

Ergebnisbericht

(gemäß Nr. 14.1 ANBest-IF)

Konsortialführung:	Universitätsklinikum Heidelberg
Förderkennzeichen:	01VSF18019
Akronym:	PREMISE
Projekttitel:	Predicting Re-admissions by Exploring Medication Inappropriateness and Sequence of prior Events in health insurance claims - Vorhersage von Rehospitalisierungen durch spezifische Medikationsrisiken und Ereignissequenzen
Autoren:	Prof. Walter E. Haefeli, PD Dr. Andreas D. Meid, Dr. Carmen Ruff
Förderzeitraum:	1. Februar 2019 – 31. Juli 2022

Inhaltsverzeichnis

I.	Abkürzungsverzeichnis.....	3
II.	Abbildungsverzeichnis.....	4
III.	Tabellenverzeichnis.....	5
1.	Zusammenfassung.....	6
2.	Beteiligte Projektpartner.....	7
3.	Projektziele.....	7
3.1	Hintergrund, Ausgangslage und Aufgabenstellung des Projektes.....	7
3.2	Ziele und Erwartungen des Projektes.....	9
4.	Projektdurchführung.....	10
5.	Methodik.....	11
5.1	Datenbasis.....	11
5.2	Klinische Definition von Index- und Rehospitalisierungsfall.....	11
5.3	Definition des Outcomes (Re-)Hospitalisierung.....	14
5.4	Medikations-basierte Analysen von Rehospitalisierungen bei Herzinsuffizienz und Herzinfarkt.....	16
5.5	Epidemiologie von Rehospitalisierungen und Charakterisierung der Erkrankungspopulationen.....	16
5.6	Vorhersage von Rehospitalisierungen in sechs Modellerkrankungen und Prüfung der Generalisierbarkeit.....	17
5.6.1	Kandidatenprädiktoren und deren Operationalisierung.....	17
5.6.2	Beschreibung der Strata von rehospitalisierten und nicht-rehospitalisierten Patienten.....	18
5.6.3	Entwicklung der Prädiktionsmodelle.....	20

5.6.4 Modellvergleiche von Verfahren aus dem Bereich künstliche Intelligenz und Machine-Learning	20
5.7 Verwendete Software	21
6. Projektergebnisse.....	21
6.1 Rehospitalisierungen in deutschen Abrechnungsdaten und Epidemiologie.....	21
6.2 Charakterisierung der Erkrankungspopulationen bezüglich Soziodemographie und Komorbiditäten	28
6.3 Vorhersage von Rehospitalisierungen für sechs Modellerkrankungen und Prüfung der Generalisierbarkeit	30
6.3.1 Modellvergleiche von Verfahren aus dem Bereich künstliche Intelligenz und Machine-Learning	35
6.3.2 Einfluss der Adhärenz mit leitliniengerechter Medikation in medikations-basierten Analysen von Rehospitalisierungen bei Herzinsuffizienz und Herzinfarkt	35
7. Diskussion der Projektergebnisse	38
8. Verwendung der Ergebnisse nach Ende der Förderung	41
9. Erfolgte bzw. geplante Veröffentlichungen	43
10. Literaturverzeichnis.....	43
11. Anhang	52
12. Anlagen.....	53

I. Abkürzungsverzeichnis

AM	Arzneimittel
AMI	Akuter Myokardinfarkt
AOK	Allgemeine Ortskrankenkasse
ARB	Angiotensin-Rezeptor-Blocker
ASS	Acetylsalicylsäure
ATC-Code	Anatomisch-Therapeutisch-Chemisches Klassifikationssystem
AUC	Area-Under-The-Curve
Bzw.	Beziehungsweise
CART	Classification and Regression Tree
COPD	Chronisch obstruktive Lungenerkrankung
COX-2	Cyclooxygenase-2
d. h.	Das heißt
DM	Diabetes mellitus, Typ 2
DRG	Diagnosis-Related-Groups
eGK	Elektronische Gesundheitskarte
EUR	Euro (€)
GC	Glucocorticoide
Ggf.	Gegebenenfalls
GINI CART	CART nach Variablenselektion mit GINI-Impurity
GINI GLM	GLM nach Variablenselektion mit GINI-Impurity
GKV	Gesetzliche Krankenversicherung
GLM	Generalisiertes lineares Modell
GLMBOOST	Boosting Variante für GLMs
HF	Herzinsuffizienz (heart failure)
ICD (-10-GM)	Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme (10. Revision, German Modification)
kNN	k-Nearest-Neighbors
KI	Konfidenzintervall
LASSO	Least-Absolute-Shrinkage-and-Selection-Operator
NEUR NET	Neuronales Netz
NSAR	Nichtsteroidale Antirheumatika
oGVD	Obere Grenzverweildauer
OR	Odds-Ratio
OS	Osteoporose
PDE-5	Phosphodiesterase-5
PERM CART	CART nach Variablenselektion mit Permutationsverfahren

