

Friedrich Loser, Techniker Krankenkasse

ADS-Update zu Anwendungsfällen, Prognosetools, Programmierhilfen und Effektmessung

Fachgruppe Actuarial Data Science, 26. April 2024

Agenda: Update zu/r

1. ADS-Anwendungsfälle und Prognosetools	S.
- Prognosetools	3/13
- Credit Scoring	4
- Schadenhäufigkeit	8
2. Programmier- und Analysehilfen	14
3. Effektmessung	21
Anhang	26

Prognosetools für tabulare Daten: Gradient Tree Boosting Machines

Top-Tools:

- ✓ XGBoost (2014)
- ✓ LightGBM (2016)
- ✓ CatBoost (2017)

Gemeinsamkeiten:

- ✓ Schnell
- ✓ Gut
- ✓ Günstig

Unterschiede:

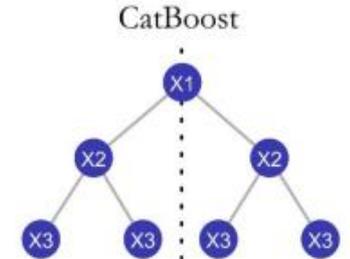
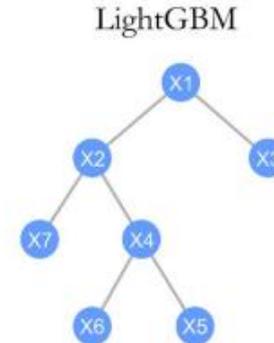
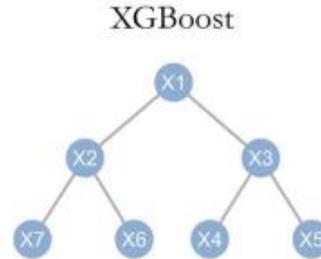
- Optimierungsverfahren
- Zusammenfassung von Kategorien
- Vorgehensweise beim Wachstum der Bäume:

Anwendungsgebiete:

- ✓ Klassifikation
- ✓ Regression
- ✓ Zeitreihen

Prinzip „Tree Boosting“:

1. Mit kleinem Baum starten
2. Zweiten Baum auf Basis der Residuen des ersten Baumes berechnen (Motto: Besser werden)
3. 2. mit „Lernrate“ ≤ 0.1 gewichten und zu 1. hinzufügen
4. Die Schritte 2. und 3. wiederholen (i.d.R. 100-1000 Iterationen)



Kann auch Textinput verarbeiten

Bildquelle (Vordergrund): <https://jcheminf.biomedcentral.com/articles/10.1186/s13321-023-00743-7>

1. Neues zu ADS-Anwendungsfällen

- Neuer Anwendungsfall 4: **Credit Scoring**
Thema: Prognose seltener Ereignisse (Klassifikation)
Autoren: Dr. Simon Hatzesberger und Friedrich Loser
Format: Python-Notebook
- Aktualisierung Anwendungsfall 1: **Schadenhäufigkeit**
Thema: KFZ-Tarifierung (Poisson-Regression)
Autoren: Daniel König und Friedrich Loser
Format: R-Notebook und neues Python-Notebook



Home Über uns Unsere Mitglieder Aktuar werden Unsere Themen Politik & Presse Forschung & Transfer

Lebensversicherung Anwendungsfälle

Altersversorgung Die Arbeitsgruppen des Ausschusses Actuarial Data Science erarbeiten verschiedene Anwendungsfälle, die nachfolgend vorgestellt werden.

Krankenversicherung

Schaden- / Unfallversicherung

ERM/Solvency II Prognose seltener Ereignisse: Credit Scoring

Investment Ausschuss Actuarial Data Science, 06.02.2024

Rechnungslegung und Regulierung Lässt sich mit Hilfe von Machine-Learning-Verfahren prognostizieren, ob ein Kunde stornieren wird, einen Schaden melden wird, einen Kredit zurückzahlen kann oder nicht? Und schneiden diese Methoden besser ab als klassische statistische Verfahren wie beispielsweise eine logistische Regression? Diese und weitere Fragen werden im Rahmen des vorliegenden Use Cases zur Binärklassifikation adressiert und auf Basis einer umfangreichen Analyse beantwortet. Konkret werden einerseits verschiedene Machine-Learning-Verfahren (u. a. CatBoost, logistische Regression ohne und mit Regularisierung, künstliche neuronale Netze, LightGBM & XGBoost) hinsichtlich ihrer Prognosegüte miteinander verglichen. Im Fokus liegen darüber hinaus auch Themen wie Datenvorverarbeitung, Erklärbarkeitsmethoden, Over- und Underfitting sowie Hyperparameter-Tuning. Das Ziel des Use Cases besteht darin, einen ersten Zugang zur Anwendung von Methoden der Actuarial Data Science anhand eines Supervised-Learning-Problems aufzuzeigen.

Verbraucherschutz

Actuarial Data Science

Anwendungsfälle

Data Science Challenge

Aktuarielles Glossar

Weitere Informationen

Quelle: <https://aktuar.de/unsere-themen/big-data/anwendungsfaelle/Seiten/default.aspx>

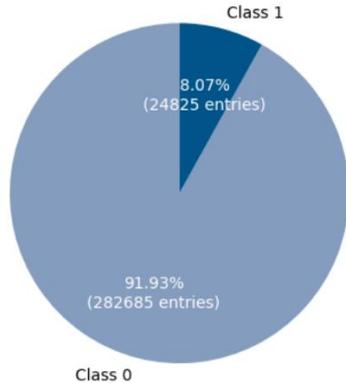
Diese Anwendungsfälle werden im Folgenden beschrieben

Credit Scoring (1): Überblick

Die Daten:

Prognose-Wettbewerbsdaten (Kaggle):
„Home Credit Default Risk“, 2018

Die Zielgröße: „Zahlungsproblem“



120 Merkmale („Features“):

- Demographie
- Kredit und Einkommen
- Externe Quellen

Das Notebook:

Table of Contents

Part A: Quick & Easy

- [1. Developing an Initial Baseline Machine Learning Model](#)

Part B: Generating Insights from Data and Modeling

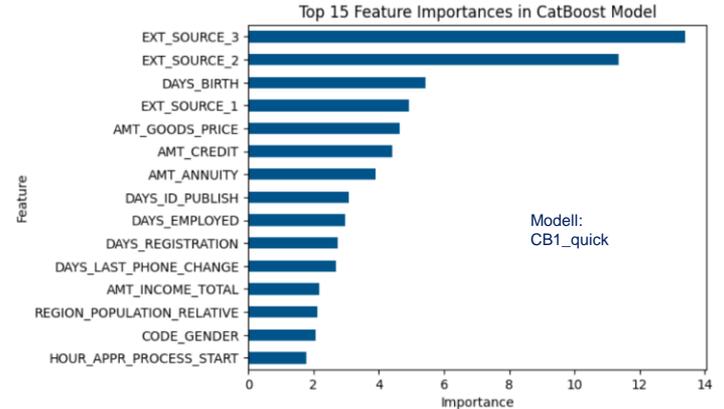
- [2. Logistic Regression: A Classical Approach to Classification](#)
- [3. Exploratory Data Analysis and Feature Engineering](#)
- [4. Enhancing CatBoost's Performance with New Features](#)
- [5. Interpreting Models for Better Understanding](#)
- [6. Data Preprocessing: Encoding, Scaling, and Subsampling](#)

Part C: Tuning and Applying Machine Learning Models

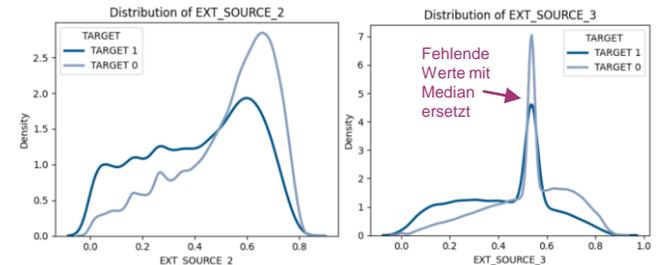
- [7. Regularized Logistic Regression to Prevent Overfitting](#)
- [8. Optimizing a Feed-Forward Neural Network](#)
- [9. Hyperparameter Optimization in CatBoost](#)
- [10. HP-Tuning LightGBM and XGBoost](#)
- [11. Model Evaluation and Application](#)
- [12. Key Insights into Model Performance and Binary Classification](#)

Appendix

Die wichtigsten Merkmale (nach 1.):



Deren Werteverteilung (ab 2.):

