der KV**

- Kopfschäden
- Daten
- Modellierung
- Ergebnisse

05 **Fazit und Ausblick**

Einführung

Einführung

Künstliche Intelligenz und Machine Learning

Künstliche Intelligenz (KI) / Artificial Intelligence (AI):

- Der Versuch menschenähnliche Entscheidungsstrukturen in einem nicht eindeutigen Umfeld nachzubilden
- **Starke künstliche Intelligenz*** = KI im philosophischen Sinne
  -> Selbstbewusste Maschinen
- **Schwache künstliche Intelligenz*** = KI im technischen Sinne
  -> Maschinen, die intelligent erscheinen

Erkenntnisse über die Art, wie die Entscheidungs-findung im Gehirn erfolgt

Mathematische Methoden zur Extraktion von Erkenntnissen aus den Daten

Machine Learning (ML):

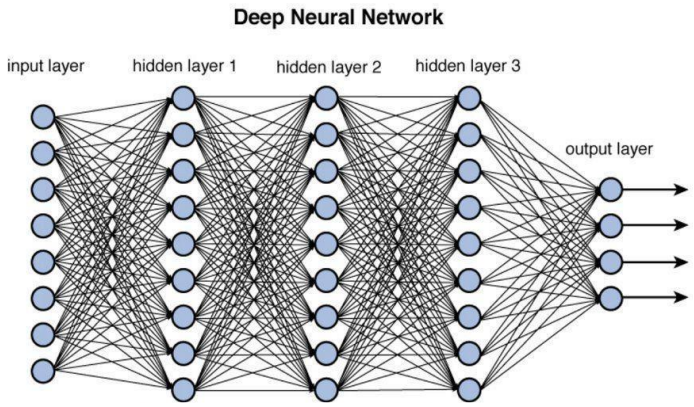
- **Allgemein:** Maschinelle Generierung von Wissen aus Erfahrung
- **Konkret:** Methoden der Programmierung (von Maschinen), so „dass ein bestimmtes Leistungskriterium anhand von Beispieldaten oder Erfahrungen aus der Vergangenheit optimiert wird.“*
- Die Maschinen ahmen dabei einige der kognitiven Fähigkeiten von Menschen nach
- Daher wird ML häufig als Teilgebiet der künstlichen Intelligenz (**artificial intelligence**) klassifiziert.

* aus: Russel, S., Norvig, P.: **Artificial Intelligence – A Modern Approach**, Prentice Hall, 2009

Machine Learning im Jahr 2021

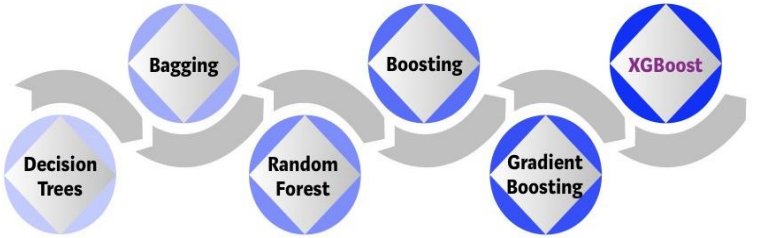
Entscheidungsbäume und neuronale Netze

Deep Learning



Zwei Ansätze haben in den letzten Jahren besonders oft Machine-Learning-Wettbewerbe gewonnen

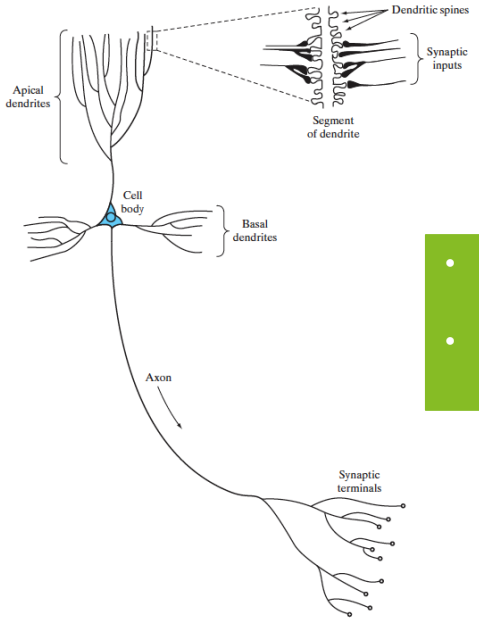
Entscheidungsbäume



Neuronale Netze

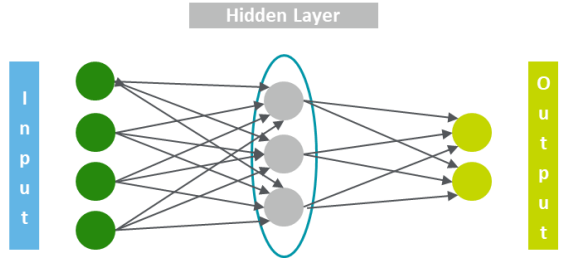
Deep Learning

Ursprung und Aufbau



- **Die Idee für das mathematische Modell kam tatsächlich aus der Gehirnforschung**
- **Aktuelle Modelle mitunter weit vom Vorbild entfernt**

Neuronale Netze



- **Idee:** „es so machen, wie das menschliche Gehirn“
- **Methode:** Verknüpfung sog. künstlicher Neuronen (Units)
- **Bestandteile:** Input Units, Hidden Layer, Output Units
- **Verknüpfung:** Austausch von Daten (= Zahlen) zwischen den Units
- Dabei hat jede Verknüpfung ein **Gewicht** (nicht konstant)
- In jeder Unit wird ein **Input** verarbeitet und ein **Output** erzeugt, der wieder ein Input für eine nachgelagerte Unit sein kann
- **Deep Learning:** Es gibt mehr als eine Hidden Layer

Deep Learning

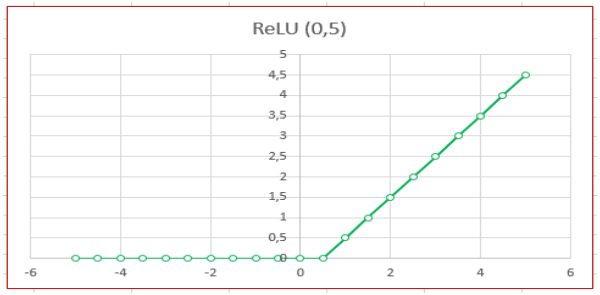
Aktivierungsfunktionen

Der **Output** einer Unit wird aus der Summe der gewichteten Inputwerte, ggf. einem Bias b und einer sog. **Aktivierungsfunktion** f, welche die Reaktion auf die Eingangsimpulse darstellt, berechnet.

Beispiele für Aktivierungsfunktionen:

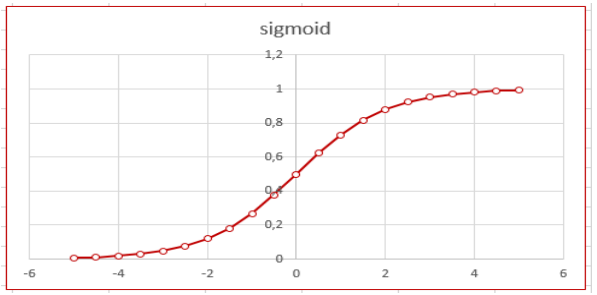
Rectified Linear Unit (ReLU)

$$f(x, \theta) = max(0, x - \theta)$$



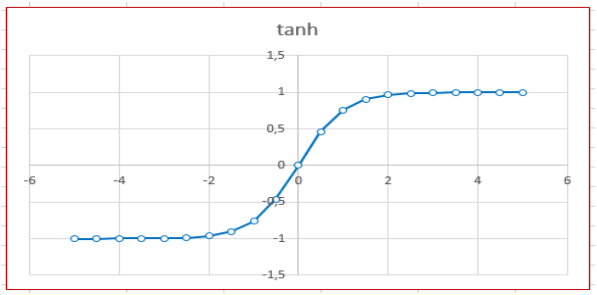
Sigmoidfunktion

$$f(x) = \frac{1}{1 + e^{-x}}$$



Tangens Hyperbolicus

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



Deep Learning

Überwachtes Lernen und Gradientenabstiegsverfahren

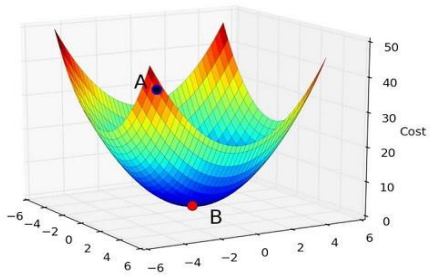
Überwachtes Lernen

- Festlegung der **Startgewichte**
- Festlegung einer **Lernrate**
- ggf. Varianten mit und ohne **Momentum**
  - z.B.: 0,75 der letzten Änderung
- Festlegung einer **Kostenfunktion**
  - z.B.: Quadratsumme (Maximum Likelihood), Euklidis Abstand (rechenintensiver), Mittlerer quadratischer Fehler
- Dann **iterativ** für jeden Trainingsdatensatz
  - Berechnung des Outputs des Netzes für den Input ei Teilmenge der Daten (**Prediction**)
  - Vergleich mit dem vorgegebenen Output („**Label**“) auf dieser Teilmenge
  - Anpassung der Gewichte durch **Backpropagation** = Minimierung der Kostenfunktion