Akronym: PREMISE

Förderkennzeichen: 01VSF18019

PERM GLM	GLM nach Variablenselektion mit Permutationsverfahren
RASI	Renin-Angiotensin-System-Inhibitoren
RF	Random Forest
RIDGE	Ridge-Regression
ROC-Kurve	Receiver-Operator-Characteristics-Kurve
Rp	Rezept
S/AF	Schlaganfall/Vorhofflimmern
SGB	Sozialgesetzbuch
START	Screening-Tool-to-Alert-to-Right-Treatment
STOPP	Screening-Tool-of-Older-People's-Prescriptions
TAH	Thrombozytenaggregationshemmer
TIA	Transitorische ischämische Attacke
u. a.	Unter anderem
UAW	Unerwünschte Arzneimittelwirkung
USA	United-States-of-America (Vereinigte Staaten von Amerika)
VKA	Vitamin K-Antagonisten
XGBOOST	eXtreme Gradient Boosting
z. B.	Zum Beispiel

II. Abbildungsverzeichnis

Abb. 1:	Mögliche Hospitalisierungsereignisse ausgelöst durch inadäquate Arzneimitteltherapie.	8
Abb. 2:	Schematische Darstellung des Projektablaufs.	11
Abb. 3:	Klinische Definition eines (nicht-)spezifischen Rehospitalisierungsfalles.	12
Abb. 4:	Spezielle Fallkonstellationen in Krankenkassendaten und deren Klassifizierung für das Projekt.	13
Abb. 5:	Vollständige Klassifizierung der Krankenhausfälle in Index- und spezifische Rehospitalisierungsfälle.	15
Abb. 6:	Flussdiagramm mit Darstellung der Datenbereinigung und Ergebnissen der Fallklassifizierung.	23
Abb. 7a:	Anteile von Index-Hospitalisierungen, welche für COPD bzw. Herzinsuffizienz rehospitalisiert wurden.	25
Abb. 7b:	Anteile von Index-Hospitalisierungen, welche für akuten Myokardinfarkt bzw. Diabetes mellitus Typ 2 rehospitalisiert wurden.	26
Abb. 7c:	Anteile von Index-Hospitalisierungen, welche für Schlaganfall, TIA oder Vorhofflimmern bzw. Osteoporose rehospitalisiert wurden.	27
Abb. 8:	Anteile der Index-Patientenpopulationen mit Komorbiditäten.	29
Abb. 9:	Anteil der Index-Fälle mit Exposition zu START-Kriterien.	31

Abb. 10:	Anteil der Index-Fälle mit Exposition zu STOPP-Kriterien.	32
Abb. 11:	Performance der entwickelten Vorhersagemodelle für spezifische Rehospitalisierungen bei sechs Modellerkrankungen.	34
Abb. 12:	ROC-Kurven der Vorhersagemodelle für Rehospitalisierungen nach akutem Myokardinfarkt (A) und Herzinsuffizienz (B) mit und ohne Betrachtung der Adhärenzprädiktoren.	37

III. Tabellenverzeichnis

Tabelle 1:	Regelwerk für die Definition von Rehospitalisierungen in sechs Modellerkrankungen.	13
Tabelle 2:	Indikationsspezifische STOPP-Kriterien.	19
Tabelle 3:	Indikationsspezifische START-Kriterien.	19
Tabelle 4:	Anzahl und Anteile von Krankenhausaufnahmen und -wiederaufnahmen der Jahre 2012-2016.	24
Tabelle 5:	Soziodemographische Charakterisierung der Erkrankungspopulationen anhand ihrer Index-Fälle.	28
Tabelle 6:	Relatives Abschneiden im Methodenvergleich der prädiktiven Performance als Rang unter verschiedenen Verfahren in sechs Modellerkrankungen.	36