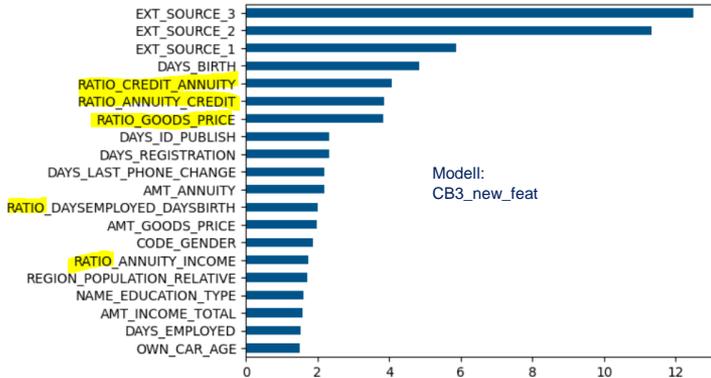


Credit Scoring (2): Merkmale und Modelle

„Feature Engineering“:

```
df_pre['RATIO_CREDIT_ANNUITY'] = df_pre['AMT_CREDIT'] / df_pre['AMT_ANNUITY']
df_pre['RATIO_GOODS_PRICE'] = df_pre['AMT_CREDIT'] / df_pre['AMT_GOODS_PRICE']
df_pre['RATIO_CREDIT_INCOME'] = df_pre['AMT_CREDIT'] / df_pre['AMT_INCOME_TOTAL']
df_pre['RATIO_ANNUITY_INCOME'] = df_pre['AMT_ANNUITY'] / df_pre['AMT_INCOME_TOTAL']
df_pre['RATIO_ANNUITY_CREDIT'] = df_pre['AMT_ANNUITY'] / df_pre['AMT_CREDIT']
df_pre['RATIO_DAYEMPLOYED_DAYS_BIRTH'] = df_pre['DAYS_EMPLOYED'] / df_pre['DAYS_BIRTH']
```

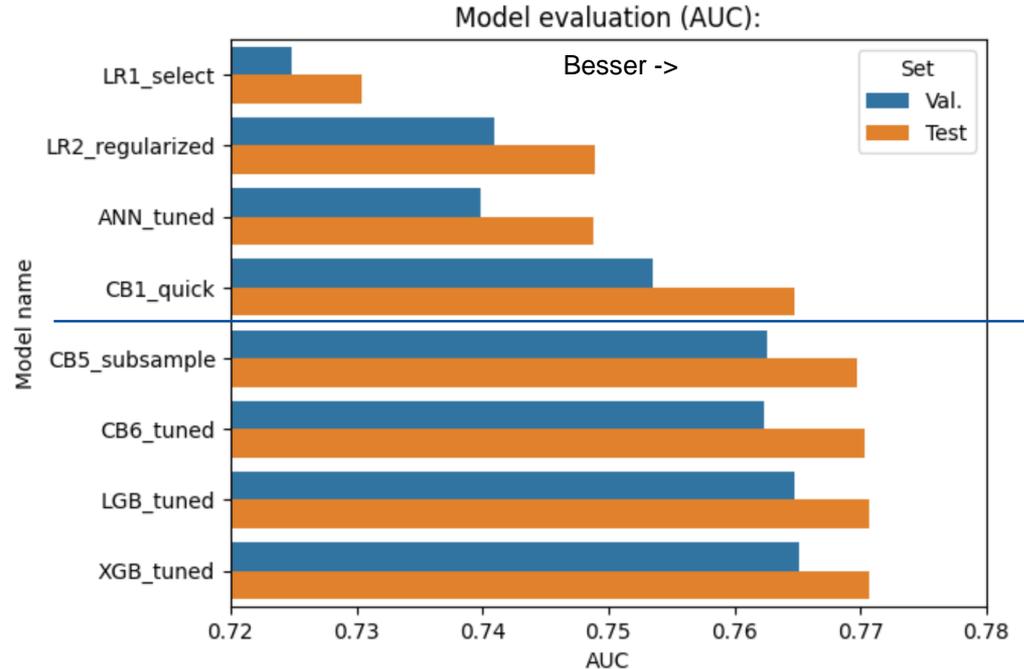
CatBoost mit den neuen Merkmalen:



- ✓ Drei neue Merkmale erweisen sich als wichtig
- ✓ Führt zu entscheidender Modellverbesserung

Quelle: <https://www.kaggle.com/code/floser/binary-classification-credit-scoring>

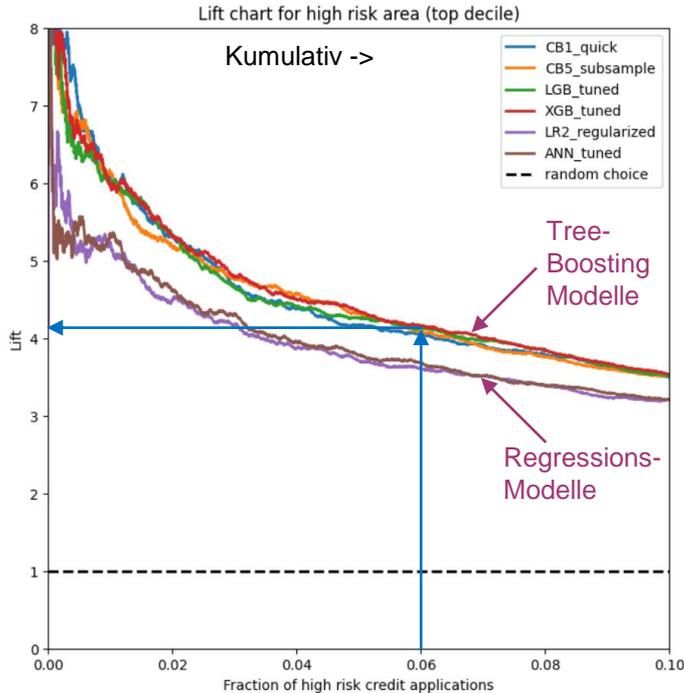
Modellgüte im Überblick:



- Modellgüte auf unbesehenen Testdaten (wegen zufälliger Stichprobenunterschiede) jeweils am Besten

Credit Scoring (4): Anwendung und Erkenntnisse

Betrachtung im „Hochrisikogebiet“:



6% Anträge mit den höchsten Prognosewahrscheinlichkeiten beinhalten 25% der Kreditausfälle ($6\% \cdot \text{Lift-Faktor}$)

Die wichtigsten Erkenntnisse für derartige Aufgabenstellungen:

1. CatBoost ermöglicht sehr einfach sehr gute Prognosemodelle
2. Feature Engineering ist entscheidender Schritt für weitere Prognoseverbesserungen
3. Rechenaufwändiges Hyperparameter-Tuning ermöglicht weitere, meist kleinere Prognoseverbesserungen
4. Subsampling (z.B. 1:3) ermöglicht wesentlich schnellere und etwas bessere Prognosemodelle

Übertragbar auf ähnliche Prognosen mit unbalancierten Daten:

- Storno
- Kauf
- Schaden
- Betrugserkennung

Schadenhäufigkeit (1): Wesentliche Änderung der Datengrundlage

Die Datenquelle: freMTPL2freq (aus R-Package CASdatasets)

	IDpol	ClaimNb	Exposure	Area	VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Density	Region
	<dbl>	<int>	<dbl>	<fct>	<int>	<int>	<int>	<int>	<fct>	<fct>	<int>	<fct>
1	1	0	0.10	D	5	0	55	50	B12	Regular	1217	R82
2	3	0	0.77	D	5	0	55	50	B12	Regular	1217	R82
3	5	0	0.75	B	6	2	52	50	B12	Diesel	54	R22
4	10	0	0.09	B	7	0	46	50	B12	Diesel	76	R72