Training/ Lernen

Gradientenabstiegsverfahren

- **Problem:** Das Netz liefert schon für den ersten Trainingsdatensatz nicht den richtigen/ erwarteten Output
- **Vorgehen:**
  - Der Fehler ist eine Funktion aller Gewichte:
  - Verbesserung des Netzes, d.h. Verringerung des Fehlers, indem die Gewichte geändert werden
  - **Konkret:** Verbesserung der Gewichte, durch herabsteigen in kleinen Schritten entlang der Fehlerfunktion
  - Der Gradient liefert die Richtung des steilsten An- bzw. Abstieges
  - $\nabla E = \left(\frac{\partial E}{\partial w_{l,k}}\right)_{l,k}$ für Gewichte $w_{l,k}$

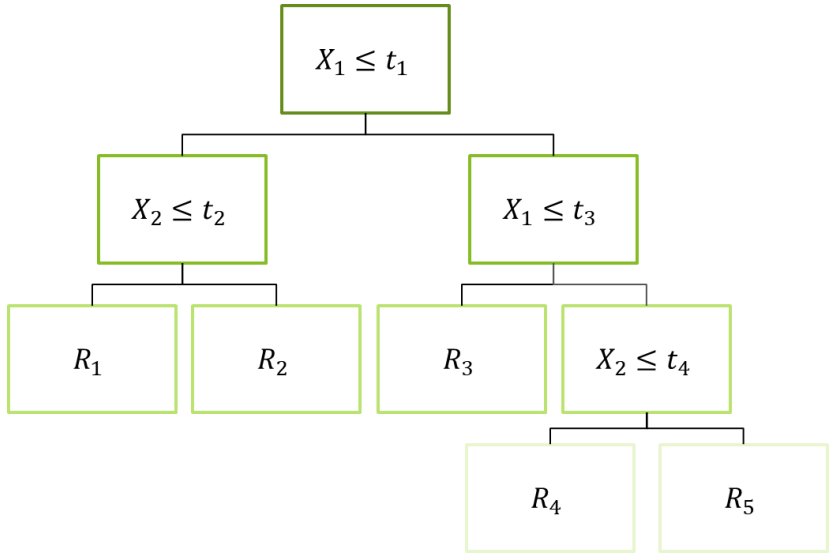


Entscheidungsbäume

Einleitung Entscheidungsbäume

Divide and Conquer

- **Entscheidungsbäume** sind eine beliebte Methode(*):
  - Basieren auf einfachen If-Then-Abfragen
  - Gute Lernperformance
  - Geeignet zur Visualisierung von Entscheidungen
- Entscheidungsbäume werden im weiten Bereich des maschinellen Lernens eingesetzt, sowohl zur **Klassifikation** als auch zur **Regression**
- Baumbasierte Methoden teilen den **Featureraum** in Rechtecke auf und fitten anschließend ein simples Modell für jedes der Rechtecke
- Zur Vereinfachung: Beschränkung auf eine **rekursive binäre** Aufteilung:
  - Großer Vorteil: Interpretierbarkeit
  - Aufteilung des Featureraums mit einem Baum vollständig beschrieben

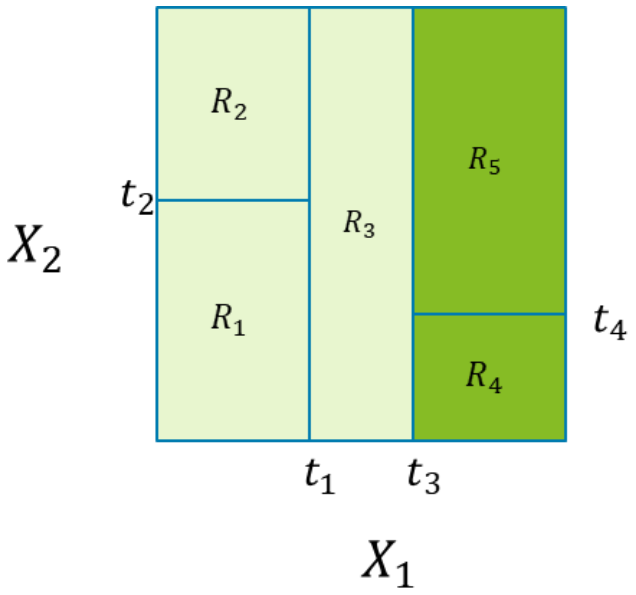


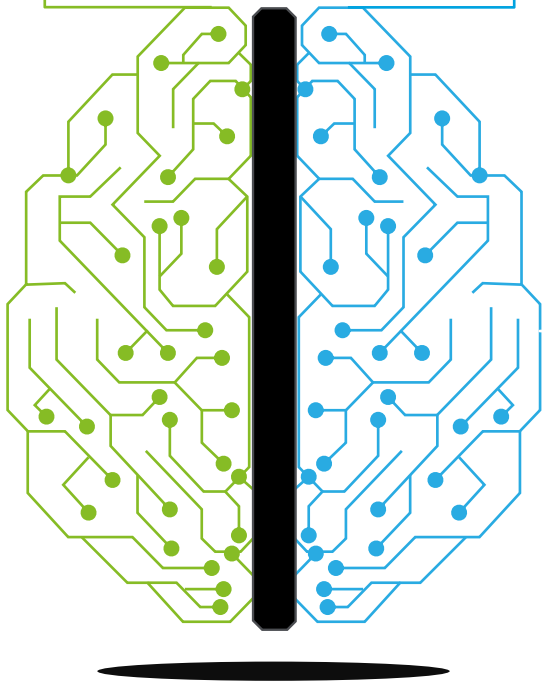
(*) Eigentlich: Die am meisten verwendete ML-Methode (P. Domingos, „The Master Algorithm“)

Einleitung Entscheidungsbäume

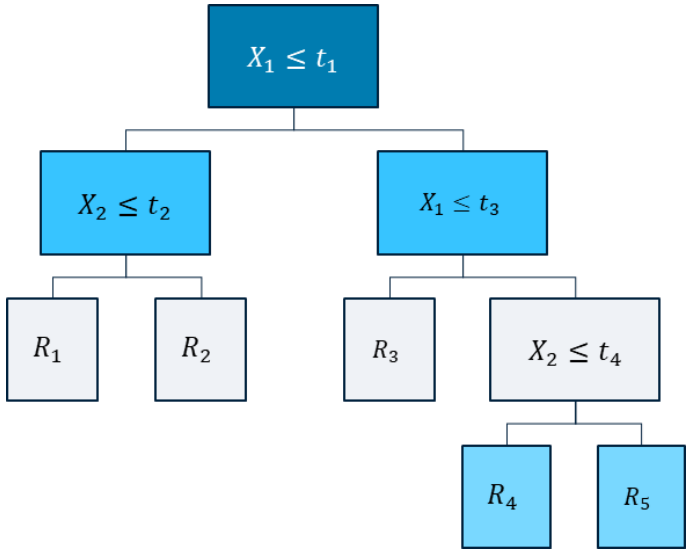
Unterteilung

Featureraum





Entscheidungsbaum



Quelle: An Introduction to Statistical Learning; G.James, D.Witten, T.Hastie, R.Tibshirani; Springer Verlag (2017)

Regressionsbäume

Der Ansatz

- Daten beinhalten p **Eingabevariablen** X (*input variables*) und eine **Zielvariable** Y (*response variable*) für jede der *N* Observationen (x_i, y_i) für $i = 1, 2, \dots, N$ mit $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$
- Mit der Annahme, dass eine Aufteilung des **Featureraumes** in *M* Regionen $R_1, R_2, \dots, R_M$ vorliegt, wird die **Zielvariable** mit einer Konstante c_m in jeder Region modelliert mit:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