1. Zusammenfassung

Hintergrund und Ziel: In Deutschland werden viele Patienten kurz nach einem Krankenhausaufenthalt erneut hospitalisiert. Neben anderen Gründen stellen Mängel in der Arzneimitteltherapie einen häufigen Auslöser für erneute Hospitalisierungen dar: Benötigte Arzneimittel werden schlecht vertragen, nicht richtig oder unregelmäßig eingenommen, falsch dosiert oder gar nicht verschrieben. Außerdem spielen vorliegende Krankheiten eine zentrale Rolle für (oftmals) vermeidbare Rehospitalisierungen. Obwohl Krankheitsinformation und Arzneimittelgebrauch routinemäßig von den gesetzlichen Krankenversicherungen (GKV) zur Abrechnung gespeichert werden, werden solche Daten bisher kaum genutzt, um das Risiko für eine Rehospitalisierung einzuschätzen und Patienten mit einem solchen Risiko gezielt zu erkennen. Aus diesem Grund soll für die wichtigsten Volkskrankheiten im PREMISE-Projekt geprüft werden, ob in Abrechnungsdaten der AOK Baden-Württemberg solche Risiken erkennbar sind und welche Vermeidungsstrategien daraus abgeleitet werden können.

Methodik: Unter Verwendung von epidemiologischen Ansätzen und mithilfe von Techniken aus dem Bereich der künstlichen Intelligenz („Machine-Learning“) sollten Risikomodelle entwickelt, getestet und optimiert werden, welche künftig helfen sollen, vermeidbare Risiken bei vulnerablen Patienten in ohnehin vorhandenen Krankenkassendaten zu entdecken.

Ergebnisse: Die Analyse von Rehospitalisierungen in GKV-Abrechnungsdaten konnte erfolgreich durchgeführt werden und die Ergebnisse gelten nach Vergleichen mit anderen Gesundheitssystemen als plausibel. Um alle relevanten Rehospitalisierungsereignisse zu erfassen, erschien die häufig genutzte Cut-Off-Periode von 30 Tagen als zu kurz. Insgesamt wurde etwa ein Drittel der Patienten innerhalb von 90 Tagen nach Entlassung wiederaufgenommen, wobei eine Unterscheidung zwischen geplanten oder ungeplanten Rehospitalisierungen nicht möglich war. Die Analyse von sechs verschiedenen Index-Erkrankungen (AMI: akuter Myokardinfarkt; COPD: chronisch-obstruktive Lungenerkrankung; DM: Diabetes mellitus Typ 2; HF: Herzinsuffizienz; OS: Osteoporose; S/AF: Schlaganfall, transitorische ischämische Attacke (TIA) oder Vorhofflimmern) ergab, dass der häufigste Grund für eine Rehospitalisierung in Verbindung mit der Index-Erkrankung (der Erkrankung für die der erste Aufenthalt erfolgte) steht. Dies wurde als spezifische Rehospitalisierung definiert. Verschreibungssituationen, die auf eine Unterversorgung hindeuten, lagen mit höherer Prävalenz vor als Situationen, die auf eine Über- und Fehlversorgung hindeuten. Allerdings wurden weitaus mehr Kriterien der Über- und Fehlversorgung für die finalen Vorhersagemodelle algorithmisch ausgewählt. Die Vorhersagegüte aus regularisierten Regressionsmodellen in einem Testdatensatz war für DM am höchsten (c-Statistik: 0,68 [95% Konfidenzintervall (CI): 0,66-0,70]), gefolgt von COPD (c-Statistik: 0,65 [95% CI: 0,64-0,67]), S/AF (c-Statistik: 0,65 [95% CI: 0,63-0,66]), HF (c-Statistik: 0,61 [95% CI: 0,60-0,62]), AMI (c-Statistik: 0,58 [95% CI: 0,56-0,60]) und OS (c-Statistik: 0,51 [95% CI: 0,47-0,56]). Die Integration der Risiken aus den krankheitsspezifischen Modellen in ein kombiniertes Modell für die Rehospitalisierung jeglicher Ursache ergab eine c-Statistik von 0,63 [95 % KI: 0,63-0,64]. Weiterführende Analysen deckten ein Optimierungspotenzial der Modelle durch Betrachtung der Adhärenz zu verordneten Arzneimitteln oder durch Verwendung alternativer Verfahren aus dem Bereich des maschinellen Lernens auf.