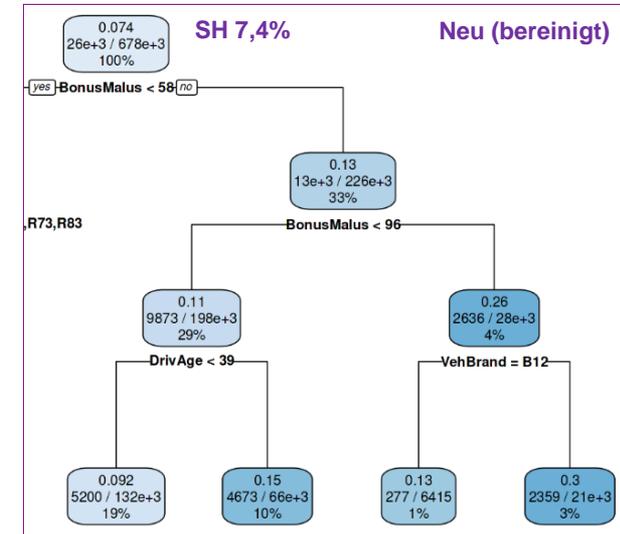
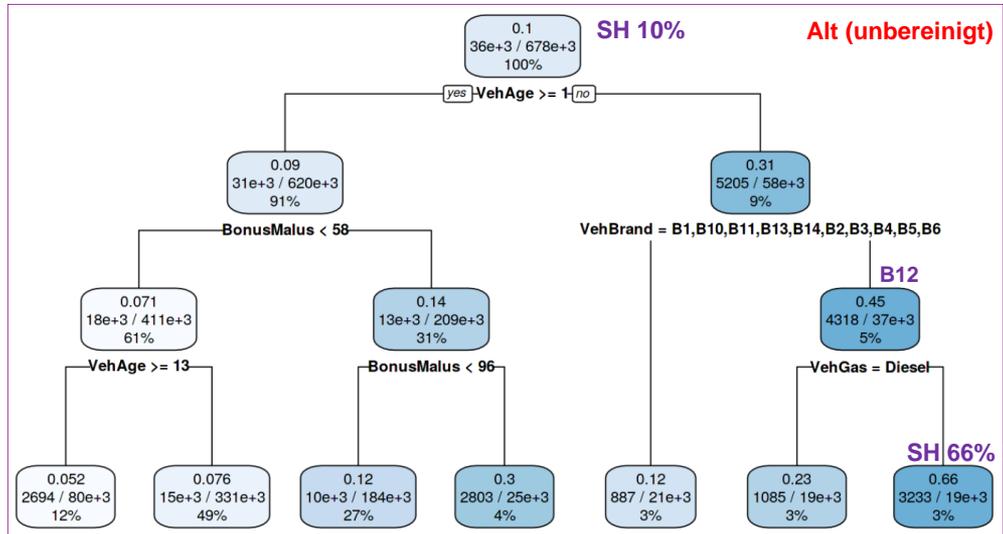
Σ 36.056 Schäden

Die Schadendatei freMTPL2sev:

	IDpol	ClaimNb
	<int>	<dbl>
1	139	1
2	190	1
3	414	1
4	424	2

Neu: Schäden, die nicht in der
Schadenhöhendatei enthalten
sind, werden ausgeschlossen

Σ 26.406 Schäden (9.650 weniger, -27%)



Einfluss der
Bereinigung:
Erheblich!

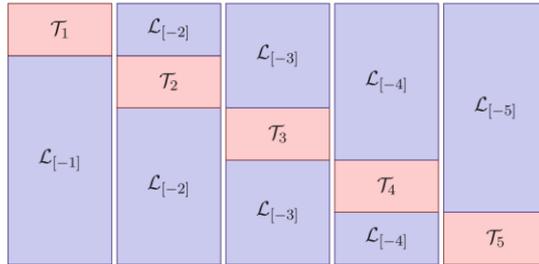
Für weitere
Infos siehe
Notebooks

Schadenhäufigkeit (2): Unterschiede zwischen den Notebooks

	R 2020	Python (2/2024)	R (3/2024)
Schadenbereinigung	Nein	Ja	Ja
Merkmalsbegrenzung	Wie Schelldorfer & Wüthrich 2019	Wie Schelldorfer & Wüthrich 2019	+ Kappung des Fahrzeugalters
Evaluationsmetrik: Poissonabweichung	Nicht gewichtet	Exposure gewichtet	Exposure gewichtet
Verallgemeinerte lineare Modelle (GLM)	GLM 2, 4, 5	GLM 2, 4, 5	GLM 1, 2, 4, 5
GLMM (Mixed Model, Credibility-Ansatz)	Für „B12RN“	-	-
Regularisierte GLMs (LASSO, Ridge Reg.)	-	-	Ja
Verallgemeinerte Additive Modelle (GAM)	-	-	Ja
Neuronale Netze: Architektur	Wie Schelldorfer & Wüthrich 2019	Portierung von Axel Kaiser	R-Code geändert wg. Fehlermeldung
Neuronale Netze: Embedding Plots	-	Ja	Ja
GBM: XGBoost ergänzt um CatBoost, LightGBM	-	Ja	Ja
Hyperparametertuning GBM-Modelle	-	Ja	- (HPs aus Python)
Boxplots (Schwankungsbreiten)	Ja	-	Ja

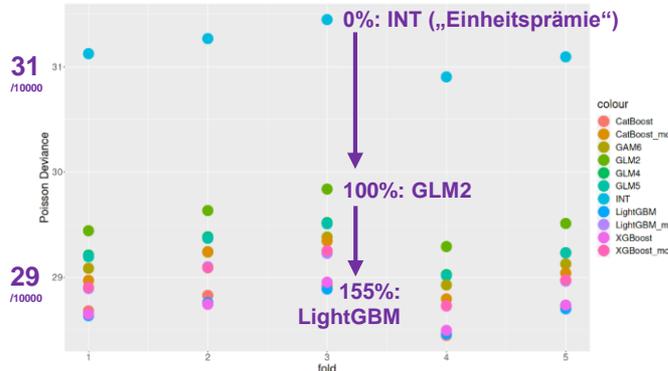
Schadenhäufigkeit (3): Kreuzvalidierung und Ergebnis

5-fache Kreuzvalidierung: Datenaufteilung

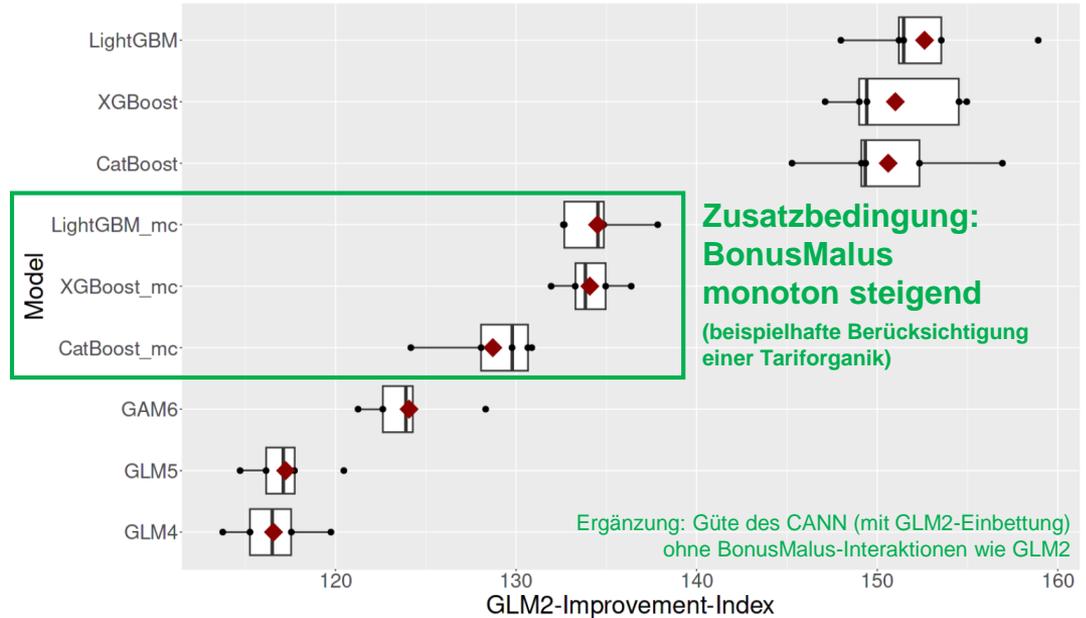


Bildquelle: Wüthrich und Merz (2023), "Statistical Foundations of Actuarial Learning and its Applications", <https://link.springer.com/book/10.1007/978-3-031-12409-9>

5-fache Kreuzvalidierung: Poissonabweichung



Das Verbesserungspotential ggü. GLM2 (Alterspolynom, aber ohne Interaktionen):



- ✓ Beste Prognosegüte mittels LightGBM (mit/ohne Beschränkung)
- ✓ GAM mit (nur) einer Interaktion bestes klassisches Modell