- Das bestmögliche $\hat{c}_m$ ist der Durchschnitt von y_i in der Region R_m:

$$\hat{c}_m = ave(y_i | x_i \in R_m),$$

denn dabei wird die Summe der Quadratresiduen $\sum((y_i - f(x_i))^2$ minimiert

Training der Bäume

Der Greedy- Algorithmus

- Finden der bestmöglichen binären Aufteilung ist bezüglich dem Minimum der Summe der Quadratresiduen sehr rechenintensiv
  - ➢ *Greedy*-Algorithmus stellt eine effiziente Lösung dar
- Starte mit dem vollständigen Datensatz, definiere zunächst eine Splittingvariable j, einen Splitpunkt s und ein Paar von Halbebenen

$$R_1(j,s) = \{X|X_j \leq s\} \text{ und } R_2(j,s) = \{X|X_j > s\}$$

- Suche die Splittingvariable j und den Splitpunkt s, welcher das folgende Minimierungsproblem löst:

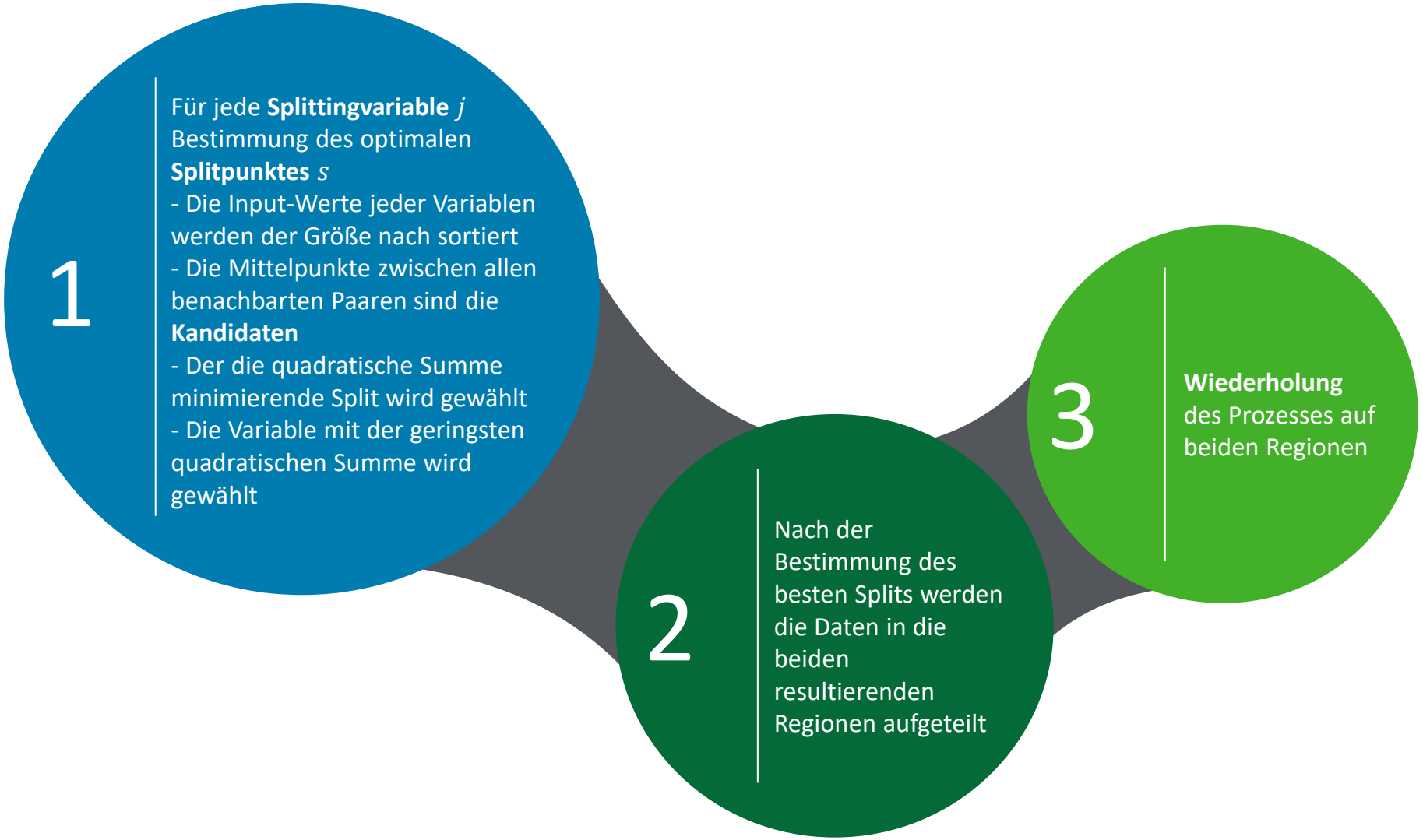
$$\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right]$$

- Für jede Wahl von j und s wird die innere Minimierung gelöst durch:

$$\hat{c}_1 = ave(y_i|x_i \in R_1(j,s)) \text{ und } \hat{c}_2 = ave(y_i|x_i \in R_2(j,s))$$

Maschinenentscheidungen

Vorgehensweise Greedy-Algorithmus

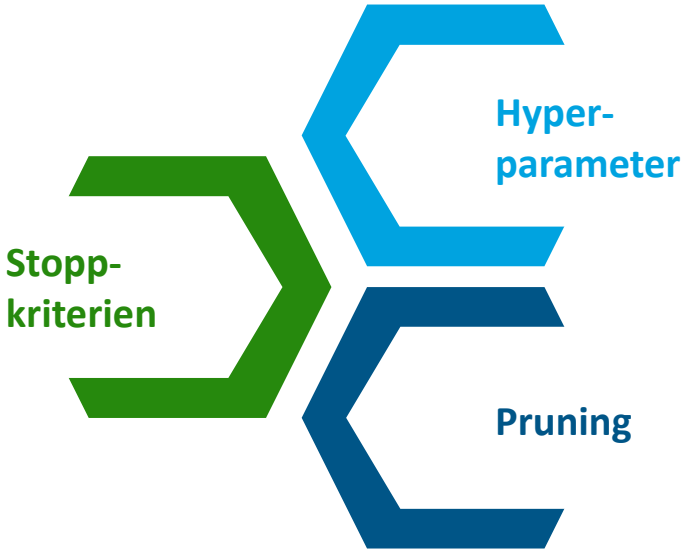


Bestimmung der Baumgröße

Regularisierung

Frage: Wie groß soll der Baum wachsen?

In der Praxis häufige Verwendung von Stoppkriterien



Baumgröße ist ein Hyperparameter (tuning parameter), der die Modellkomplexität steuert

Darüber hinaus wird desöfteren „Pruning“ verwendet, um ein Overfitting (Überanpassung) zu vermeiden

Bestimmung der Baumgröße

Stoppkriterien

Random Forests

Bagging

Sind $X_1, X_2, \dots, X_B$ unabhängig, identisch verteile Zufallsvariablen mit Varianz $Var(X_i) = \sigma^2 \; \forall \; i \in \{1, \dots, B\}$, so ist die Varianz von $\bar{X} \coloneqq \frac{1}{B}\sum_{i=1}^{B}(X_i)$ gegeben durch

$$Var(\bar{X}) = \frac{\sigma^2}{B}$$

Bagging

= **B**otstraps **Agg**regat**ing**, macht von dieser Eigenschaft Gebrauch

Bootstraps

Aggregation

Bootstraps

Was sind Bootstraps?

- Kernidee: Unabhängiges Ziehen mit Zurücklegen

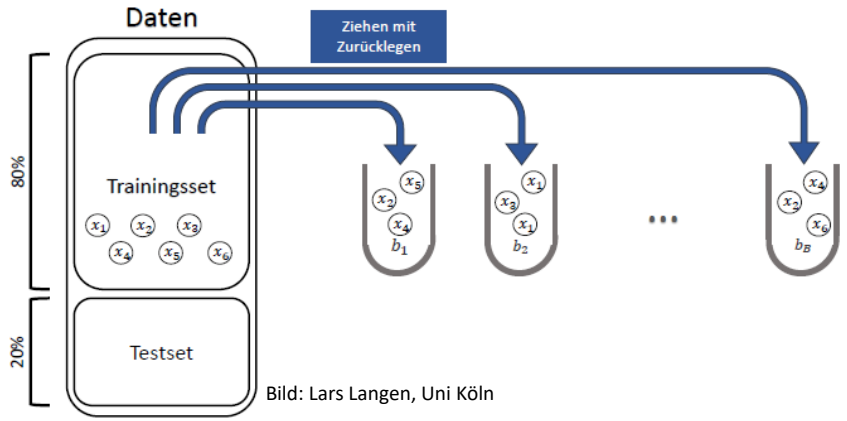


Bild: Lars Langen, Uni Köln

Generieren von B Mengen $b_1, b_2, \ldots, b_B$

B-maliges Ziehen mit Zurücklegen aus dem Trainingsset T

Jede Menge b_i soll $|T|$ Elemente enthalten

Erwartung: $1 - {}^1/_e = 63{,}2\ \%$ einzigartige Elemente in b_i, $36{,}8\ \%$ Dubletten