Diskussion: Ein multidisziplinärer Ansatz für die Patientenversorgung während des Index-Falles erscheint vielversprechend, da bereits vorhandene Komorbiditäten zu einem beträchtlichen Teil des Rehospitalisierungsrisikos beitragen. Daneben sagten potenziell inadäquate Verschreibungen Rehospitalisierungen für die meisten Krankheiten erfolgreich voraus und eröffnen damit die Möglichkeit für Interventionen zur Verbesserung dieser veränderbaren Risikofaktoren. Sobald geeignete Rahmenbedingungen für einen Einsatz in der Praxis vorliegen, sollten auch Methoden des maschinellen Lernens stärker zur Entwicklung und Verbesserung von Vorhersagemodellen genutzt werden, insbesondere bei komplexen älteren Patienten mit vielen Grunderkrankungen.

2. Beteiligte Projektpartner

Institution	Ansprechpartner	Verantwortlichkeit/Rolle
Klinische Pharmakologie und Pharmakoepidemiologie, Universitätsklinikum Heidelberg	Prof. Dr. med. Walter E. Haefeli PD Dr. Andreas D. Meid*	Konsortialführung, Projektkoordination Biometrie, Pharmakoepidemiologie
Fakultät Statistik, Technische Universität Dortmund	Prof. Dr. Andreas Groll	Konsortialpartner, statistische Modellierung
AOK Baden-Württemberg	Madelaine Koschei	Konsortialpartner, Bereitstellung der Routedaten

* Fachlicher Ansprechpartner für Rückfragen nach Projektende

3. Projektziele

3.1 Hintergrund, Ausgangslage und Aufgabenstellung des Projektes

Hospitalisierungen stellen für Patienten und das Gesundheitssystem eine enorme Belastung dar. Beispielsweise gab es im Jahr 2018 in Deutschland insgesamt 19,4 Millionen Hospitalisierungen mit geschätzten Gesamtkosten von 94 Milliarden EUR (DESTATIS Statistisches Bundesamt, 2020a, 2020b). Dabei wird ungefähr jeder dritte hospitalisierte Patient¹ im selben Jahr erneut oder gar mehrmals hospitalisiert (Swart, 2005). Aufgrund der Tatsache, dass Therapieversagen und therapiebedingte Komplikationen ein häufiger Grund für Rehospitalisierungen sind, gelten viele dieser Ereignisse als vermeidbar (Donze et al., 2016; Freund et al., 2013; Yam et al., 2010). In Bezug auf die Arzneimitteltherapie lassen sich arzneimittelbedingte (Re-)Hospitalisierungen beispielsweise dadurch klassifizieren, dass wirksame Pharmakotherapien vergessen oder vorenthalten werden (etwa Leitlinientherapie bei Herzinsuffizienz; C in Abb. 1) (Meid & Haefeli, 2016), dass es zu Versorgungslücken mit essenziellen Medikamenten kommt (z. B. durch verzögertes Ausstellen oder Einlösen von Rezepten, D) (Cruden et al., 2014; Ho et al., 2008), dass Therapieregime nicht eingehalten werden (mangelnde Adhärenz bei Antikoagulantien, E), Dosierungen zu niedrig (F) oder zu hoch (H) gewählt oder Wechselwirkungen durch neu angesetzte Therapien nicht erkannt werden (G, I) (**Abb. 1**).

¹ Zur besseren Lesbarkeit wird ausschließlich das generische Maskulinum verwendet. Diese Form bezieht sich immer auf Personen jedes Geschlechts.