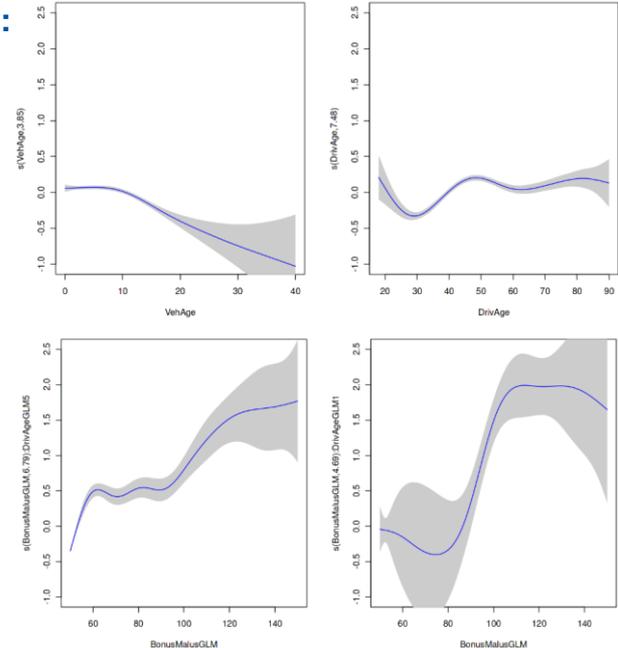
Schadenhäufigkeit (7): „5. Back to GLM and GAM: Interactions“

Table of Contents

- 5.1 Interactions with EIX
- 5.2 Interactions with SHAP
- 5.3 Friedmann's H-statistic
- 5.4 Back to GLM and GAM
 - 5.4.1 "GLM4" featuring neural net residual analyses
 - 5.4.2 "GLM5" Improved GLM with Interactions from XGBoost and LightGBM
 - 5.4.3 "GAM6" Improved GAM with an Interaction
- 6. Cross Validation and Boxplots: Results

**Ergebnis:
Starke Interaktion
zwischen DrivAge
und BonusMalus**

GAM-Splines:



+ fünf
weitere

BonusMalus-Splines zeigen kein monoton steigendes Verhalten (wie auch die Polynome bei GLM4 und GLM5)

Schadenhäufigkeit (8): Zusammenfassung

- Datenbereinigung führt zu geringeren Prognosegüten (starkes Signal fehlt)
- Neuronale Netze und klassische Modelle hier vergleichbar gut
- GAM schneidet deutlich besser als GLM (mit Polynomen) ab
- Gradient-Boosting-Tools erzielen (weiterhin) mit Abstand die genauesten Schadenprognosen
- LightGBM ist am Schnellsten und am Besten

3.5.2 HP-Tuning XGBoost: Fight Against Over-Fitting :-)



(Basis: DALLE3-generiertes Bild, siehe Python-Notebook)

Prognosetools für Text und Bild: Gigantische neuronale Netze („KI“)

Modellgröße GPT-3:
(Basis von ChatGPT 2022)

Vergleich mit neuronalem Netz zur
Schadenhäufigkeitsmodellierung
mit 792 (statt 175 Mrd.) Gewichten:

Regional-Embedding:

- 22 (statt 50.257) Ausprägungen
- 2 (statt 12.288) Dimensionen

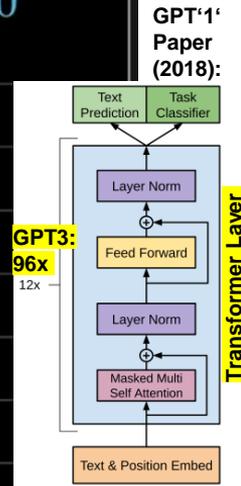
Zur Transformer-Technologie
siehe auch Herbsttagung 2022:

Quelle: Ferraris & Nagelin (2020), „The Art of Natural Language Processing: Classical, Modern and Contemporary Approaches to Text Document Classification“.

Quelle: „Attention Is All You Need“ (2017): **Transformers**, <https://arxiv.org/abs/1706.03762>

Über 80.000 vortrainierte Sprachmodelle verfügbar

GPT-3		Total weights:
Embedding	$12,288 \cdot 50,257$ $d_embed * n_vocab$	$= 617,558,016$
Key	$128 \cdot 12,288 \cdot 96 \cdot 96$ $d_query * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Query	$128 \cdot 12,288 \cdot 96 \cdot 96$ $d_query * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Value	$128 \cdot 12,288 \cdot 96 \cdot 96$ $d_value * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Output	$12,288 \cdot 128 \cdot 96 \cdot 96$ $d_embed * d_value * n_heads * n_layers$	$= 14,495,514,624$
Up-projection	$49,152 \cdot 12,288 \cdot 96$ $n_neurons * d_embed * n_layers$	$= 57,982,058,496$
Down-projection	$12,288 \cdot 49,152 \cdot 96$ $d_embed * n_neurons * n_layers$	$= 57,982,058,496$
Unembedding	$50,257 \cdot 12,288$ $n_vocab * d_embed$	$= 617,558,016$
		Organized into 27,938 matrices



Aber was ist ein GPT? Visuelle Einführung in Transformers | Deep Learning, Kapitel 5

<https://www.youtube.com/watch?v=wjZfJX0v4M>

3Blue1Brown
6,08 Mio. Abonnenten

Agenda: Update zu/r

1. ADS-Anwendungsfälle und Prognosetools	S.
- Prognosetools	3/13
- Credit Scoring	4
- Schadenhäufigkeit	8
2. Programmier- und Analysehilfen	14
3. Effektmessung	21
Anhang	26

Programmierhilfen: Warum kann „KI“ so gut programmieren?

- **Transformer-Technologie: Gigantische künstliche neuronale Netze mit enormer Modellkapazität**
- **Programmiersprachen sind „Sprachen“ mit begrenztem Vokabular und klaren Regeln**
- **„Wissen“: Spezielle Sprachmodelle wurden mit umfangreichen Codes nachtrainiert, z.B.:**
 - **Codex beruht auf GPT-3 und wurde massiv mit GitHub-Codes „gefüttert“**
 - **GitHub Copilot verwendet Codex zur Code-Generierung (i.d.R. kostenpflichtig)**
- **„Kontext“: Programmierumgebungen (IDE) können Projektcode bereitstellen**
 - **Kenntnis des Projekts ermöglicht bessere Codevorschläge**
 - **Copilot z.B. in IDEs VS Code, Neovim und JetBrains integriert (i.d.R. kostenpflichtig)**

Programmierhilfen: 1. Konversation mit ChatGPT

Beispiel: ChatGPT erstellt R- und Python-Codes für Ereignisprognose

Siehe Artikel „Doppel-KI-Einsatz: Prognostizieren und Programmieren mit künstlicher Intelligenz“, [Der Aktuar 2/2023](#)

Erfahrungen:

- Überwiegend gute, teils lehrreiche Umsetzung („Sparringpartner“)
- Codes müssen in Programmierumgebung kopiert und dort ausgeführt werden
- Bei Fehlermeldungen ein hin und her zwischen Programmierumgebung und ChatGPT-Prompt
- Wenn KI einen Fehler nicht beheben kann, ist Spezialwissen nötig

Aus Datenschutzgründen ggf. auf Bezahlversion ChatGPT Team / Enterprise oder andere Anbieter ausweichen

Auszug:

„1. Daten einlesen und prüfen: Lesen Sie den Datensatz „application_train.csv“ ein, geben Sie die Anzahl der Spalten und Zeilen aus und zeigen Sie die ersten Datensätze an.“

Zu 1. generierter R-Code (mit Erläuterungen)

Leider kann ich als Text-basiertes Modell kein R-Code ausführen oder interaktive Notebooks erstellen. Allerdings kann ich den R-Code für Ihre Anforderungen bereitstellen. Bitte beachten Sie, dass Sie die erforderlichen Pakete installiert haben müssen, um diesen Code auszuführen.

1. Daten einlesen und prüfen

```
R  
library(readr)  
data = read_csv("application_train.csv")  
print(paste("Anzahl der Zeilen: ", nrow(data)))  
print(paste("Anzahl der Spalten: ", ncol(data)))  
head(data)
```

Ergebnis: Umsetzung gelungen. Der Datensatz umfasst 307.511 Kreditanträge und 122 Merkmale, darunter die aufsteigend sortierte Antragsnummer SK_ID_CURR und die Zielgröße TARGET (Zahlungsschwierigkeiten 1, sonst 0). Die 120 weiteren, beschreibenden Merkmale können als sogenannte „Features“ in die Modellierung einfließen.