Bagging

Der Weg zu einem Schätzer

- Fitte ein Modell für alle $b_1, b_2, \ldots, b_B$
- Dadurch erhalten wir $\hat{f}^{b_1}, \hat{f}^{b_2}, \ldots, \hat{f}^{b_B}$ als Ergebnisse der Fits
- Darüber bilden wir den Durchschnitt:

$$\hat{f}_{bag} = \frac{1}{B}\sum_{i=1}^{B} \hat{f}^{b_i}$$

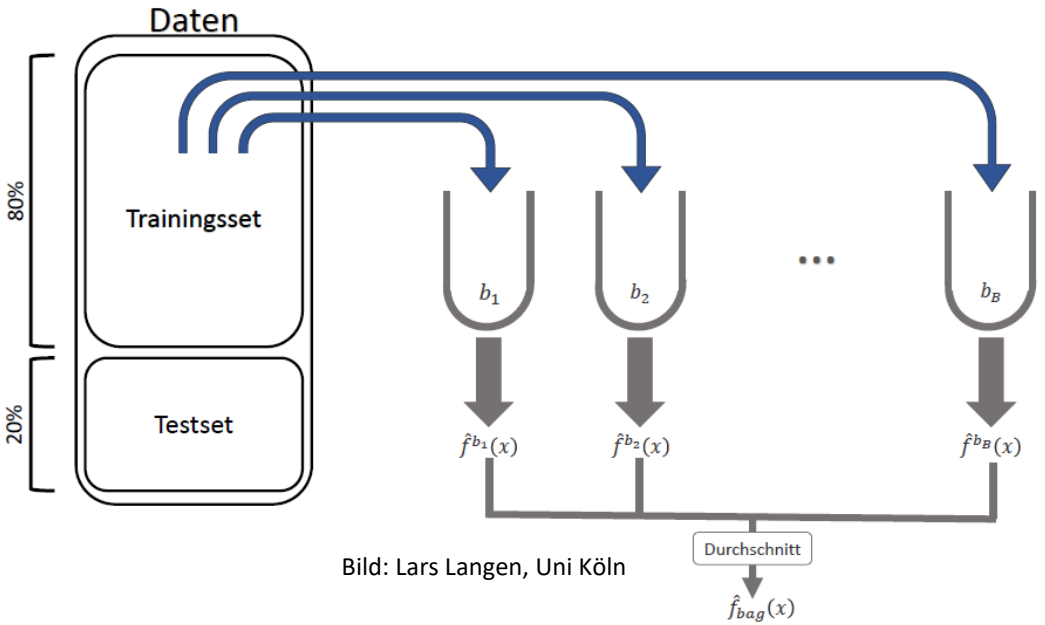


Bild: Lars Langen, Uni Köln

Bagging

Anwendung auf Entscheidungsbäume

- Bagging lässt sich speziell auf **Entscheidungsbäume** anwenden
- Sie verringern die Varianz der Entscheidungsbäume
- **Warum ist das wichtig?**

Random Forests

Bagging mit Splitkandidaten

- Wenn p die Anzahl der erklärenden Variablen, so stehen in jedem Split p Splitkandidaten zur Verfügung

- Idee: Bei jedem Split nur $m < p$ zufällig ausgewählte Splitkandidaten zulassen
  - Häufig wird $m = \sqrt{p}$ gewählt

- Ähnliche Überlegung von eben:
  - $^{p-m}/_{p}$ der Splits haben die eine wichtige Variable gar nicht zur Auswahl

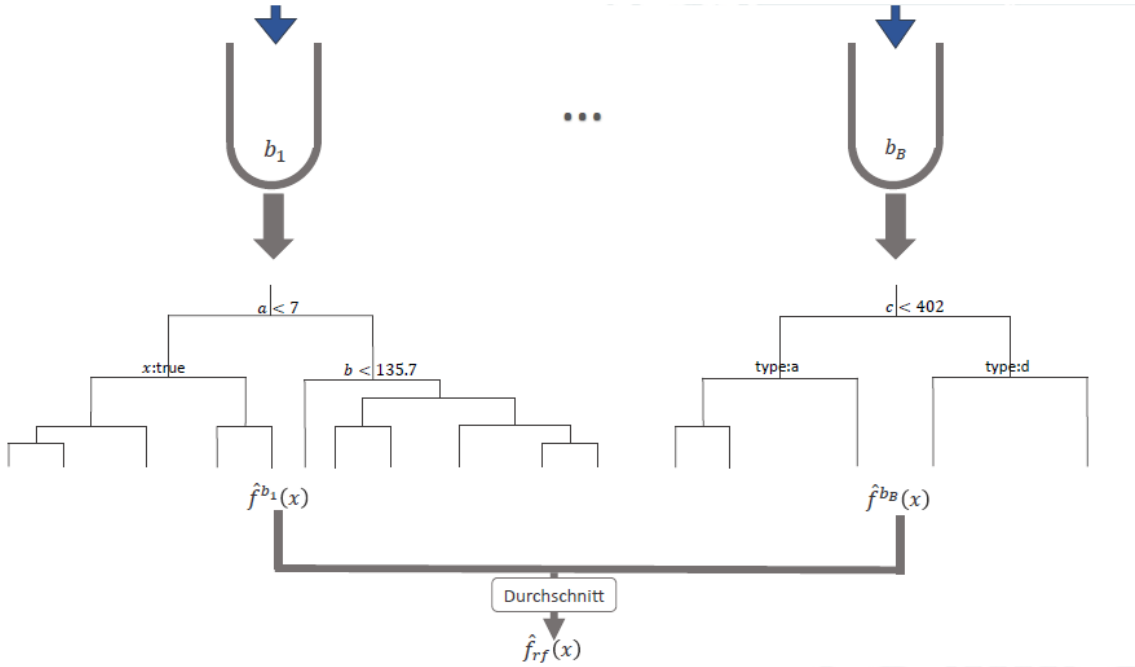


Bild: Lars Langen, Uni Köln

Leistungsschätzungen in der KV

Private Krankenversicherung

Kopfschäden und die konstante Prämie

Beitragskalkulation

Bestimmung eines altersunabhängigen Beitrags, wodurch in jungen Jahren zu viel gezahlt wird, um für die höheren Leistungen im hohen Alter zu „sparen“

Alterungsrückstellung

Basiert auf dem Äquivalenzprinzip und der Tatsache, dass Ältere i. d. R. höhere Leistungen haben

Beitragsbestimmung

Der oben genannten Beitragsbestimmung liegt die Fiktion zeitlich konstanter Kopfschäden zu Grunde

Änderung im Zeitverlauf

Die Kopfschadenreihe

- wird durch Beitragsanpassungen abgebildet
- dabei wird die Veränderung der Kopfschadenreihe berücksichtigt

Kopfschäden

Modellierung nach KVAV, §6

Bezeichnungen

VP_i: die i-te versicherte Person eines Kollektivs von n Versicherten

S_i: die jährliche Versicherungsleistung der VP_i mit $i = 1, \dots, n$

Kollektiv

Ein Kollektiv bezüglich der Kalkulation ist definiert durch **Tarif** ϑ, **Geschlecht** λ und **Alter** x (gemessen in Jahren)

Kopfschaden

Der Kopfschaden eines x-jährigen Versicherungsnehmers im Kalenderjahr μ ist definiert als $K_x(\mu) := E[S_i(\mu)]$ für $i = 1, \dots, n$