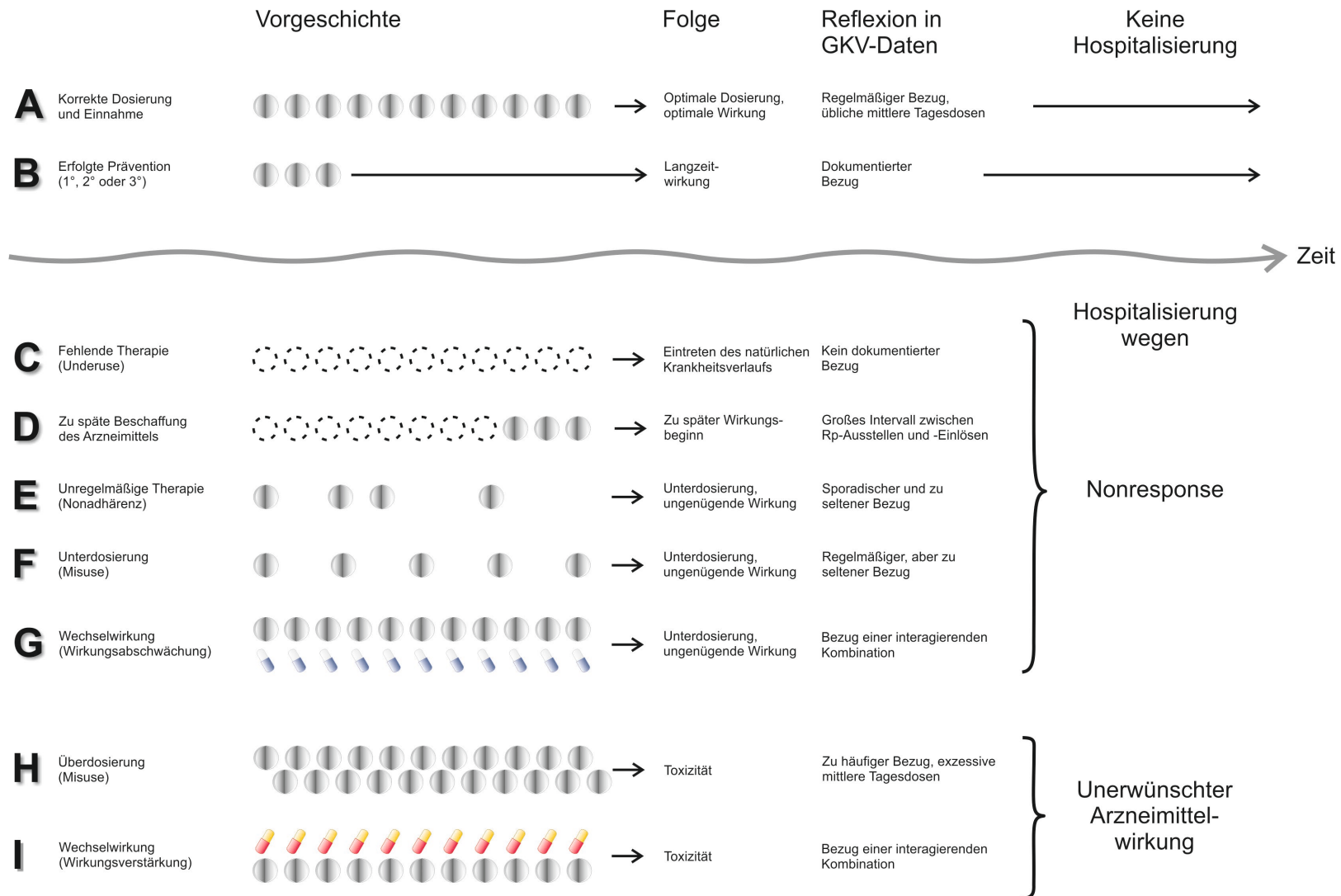


Abb. 1: Mögliche Hospitalisierungsereignisse ausgelöst durch inadäquate Arzneimitteltherapie (nach PREMISE-Innovationsfonds-Antrag). Dargestellt ist der zeitliche Verlauf von Arzneimitteltherapien im Idealfall (A, B) und seine Varianten, die durch fehlende Wirkung (Nonresponse, C-G) oder Toxizität (H, I) zu Hospitalisierungen führen können. Ebenfalls aufgeführt ist eine mögliche Darstellung der verschiedenen Versorgungssituationen in GKV-Daten. **GKV:** Gesetzliche Krankenversicherung; **Rp:** Rezept.

Für Patienten stellen Hospitalisierungen einen gravierenden Einschnitt im Leben dar, ungeachtet der Tatsache ob sie vermeidbar, nicht-vermeidbar oder geplant sind. Eine Hospitalisierung ist gleichzeitig ein starker Risikofaktor für eine zukünftige Rehospitalisierung (Mazzaglia et al., 2007; Smith et al., 2017). Aus klinischer und gesundheitspolitischer Sicht gilt es folglich Rehospitalisierungen zu vermeiden, da hierdurch die Lebensqualität von Risikopatienten aufrechterhalten, die Patientensicherheit erhöht und Kosten im Gesundheitswesen gesenkt werden können (Billings & Mijanovich, 2007; Lewis, 2010). Eine wirksame Früherkennung oder gar interventionelle Vorbeugung von Rehospitalisierungen hätte großes Potenzial die Patientensicherheit zu erhöhen und Gesundheitskosten einzusparen (Meid, Groll, Schieborr, Walker, & Haefeli, 2017). Voraussetzung ist eine möglichst genaue Erkennung des individuellen Rehospitalisierungsrisikos.