Zu 2. generierter Python-Code

```
## 2. Fehlende Werte erkennen und ersetzen  
missing = df.isnull().sum() * 100  
print(missing[missing > 0])  
  
for col in df.columns:  
    if df[col].dtype == 'object':  
        df[col].fillna('Missing', inplace=True)  
    else:  
        df[col].fillna(df[col].median(), inplace=True)
```

Programmierhilfen: 2. Integration in Jupyter Notebook (kostenfrei)

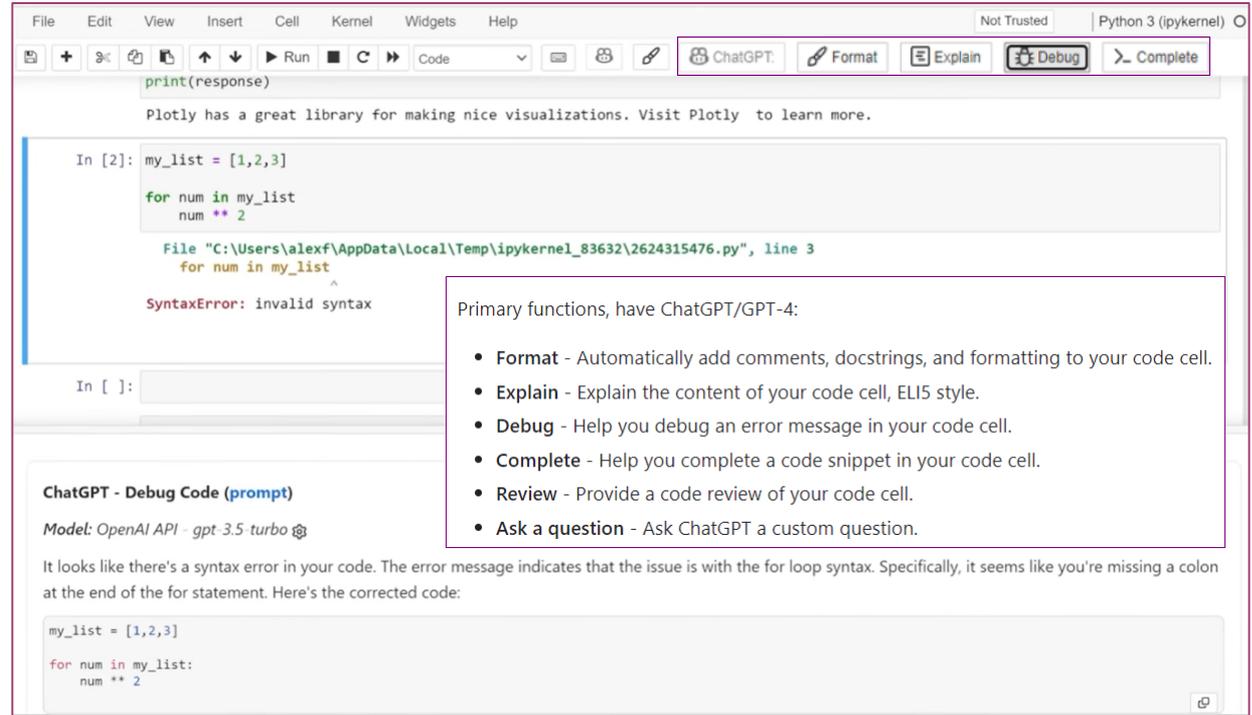
Notwendige Schritte:

1. Zugang zu ChatGPT
via `openai.api_key`

2. Browser-Erweiterung, z.B.
„ChatGPT for Jupyter“

<https://github.com/TiesdeKok/chat-gpt-jupyter-extension>

⚠ September 2023 Update: This repository is archived as I am unable to maintain it in my new role. I recommend using: <https://github.com/jupyterlab/jupyter-ai> as an alternative.



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for various actions. The main area contains a code cell with the following Python code:

```
print(response)

Plotly has a great library for making nice visualizations. Visit Plotly to learn more.

In [2]: my_list = [1,2,3]

for num in my_list
    num ** 2

File "C:\Users\alex\AppData\Local\Temp\ipykernel_83632\2624315476.py", line 3
    for num in my_list
        ^
SyntaxError: invalid syntax
```

Below the code cell, there is a ChatGPT extension interface. It shows a prompt: "ChatGPT - Debug Code (prompt)" and the model response: "Model: OpenAI API - gpt-3.5 turbo". The response explains the syntax error and provides the corrected code:

```
my_list = [1,2,3]

for num in my_list:
    num ** 2
```

A callout box on the right side of the screenshot lists the primary functions of the ChatGPT/GPT-4 extension:

- **Format** - Automatically add comments, docstrings, and formatting to your code cell.
- **Explain** - Explain the content of your code cell, ELI5 style.
- **Debug** - Help you debug an error message in your code cell.
- **Complete** - Help you complete a code snippet in your code cell.
- **Review** - Provide a code review of your code cell.
- **Ask a question** - Ask ChatGPT a custom question.

How to Integrate ChatGPT in Jupyter Notebooks for Free!

<https://www.youtube.com/watch?v=2izXQUoXyJA>

Programmierhilfen: 3. KI-Integration in Programmierumgebung

Beispielhafte Tätigkeiten: Kann

- ✓ Code kontextbezogen vervollständigen
- ✓ aus Kommentarzeile Code-Vorschlag erzeugen
- ✓ Code formatieren, erklären, korrigieren, dokumentieren und Änderungen (Commit) zusammenfassen
- ✓ Testkonzept und Testfälle erstellen

Beispiel: GitHub Copilot in VS Code (kostenpflichtig)

- **Potential: Beeindruckende Fähigkeiten** (siehe z.B. [Video](#))
- **Nutzen: Kann Nutzer-Zufriedenheit erhöhen und Produktivität verdoppeln** ([Quelle](#))
- **Gefahr: Kann zu schlechterer Code-Qualität führen** ([Quelle](#))
- **Verbreitung: KI-Unterstützung wird bereits von der Mehrheit der Entwickler verwendet** ([Quelle](#))

Pers. Einschätzung: KI-Unterstützung wird zu gängiger Erweiterung bereits vorhandener Programmierhilfen (in IDEs)

Datenanalyse mit KI (1): z.B. ChatGPT Data Analyst

Aus Datenschutzgründen ggf. auf Bezahlversion ChatGPT Team / Enterprise oder andere Anbieter ausweichen

You **Upload:**


archive.zip
Zip Archive

```

"Country", "Year", "Gender", "Age", "log_mortality"
"CHE", 1950, "Female", 0, -3.60112501992636
"CHE", 1950, "Female", 1, -5.79392602450917
... line 143219:
"USA", 2016, "Male", 100, -0.874973659924957
    
```

Preise compare the delopment of human mortality between the given countries

Data Analyst
To compare the development of human mortality between the countries in your dataset, I'll first need to extract and examine the contents of the uploaded file. Let's begin by unzipping the file and inspecting its contents.

The extracted zip file contains a file named "HMX.csv". I will load this CSV file to analyze the data and compare human mortality development across different countries. Let's proceed with loading and examining the contents of this file. [-] <- Python-Code

The dataset contains the following columns:

1. **Country:** Country code.
2. **Year:** Year of the data.
3. **Gender:** Gender (Male/Female).
4. **Age:** Age group.
5. **log_mortality:** Logarithm of the mortality rate.

You
Please calculate the life expectancy for a girl borne in 1950 vs 2010 for each country and visualize the results

Data Analyst
To calculate life expectancy for a girl born in 1950 versus 2010 in each country, we need to:

- Filter the data for females and for the years 1950 and 2010.
- Calculate life expectancy for each country for these two years.

Life expectancy at birth can be estimated using mortality rates across different age groups. Specifically, life expectancy is the average number of years a newborn is expected to live if current mortality rates continue to apply.

I'll proceed with these calculations and then visualize the results. The life expectancy for a girl born in 1950 versus 2010 has been calculated for each country in your dataset. However, it appears that not all countries have data for both years. For example, Germany ("DEUT", "DEUTE" ↓ d "DEUTW") only has data for 2010.

Ergebnis: Lebenserwartung viel zu hoch!

Ursache: log_mortality „falsch“ interpretiert und via 10() statt exp() in mortality ungerechnet**

You
Firstly, he data field log_lortality contains the natural logarithm of the mortality rate. Thus the exponent-function should be used to get the mortality rate.

Data Analyst
Thank you for clarifying that the "log_mortality" field represents the natural logarithm of the mortality rate. To correctly calculate the mortality rates, we should use the exponent function (exp) instead of the base-10 exponentiation I used earlier.

...