Annahme

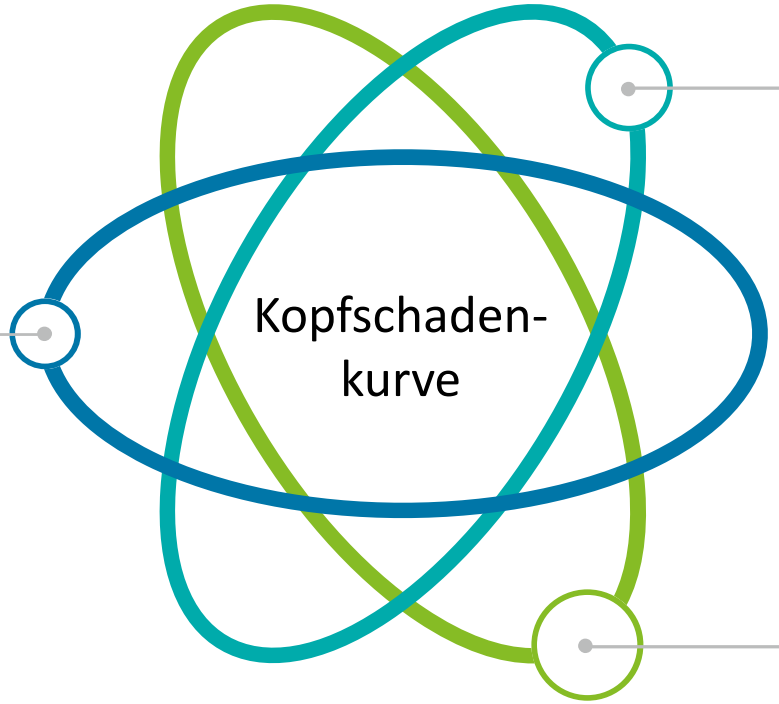
Die Schadenhöhenverteilung aller Personen des Kollektivs identisch

- $S_1, \dots, S_n$ unabhängig und identisch verteilte zufällige Größen

Kopfschadenkurve

Erklärung

Als Vorarbeit für die Beitragsbestimmung ist eine Kopfschadenkurve für die Kalkulation festzulegen

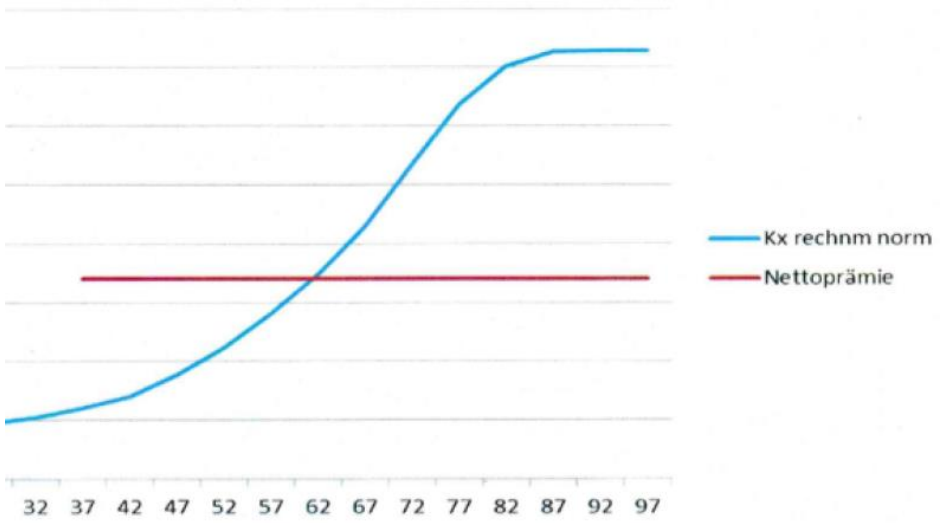


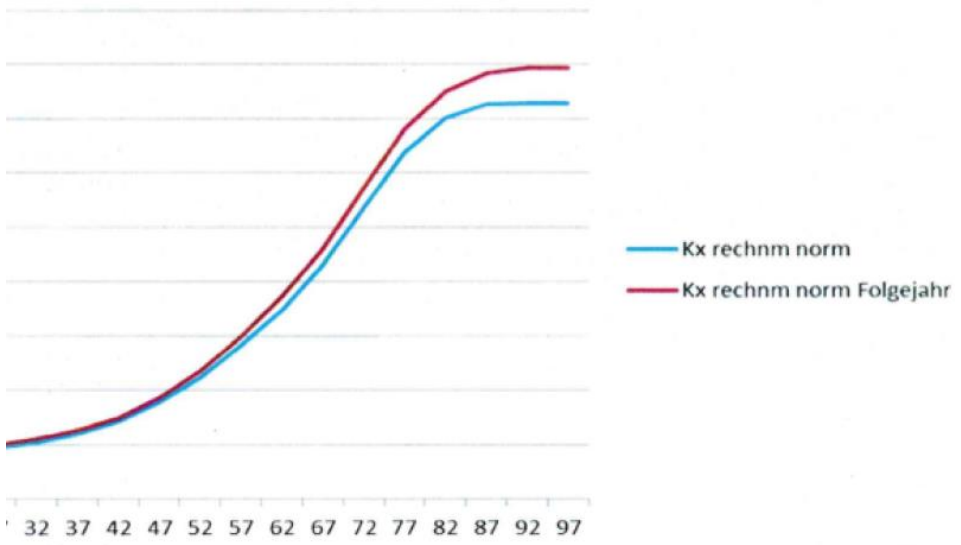
Es ist im Jahr μ eine Prognose der Kopfschäden pro Tarif, Geschlecht und Alter für das Jahr $\mu + 1$ bzw. $\mu + 2$ zu erstellen

Dabei wird die Zeitreihe der tatsächlichen (beobachteten) Kopfschäden der letzten Jahre verwendet

Kopfschadenkurve

Beispiel





Erste Modellierung

Zielsetzung

- Ziel der ersten, vereinfachten Modellierung ist es, für einen vorgegeben Tarif ϑ und Geschlecht λ im Jahr μ die Kopfschadenreihe für das Jahr $\mu + j$ mit $j = 1, 2$ zu prognostizieren
- Verwendung der Daten der Jahre $\mu - 1, \dots, \mu - k$
- Im ersten Schritt beschränken wir uns auf die aktuellen Versicherten im zu untersuchenden Tarif und sagen deren künftigen Schäden $S_{i,\mu+1} = L(VP_i, \mu + 1)$ individuell pro Person vorher
- Der gesuchte Wert wird dann durch $K_x(\vartheta, \lambda, \mu + 1) = \frac{1}{n} \sum_{i=1}^{n} L(VP_i, \mu + 1)$ geschätzt werden

Datensatz

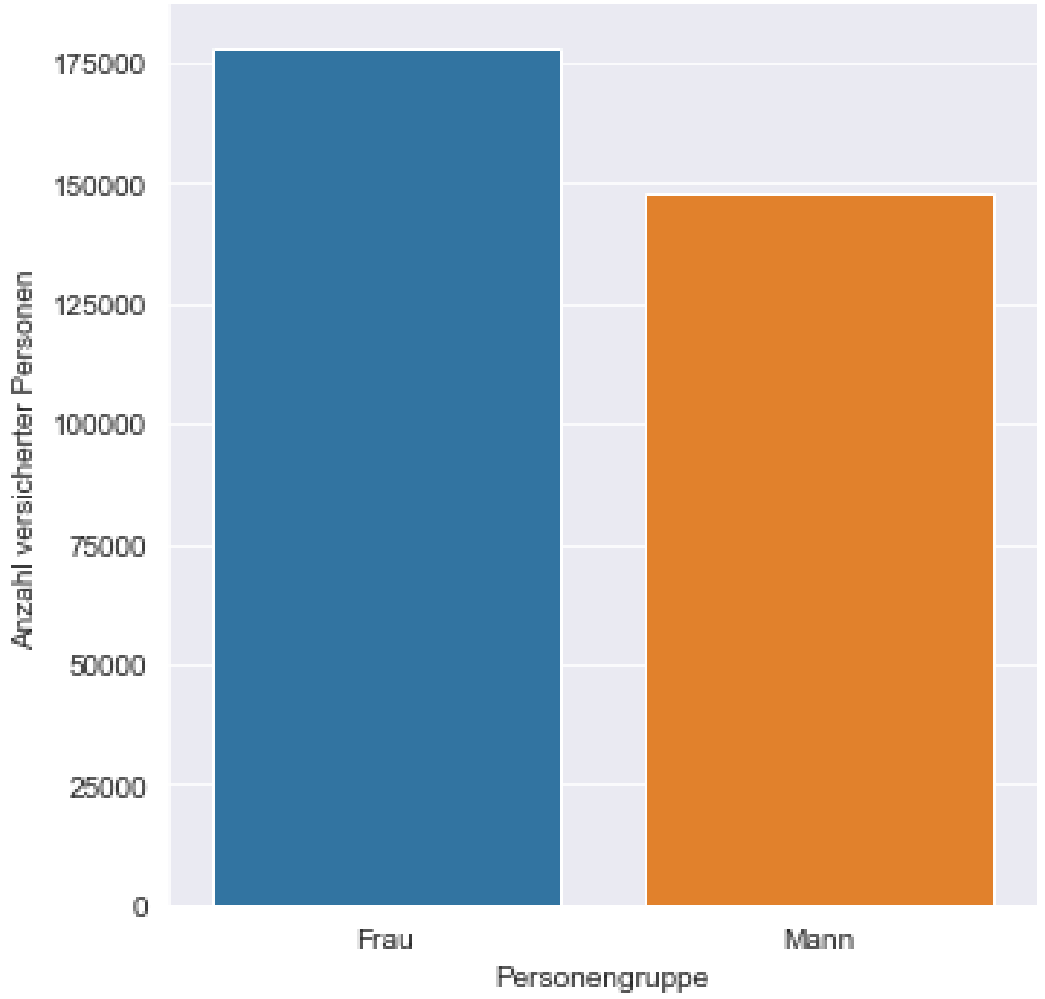
Aufbau

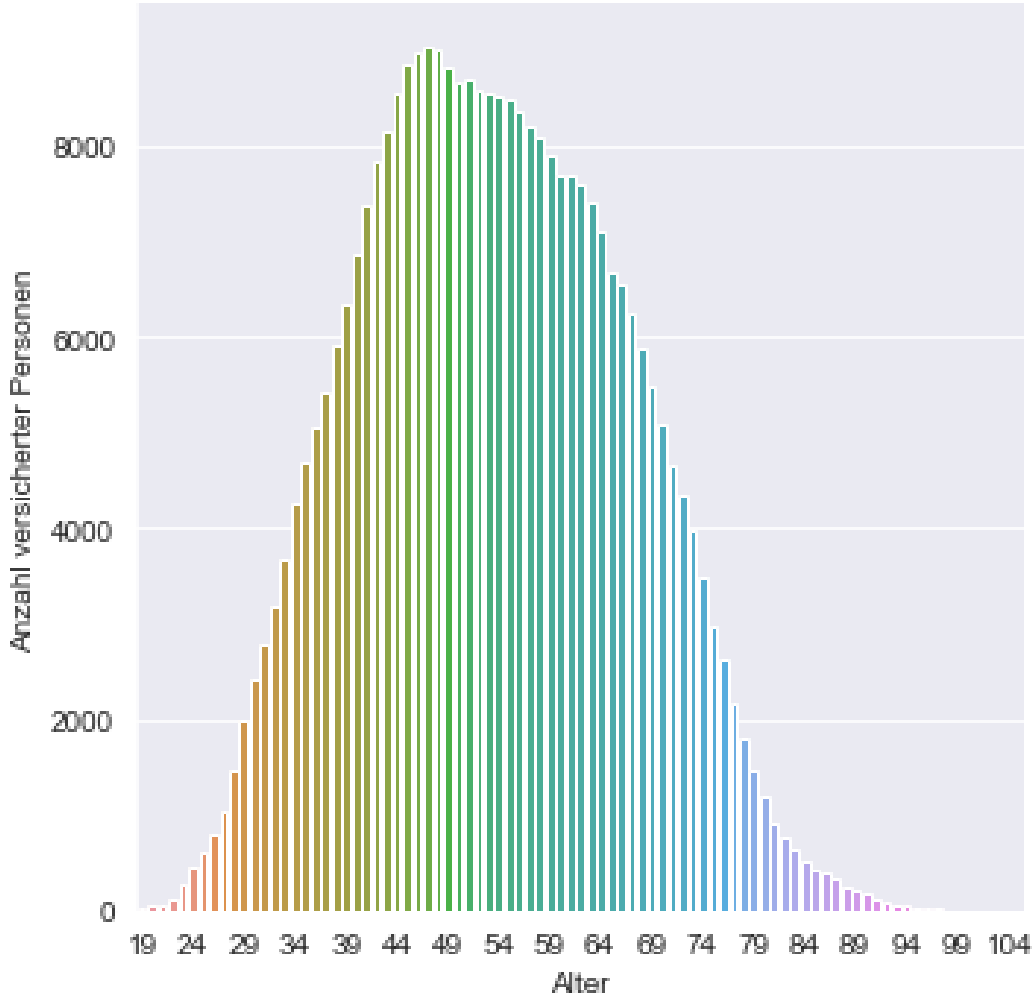
- Es liegt ein Datensatz mit insgesamt 325.568 Observationen vor:
  - Beobachtungsjahre $\mu = \{2006, \dots, 2017\}$
  - Alter $x = \{19, \dots, 105\}$
  - Geschlecht $\lambda = \{Mann, Frau\}$
  - verdichteter Leistungsbetrag L mit $\min = 0€$ und $\max = 339.921€$
- Der Datensatz wurde im Vorhinein auf den zu untersuchenden Tarif $\vartheta = Krankenbeihilfetarif\ 30$ selektiert

Personengruppe	Alter	Beobachtungsjahr	Leistungsbetrag
Frau	19	2006	3968.32
Frau	20	2006	267.6
Frau	20	2006	212.69
Frau	20	2006	250.7
Frau	20	2006	993.87
Frau	20	2006	2194.21
Frau	20	2006	335.31
Frau	21	2006	1949.39
Frau	21	2006	1020.96
Frau	21	2006	45.32
Frau	21	2006	356.08