Country	Year	Life Expectancy
JPN	1950	61.04 years
JPN	2010	86.45 years
USA	1950	71.23 years
USA	2010	81.41 years

These values are more in line with expected life expectancy figures for those periods and countries. The data now shows a significant increase in life expectancy from 1950 to 2010 in both Japan and the USA, which aligns with historical trends in improved healthcare and living conditions. [-]

Ergebnis: Korrekt

Schlägt Visualisierungen, Vergleiche etc. vor

Datenanalyse mit KI (2): ChatGPT Data Analyst verarbeitet pdf

PKV-Sterbetafel 2010

Alter	qx	qy	Alter	qx	qy
0	0,000273	0,000255	52	0,001954	0,001428
1	0,000273	0,000255	53	0,002205	0,001560
2	0,000273	0,000224	54	0,002477	0,001702
3	0,000143	0,000112	55	0,002774	0,001859
4	0,000068	0,000054	56	0,003087	0,002034

Die Grafik zeigt die Sterbewahrscheinlichkeiten (qx) in Abhängigkeit vom Alter. Als nächstes speichere ich die Daten in eine CSV-Datei. [-]

Die Daten wurden erfolgreich in eine CSV-Datei gespeichert. Sie können die Datei hier herunterladen: PKV Sterbetafel 2010 qx CSV. [-]

You ... 81 ... 102

dl_st_2010_pkv_sterbetafel_va.pdf PDF

← Upload

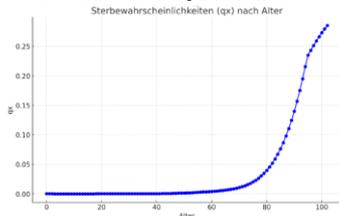
Bitte die Tabellenwerte für qx auf Seite 2 einlesen, grafisch darstellen und in eine csv-Datei schreiben.

Data Analyst
Ich werde nun die Werte für qx aus der Tabelle extrahieren, grafisch darstellen und in eine CSV-Datei speichern.

Zuerst extrahiere ich die Daten aus der PDF-Datei. Die Daten für qx wurden erfolgreich extrahiert. Hier sind die ersten fünf Zeilen der Tabelle:

Alter	qx
0	0,000273
1	0,000273
2	0,000273
3	0,000143
4	0,000068

Nun werde ich diese Daten grafisch darstellen. [-]



Nun werde ich qy grafisch darstellen. [-]



Die Grafik zeigt die Sterbewahrscheinlichkeiten (qy) in Abhängigkeit vom Alter. Als letzter Schritt speichere ich diese Daten in einer CSV-Datei. [-]

Die vollständigen Daten, einschließlich qx und qy , wurden in eine CSV-Datei gespeichert. Sie können die Datei hier herunterladen: PKV Sterbetafel 2010 Vollständige CSV. [-]

You
Bitte aus den Sterbewahrscheinlichkeiten die Lebenserwartung für Männer (qx) und Frauen (qy) berechnen

Data Analyst
Um die Lebenserwartung zu berechnen, werde ich die Sterbewahrscheinlichkeiten qx und qy verwenden. Die Methode zur Berechnung der Lebenserwartung basiert auf der Summe der Überlebenswahrscheinlichkeiten von jedem Alter bis zum Ende des Datensatzes. Die Formel für die Lebenserwartung e_x für ein bestimmtes Alter x lautet:

$$e_x = \sum_{k=0}^{T-x} l_{x+k} \cdot \frac{1}{l_x}$$

wobei l_x die Anzahl der Überlebenden im Alter x und T das höchste Alter in der Tabelle ist. l_x kann aus den Sterbewahrscheinlichkeiten berechnet werden, indem man beginnt mit $l_0 = 100,000$ (eine übliche Annahme für Lebensversicherungen) und dann für jedes darauf folgende Jahr:

$$l_{x+1} = l_x \cdot (1 - q_x)$$

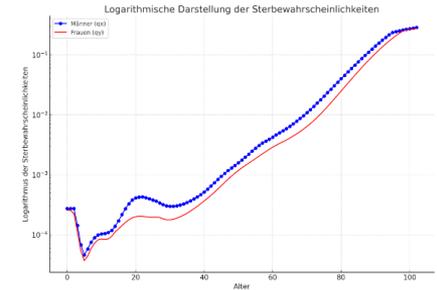
Ich werde diese Berechnung für beide Geschlechter (Männer basierend auf qx und Frauen basierend auf qy) durchführen. Die Lebenserwartung wurde für Männer (basierend auf qx) und Frauen (basierend auf qy) berechnet. Hier sind die ersten fünf Zeilen der Tabelle, die die Lebenserwartung von der Geburt an (Alter 0) zeigen:

Alter	Lebenserwartung Männer	Lebenserwartung Frauen
0	84.35 Jahre	87.78 Jahre
1	83.38 Jahre	86.61 Jahre
2	82.40 Jahre	85.83 Jahre
3	81.42 Jahre	84.85 Jahre
4	80.43 Jahre	83.86 Jahre

Möchten Sie diese Ergebnisse in einer CSV-Datei speichern oder weitere Analysen durchführen? [-]

You
Bitte die Ergebnisse in einer Excel-Datei speichern und die Sterbewahrscheinlichkeiten von Männern und Frauen in einer Grafik logarithmisch darstellen.

Data Analyst



Die Ergebnisse wurden erfolgreich in einer Excel-Datei gespeichert. Sie können die Datei hier herunterladen: PKV Sterbetafel 2010 Lebenserwartung Excel.

Die Grafik zeigt die logarithmische Darstellung der Sterbewahrscheinlichkeiten für Männer (qx) und Frauen (qy). Diese Darstellung hebt die Unterschiede in den Sterberaten über verschiedene Altersgruppen hinweg hervor, besonders in höheren Altersbereichen. [-]

Ergebnis: Alles korrekt

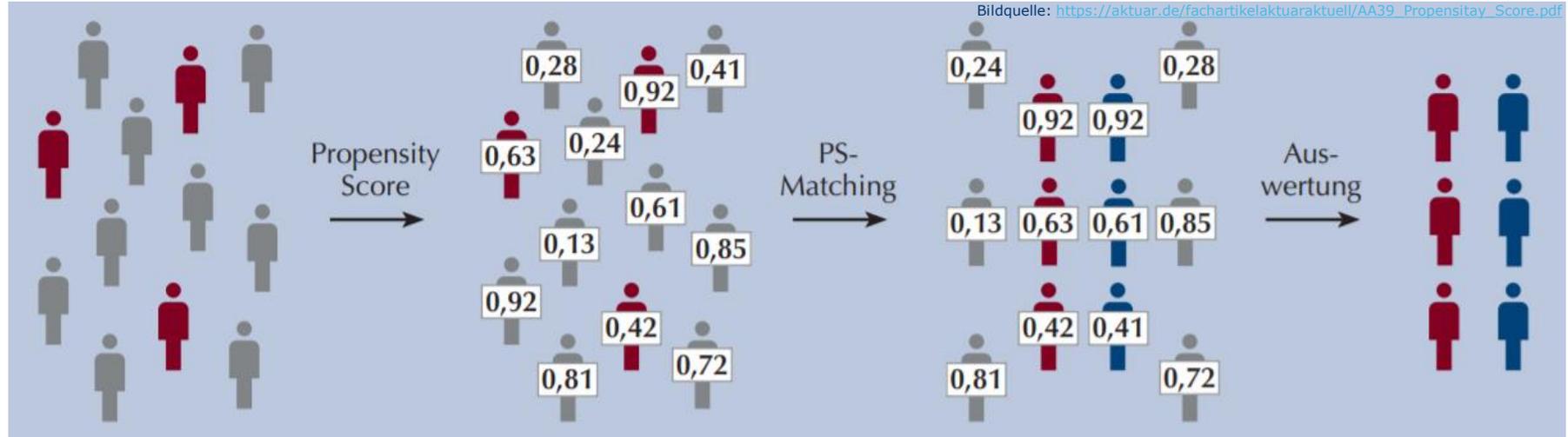
	A	B	C	D	E
1	Age	qx	qy	ex_men	ex_women
2	0	0,000273	0,000255	84,35338	87,78445
3	1	0,000273	0,000255	83,37615	86,80659
4	2	0,000273	0,000224	82,39864	85,82847
5	3	0,000143	0,000112	81,42087	84,84748
6	4	0,000068	0,000054	80,43237	83,85687

Agenda: Update zu/r

1. ADS-Anwendungsfälle und Prognosetools	S.
- Prognosetools	3/13
- Credit Scoring	4
- Schadenhäufigkeit	8
2. Programmier- und Analysehilfen	14
3. Effektmessung	21
Anhang	26

3. Effektmessung mittels Propensity Score Matching:

„Status quo“: Meist verbreitetes
Verfahren, siehe [Artikel](#)



Teilnehmer und Vergleichsgruppe selektieren, Datenaufbereitung

Propensity Score berechnen:

- Logistische Regression
- LASSO
- ML-Methoden

=> Teilnahmewahrscheinlichkeit

Paarbildung über Teilnahmewahrscheinlichkeit durchführen und Matching-Güte überprüfen

Maßnahmenerfolg messen, z.B. DID *

* Der Difference-In-Difference-Schätzer vergleicht die durchschnittliche Veränderung bei den Teilnehmern vor und nach der Maßnahme mit der entsprechenden durchschnittlichen Veränderung bei den Nicht-Teilnehmern

Neuere Entwicklungen:

- **Kritik am Propensity Score Matching:**
„Why Propensity Scores Should Not Be Used for Matching” ([King & Nielsen, 2019](#))
- **Gewichtungsverfahren „Entropy Balancing“** ([Hainmueller, 2012](#)) **gewinnt an Bedeutung**
- **Aktuelle Verfahrensvergleiche mit Gesundheitsdaten** ([Gansen, 2018](#)), ([Larrain & Groene, 2023](#))
- **Neu mit ML-Bezug: „Hierarchically Regularized Entropy Balancing“** ([Xu & Yang, 2022](#))

Gewichtungsverfahren mit Regularisierung

Umgesetzt (inkl. Parallelisierung) im R-Package hbal

Gutes Tutorial: <https://yiqingxu.org/packages/hbal/articles/tutorial.html>

“In the simplest set-up, user can just pass in {Treatment, X, Y}. The default settings will serially expand X to include higher order terms, hierarchically residualize these terms, perform double selection to only keep the relevant variables and use cross-validation to select penalties for different groupings of the covariates.”

Zusammenfassung

- **Neue Anwendungsfälle (Python/R) stehen für Interessierte bereit**

<https://aktuar.de/unsere-themen/big-data/anwendungsfaelle/Seiten/default.aspx>

- **Gradient Boosting für „Risikodaten“ bestens geeignet (XGBoost, LightGBM, CatBoost)**
- **KI-Assistenz zeigt starkes Potential bei Programmierung und Datenanalyse**
- **Neuer, erfolgversprechender ML-Ansatz bei Effektmessung**

**Vielen Dank
für Ihre
Aufmerksamkeit.**

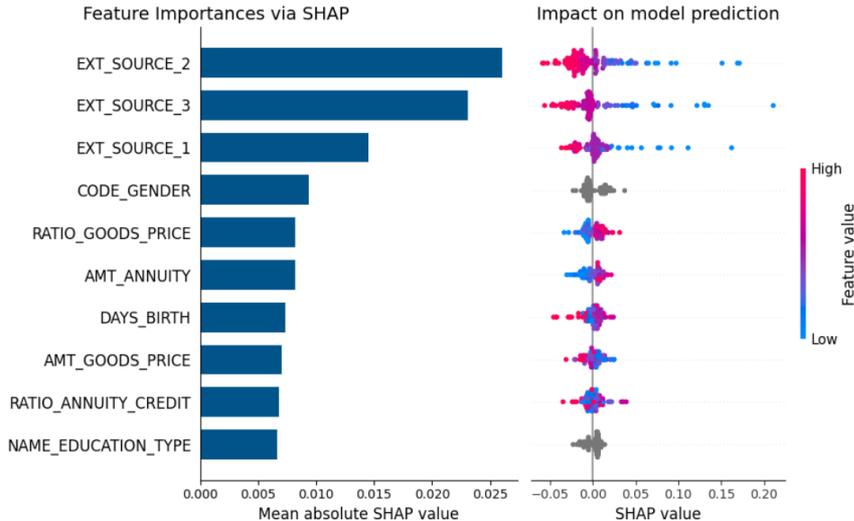
Friedrich Loser, Techniker Krankenkasse
FC, Bramfelder Straße 140, 22305 Hamburg

Agenda: Update zu/r

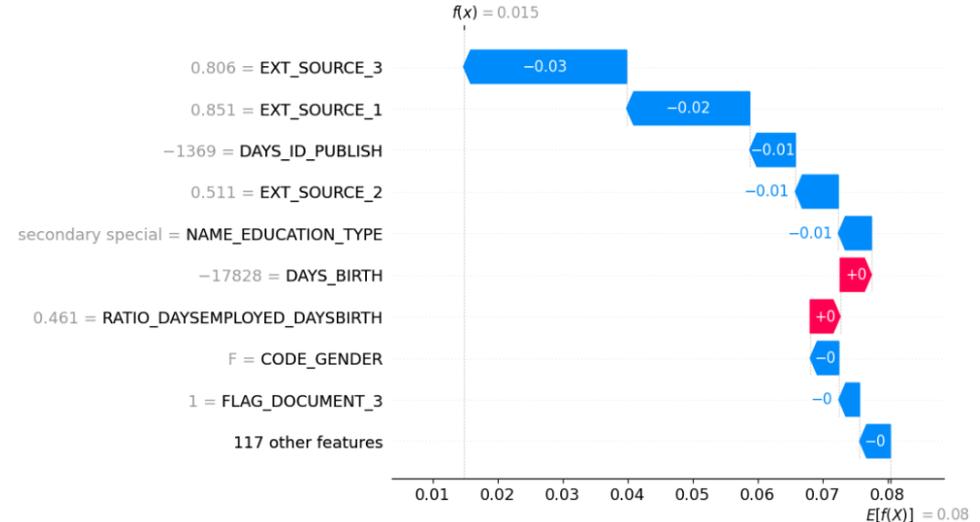
1. ADS-Anwendungsfälle und Prognosetools	S.
- Prognosetools	3/13
- Credit Scoring	4
- Schadenhäufigkeit	8
2. Programmier- und Analysehilfen	14
3. Effektmessung	21
Anhang	26

Credit Scoring (3): Interpretierbarkeit mittels SHAP

Global: Die wichtigsten Merkmale



Lokal: Fallbezogen via Wasserfallplots



- ✓ SHAP hat solide spieltheoretische Basis
- ✓ SHAP erlaubt globale und lokale Aussagen
- kernelSHAP ist langsam und ignoriert Merkmalsabhängigkeiten

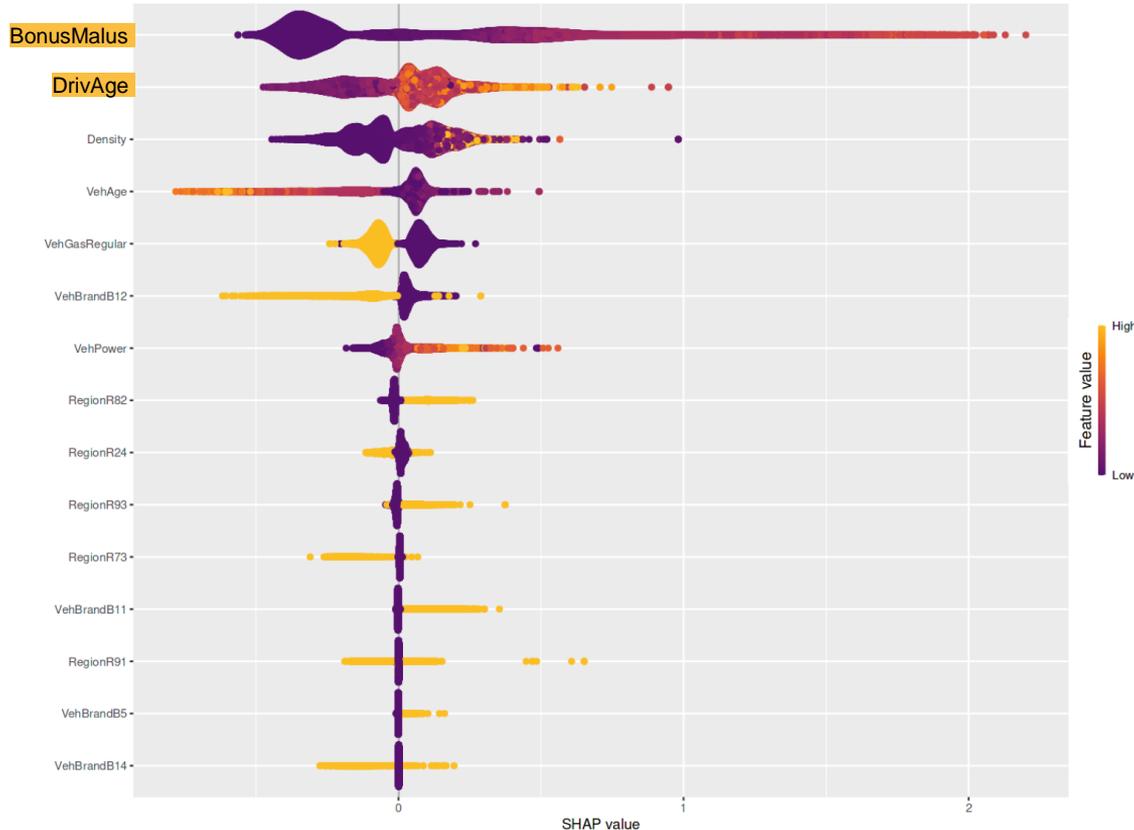
Wichtiger Unterschied zu Logistischer Regression bzw. davon abgeleiteter Score-Karten: Vereinfachtes Modell des Modells. Ergebnis kann stark von der einbezogenen Umgebung abhängen.