Datensatz

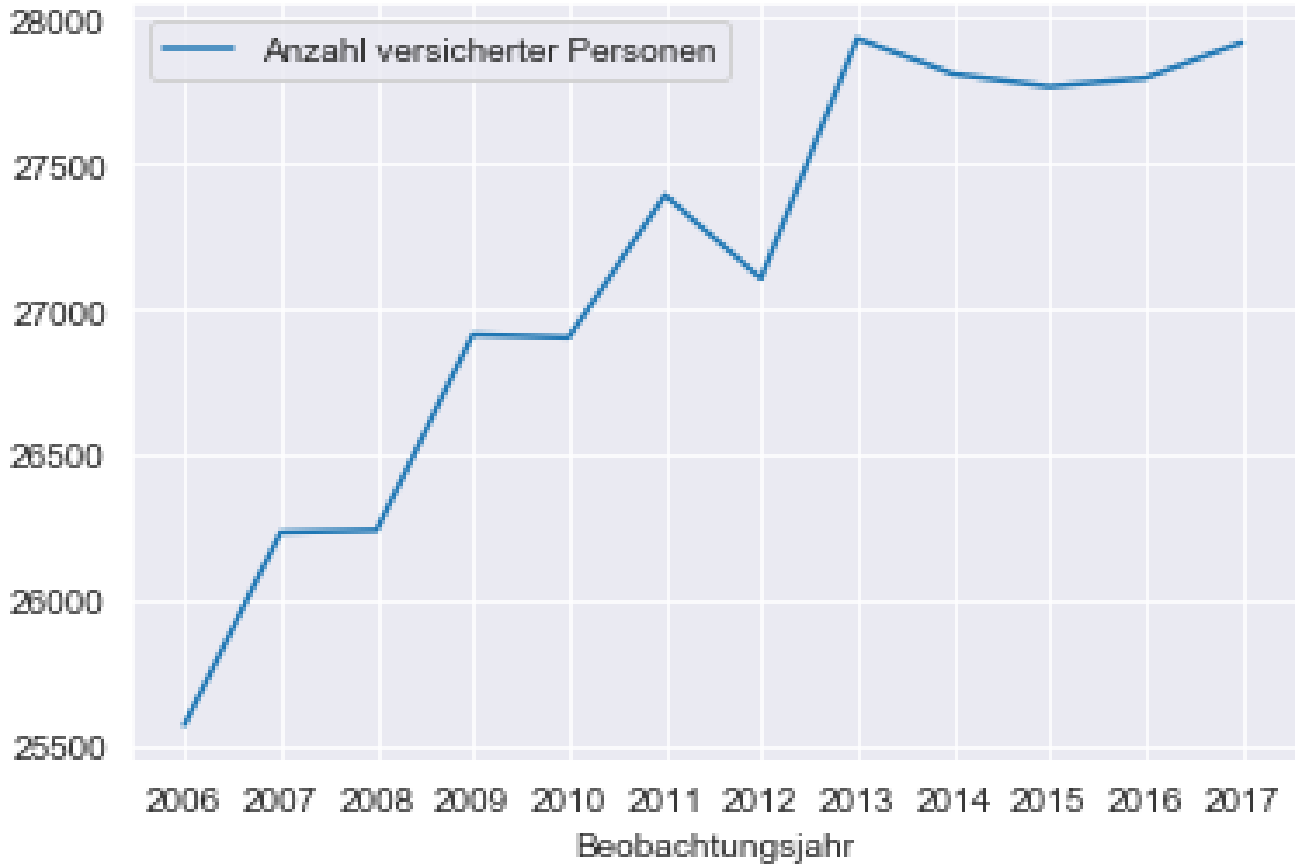
Information





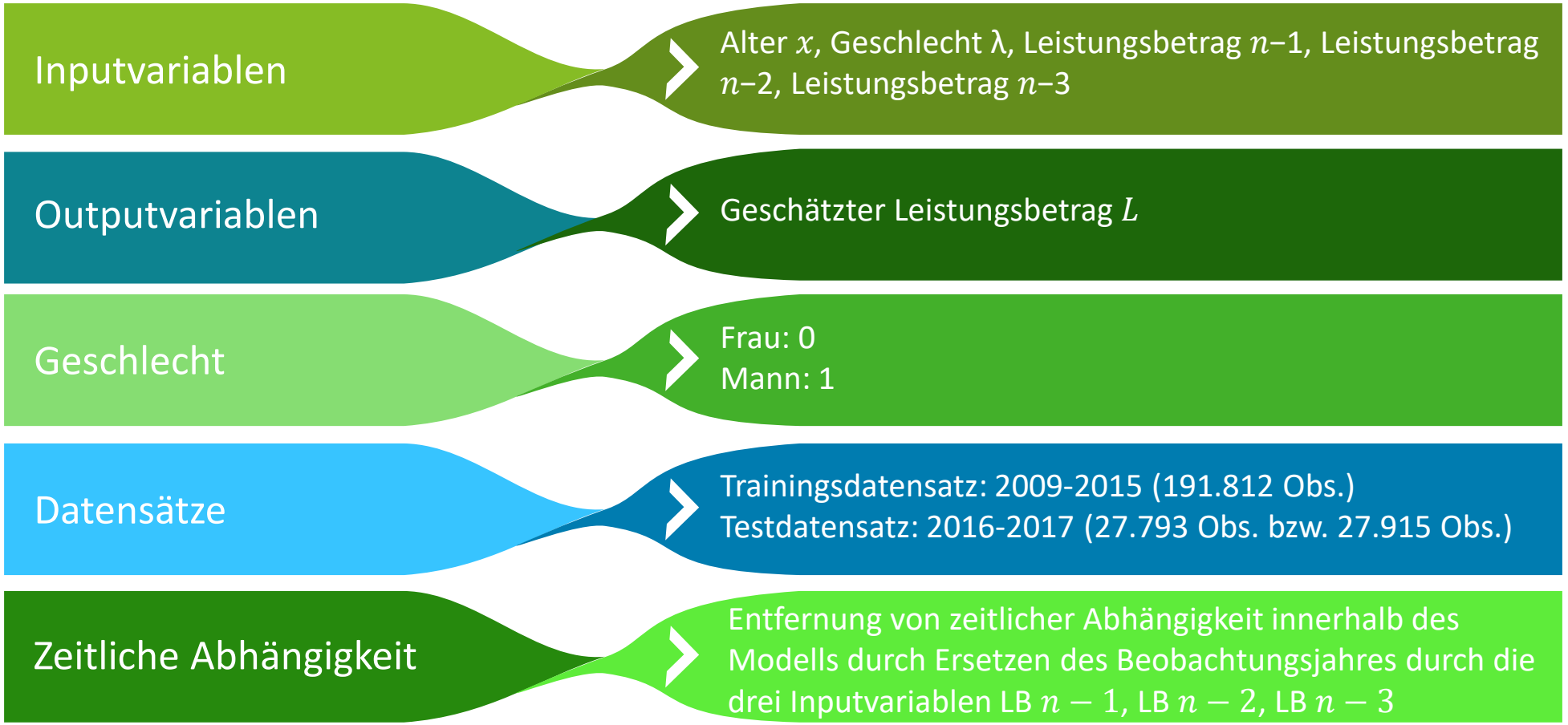
Datensatz

Bestandsentwicklung



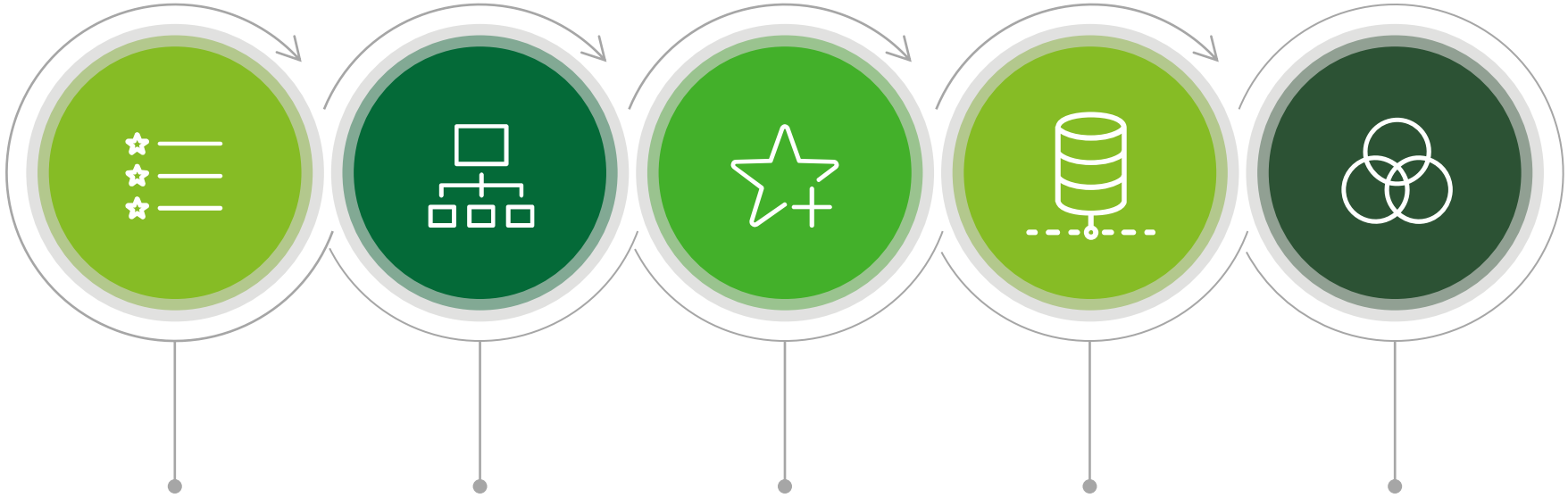
Modellierung

Grundlagen



Modellierung

Hyperparameteroptimierung



n_estimators
300

max_leaf_nodes
90

max_features
2

min_samples_split
2%

min_samples_leaf
90

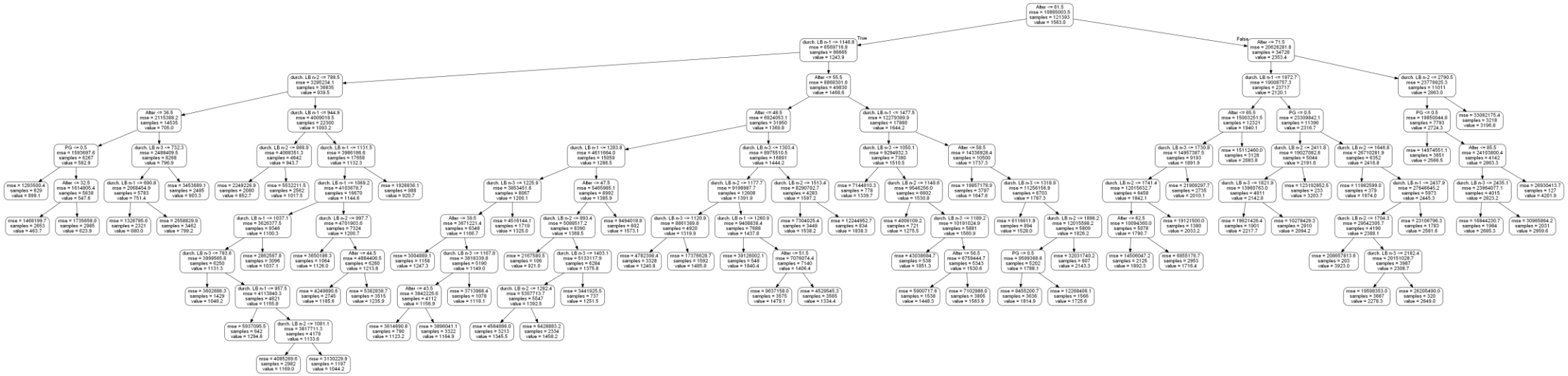
Modellierung

Auszug

Personengruppe	Alter	Beobachtungsjahr	Leistungsbetrag n	Leistungsbetrag n-1	Leistungsbetrag n-2	Leistungsbetrag n-3
Frau	20	2009	461.64	190.06	0	0
Frau	20	2009	834.42	190.06	0	0
Frau	20	2009	0	190.06	0	0
Frau	20	2009	152.06	190.06	0	0
Frau	21	2009	217.01	600.84	134.37	0
Frau	21	2009	0	600.84	134.37	0
Frau	21	2009	175.46	600.84	134.37	0
Frau	21	2009	2909.98	600.84	134.37	0
Frau	22	2009	5982.56	1494.84	253.74	3968.32
Frau	22	2009	205.05	1494.84	253.74	3968.32
Frau	23	2009	360.8	836.91	460.15	709.06

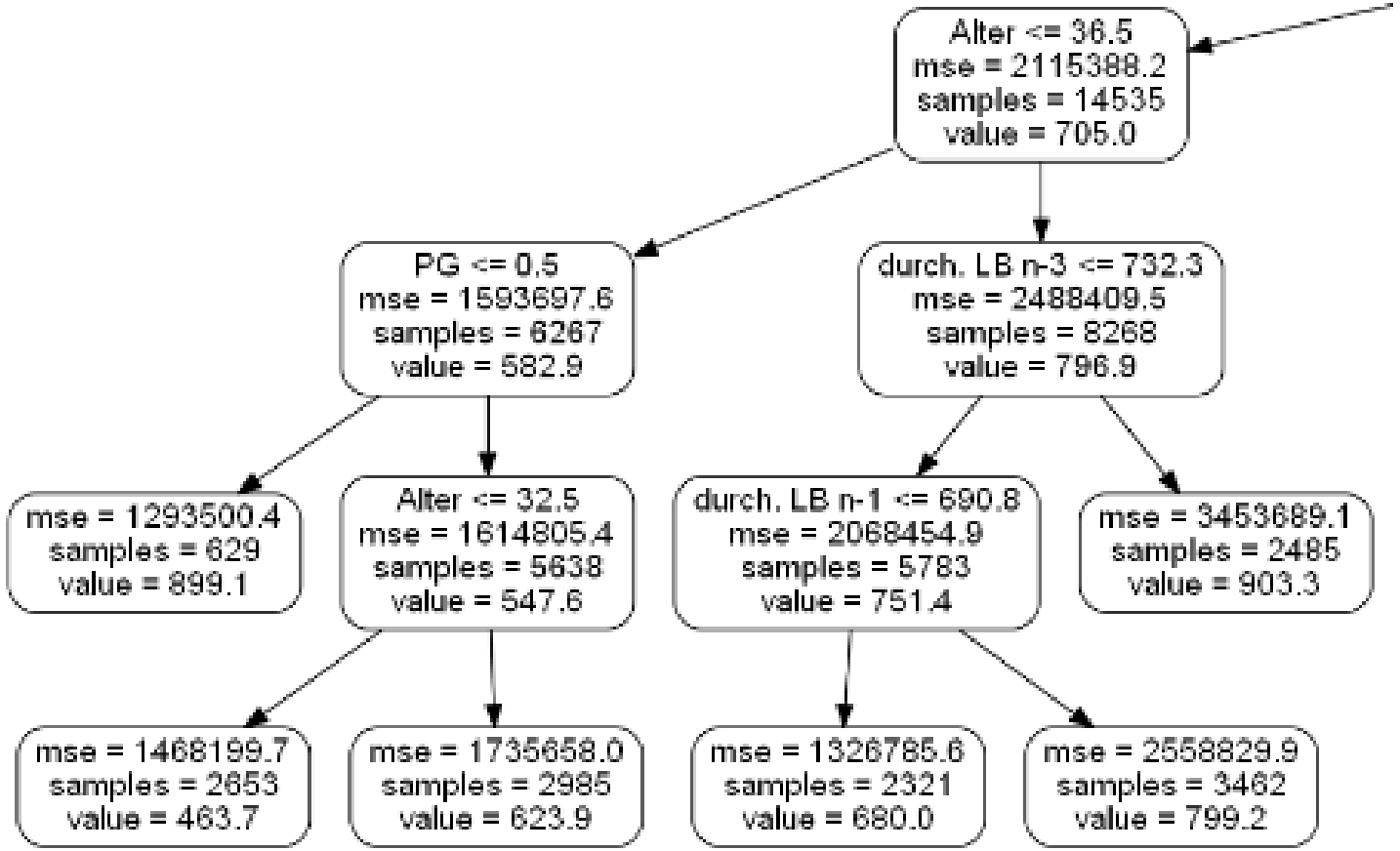
Modellierung

Entscheidungsbaum



Modellierung

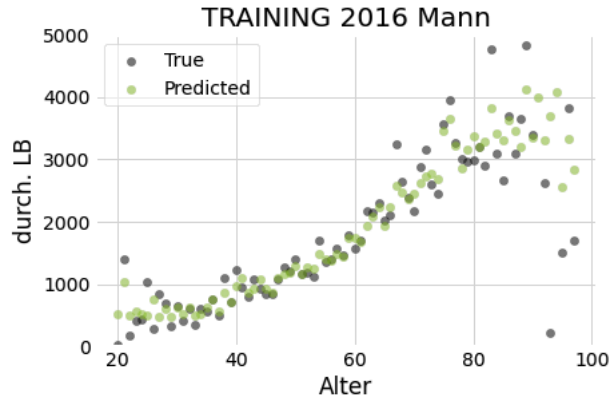
Ausschnitt Entscheidungsbaum



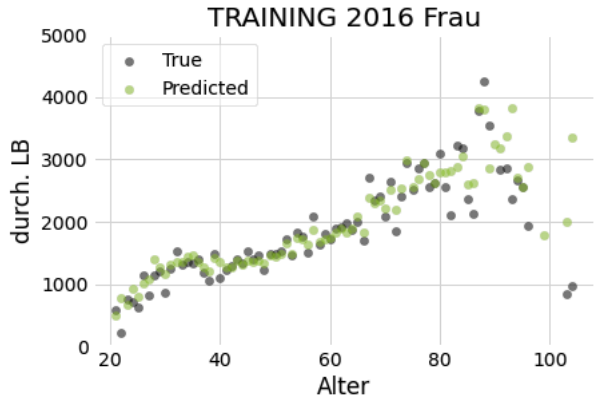
Ergebnisse

Vorhersage pro Geschlecht für die beiden Test-Jahre

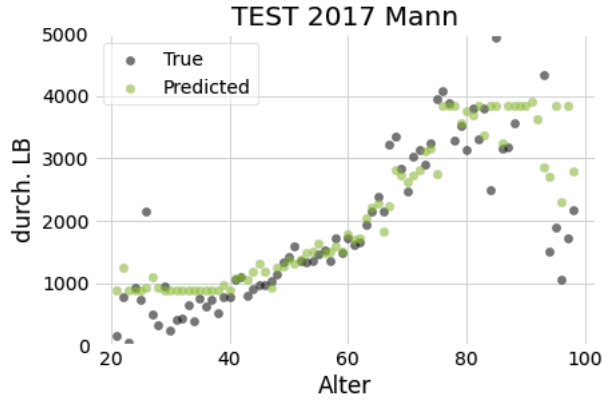
Durchschnittlicher Leistungsbetrag für das Trainingsjahr 2016 bezüglich Alter für das Geschlecht Mann



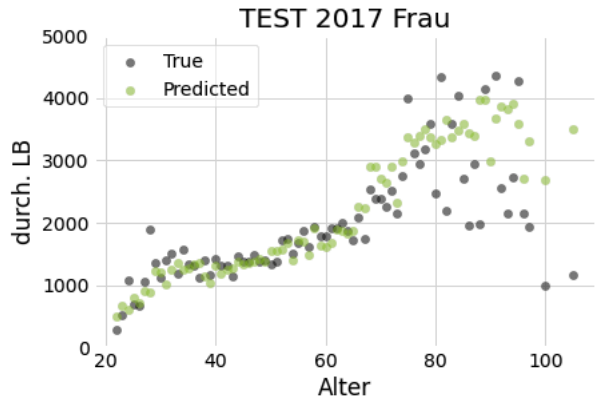
Durchschnittlicher Leistungsbetrag für das Trainingsjahr 2016 bezüglich Alter für das Geschlecht Frau



Durchschnittlicher Leistungsbetrag für das Testjahr 2017 bezüglich Alter für das Geschlecht Mann



Durchschnittlicher Leistungsbetrag für das Testjahr 2017 bezüglich Alter für das Geschlecht Frau



Überblick:

- Summe Leistungshöhe 2017 tatsächlich: 53.023.750,89
- Summe Leistungshöhe 2017 Vorhersage: 53.456.852,43
- Differenz : 433.101,54
- Fehler: 0,82%

Fazit und Ausblick

Schadenvorhersage

Fazit und Ausblick