Schadenhäufigkeit (4): Interpretierbarkeit mittels SHAP

Modell: LightGBM_mc
(mit Zusatzbedingung BonusMalus
monoton steigend)

Global:

Feature Importance

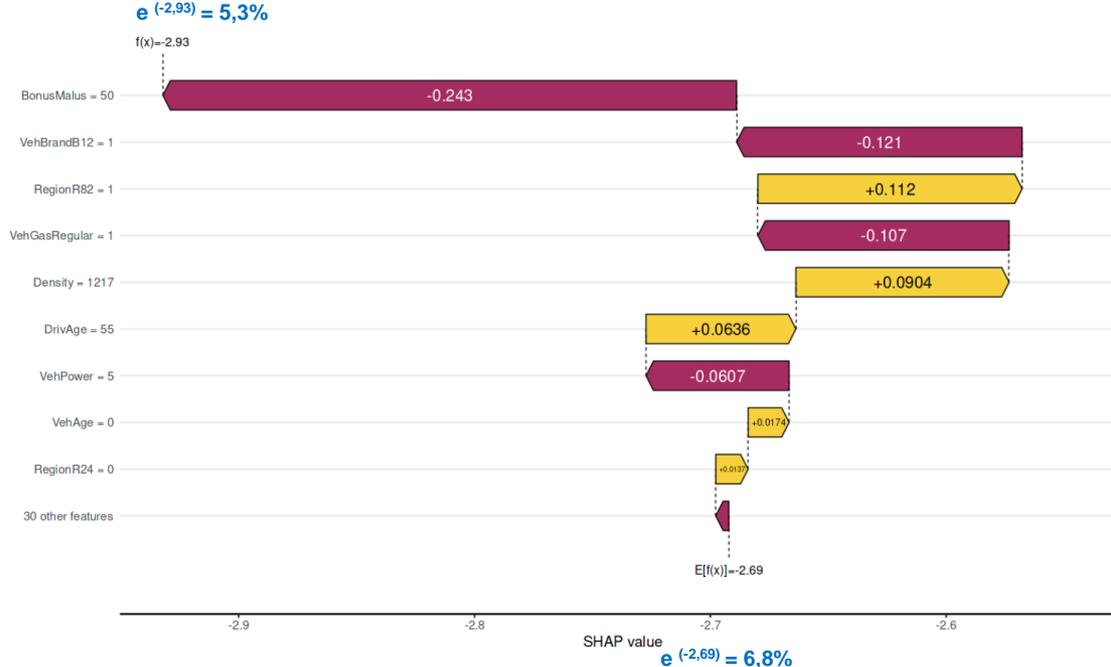


Schadenhäufigkeit (5): Interpretierbarkeit mittels SHAP

Lokal:

```
# Waterfall plot of first observation (log-scale)  
sv_waterfall(shap_lgb_test, row_id = 1)
```

Modell: LightGBM_mc
(mit Zusatzbedingung BonusMalus
monoton steigend)

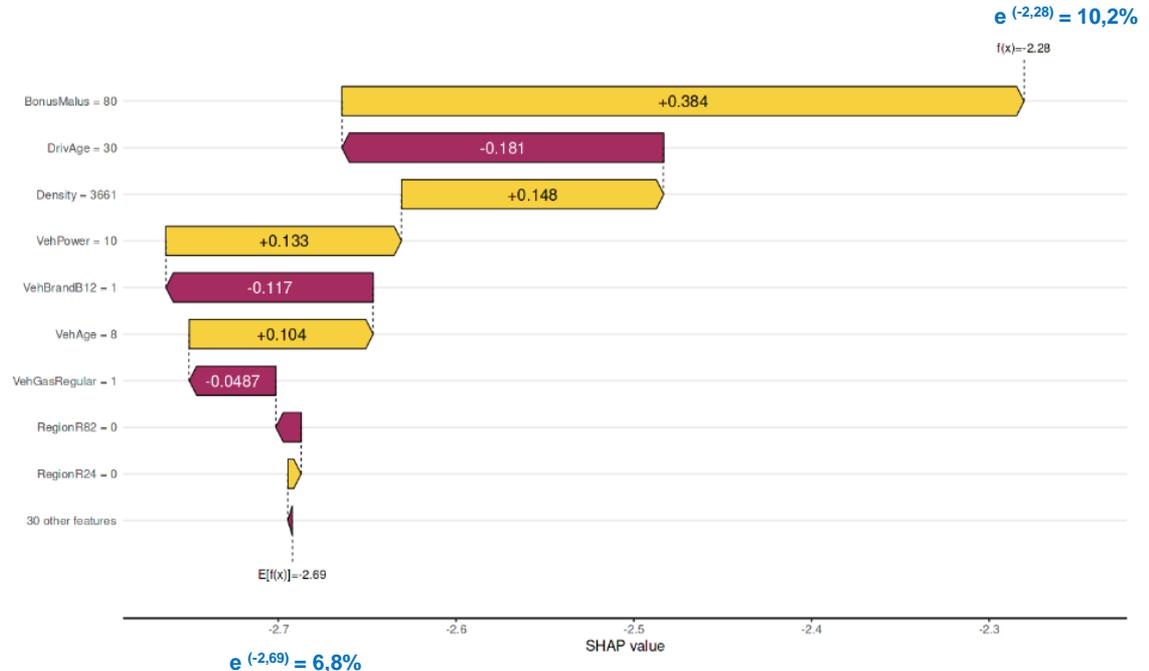


Schadenhäufigkeit (6): Interpretierbarkeit mittels SHAP

Modell: LightGBM_mc
(mit Zusatzbedingung BonusMalus
monoton steigend)

Lokal:

```
# Waterfall plot of eighth observation  
sv_waterfall(shap_lgb_test, row_id = 8)
```



Zu 3. Beispiel: Fallstudie Hausarztzentrierte Versorgung

TABELLE 1

Studienpopulation: Charakteristika der an der hausarztzentrierten Versorgung teilnehmenden Patienten (Interventionsgruppe, IG) im Vergleich zu Nicht-Teilnehmern (Nicht-IG) und zur Kontrollgruppe (KG)

Parameter	IG n = 40 298	Nicht-IG n = 322 732	relative Abweichung IG – Nicht-IG vor dem Matching (p-Wert)	KG n = 40 298	relative Abweichung IG – KG (p-Wert)
Geschlecht männlich	40,9 %	45,4 %	-11,2 % (< 0,001)	40,0 %	2,2 % (0,013)
Alter: Mittelwert ± SD;	64,8 ± 16,4;	56,9 ± 18,8;	12,2 %	65,1 ± 16,0;	-0,5 %

TABELLE 2

Vergleich der Versorgungskosten (Difference in Difference [DiD])

DiD:

Endpunkt	t1		t2		t2-t1		DiD	
	IG € (SD)	KG € (SD)	IG € (SD)	KG € (SD)	IG €	KG €	absoluter Interventi- onseffekt*1 (p-Wert [95%-KI])	relative Änderung*2 %
primärer Endpunkt								
durchschnittliche direkte Gesamtkosten*3	4 785 (8 048)	4 886 (8 485)	5 439 (9 648)	5 457 (10 029)	654	571	83 (0,215 [-48; 215])	1,7
sekundäre Endpunkte								
Kosten der Hausarztkonsultation	438 (246)	420 (258)	480 (278)	435 (278)	42	15	27 (< 0,001 [25; 30])	6,2
Kosten der Facharztkonsultation	551 (1 515)	590 (1 928)	529 (1 500)	546 (1 500)	-22	-44	22 (0,385 [1,3; 43])	4,1
Kosten der Arzneimittelverordnung	1 108 (2 974)	1 117 (2 321)	1 201 (3 406)	1 254 (3 406)	94	137	-44 (0,001 [-74; -13])	-3,9

- Propensity Score Matching:
Logistische Regression mit 222 Variablen
1:1-Matching ohne Zurücklegen
- Effektmessung über DiD-
Schätzer

„Schlussfolgerung:
Das Programm erreicht keine
Einsparungen bei den direkten
Gesamtkosten, kann aber
möglicherweise ...“

Auszüge aus Freytag A et al.,
„Effekte hausarztzentrierter Versorgung“,
Deutsches Ärzteblatt 25.11.2016

**Besuchen Sie
unsere Webseite**

www.aktuar.de