Die ältere Patientenpopulation weist im Vergleich zur jüngeren Patientenkohorte neben einem erhöhten Risiko für gesundheitliche Ereignisse außerdem eine höhere Anzahl an (Begleit-)Krankheiten auf. Etwa 62 % der Patienten im Alter von 60-69 Jahren haben zwei chronische Erkrankungen (Puth, Weckbecker, Schmid, & Munster, 2017) und gelten als multimorbide. Dies verdeutlicht der Fakt, dass ein Viertel der Patienten im Alter von 66-70 Jahren an 6-10 Erkrankungen leiden (Kostev & Jacob, 2018). Daneben liegt auch die Arzneimittellast bei Älteren deutlich höher als bei Jüngeren, weshalb auch Wechselwirkungen zwischen Arzneimitteln (*drug-drug interactions*) oder zwischen Arzneimitteln und Grunderkrankungen (*drug-disease interactions*) häufiger beobachtet werden. Dies schlägt sich auch in der Anzahl der stationären Krankenhausfälle in den Altersgruppen nieder. So entfielen im Jahr 2018 in Deutschland 44 % der Krankenhausfälle auf die Gruppe ≥ 65 Jahre (DESTATIS Statistisches Bundesamt, 2020b). Dies stellt auch einen möglichen Grund dar, warum die Pro-Kopf-Ausgaben für medizinische Versorgung in dieser Patientengruppe signifikant höher als in den Altersgruppen < 65 Jahre (Statistisches Bundesamt, 2006) sind. In der Altersgruppe ≥ 65 Jahre würde eine Reduktion der Rehospitalisierungen zu umfangreichen Einsparungen im Gesundheitswesen führen. Hierbei könnte die Identifizierung von zugrundeliegenden Ursachen für Rehospitalisierungen und eine möglichst genaue Erkennung des individuellen (Re-)Hospitalisierungsrisikos ebenfalls dazu beitragen, diese spezielle Patientenpopulation vor belastenden Ereignissen zu schützen.

3.2 Ziele und Erwartungen des Projektes

Erstaunlicherweise berücksichtigen bisherige Definitionen von Rehospitalisierungen den ätiologischen Aspekt der Pharmakotherapie kaum oder nur unpräzise, obwohl vieles dafürspricht, dass dies eine wesentliche Einflussgröße ist (**Abb. 1**). Ziel dieses Projektes war es deshalb, arzneimittelbedingte Rehospitalisierungen zu spezifizieren und so die Vorhersagegüte bisheriger Modelle zu verbessern. Sollten sich Rehospitalisierungen durch eine besser an das individuelle Risiko angepasste Arzneimitteltherapie vermeiden lassen, so könnten entsprechende Maßnahmen auf Grundlage dieser Risikoerkennung die Patientensicherheit maßgeblich erhöhen (siehe auch **Abschnitt 8**).

Die bisher genutzten Prädiktoren und assoziierten Risikofaktoren für Rehospitalisierungen können in soziodemographische, klinische, labordiagnostische und andere Messungen, Prädiktoren zum Arzneimittelgebrauch und sonstige Prädiktoren (etwa die Inanspruchnahme von Gesundheitsleistungen) unterteilt werden. Die Gruppe der Prädiktoren mit Arzneimittelbezug ist bisher jedoch verhältnismäßig klein und ohne große inhaltliche Tiefe. Diese Tatsache in Kombination mit der bisher gängigen Praxis, nur Rehospitalisierungen jeglicher Ursache zu erforschen, birgt Chancen zur Weiterentwicklung von Prädiktionsmodellen zur präziseren Identifikation von Risikopatienten

Viele bisherige Vorhersagemodelle betrachteten ausschließlich Rehospitalisierungen jeglicher Ursache (z. B. Shadmi et al., 2015), ohne dass zwischen möglicherweise geplanten, ungeplanten oder vermeidbaren Rehospitalisierungen unterschieden wurde. Um zwischen

Index-Hospitalisierung und Rehospitalisierung einen inhaltlichen und zeitlichen Bezug herzustellen, sind in Routinedaten in einem ersten Schritt weitreichende Festlegungen erforderlich, welche im Rahmen des Projekts entwickelt und validiert wurden. Erst dadurch wurde eine differenzierte Risikoerkennung möglich, die es erlaubte, Risikopatienten für eine Rehospitalisierung in deutschen Abrechnungsdaten zu identifizieren.

Dazu wurden folgende vier Zielsetzungen verfolgt:

- Etablierung einer Methodik zur Analyse von Rehospitalisierungen in deutschen Abrechnungsdaten und Beschreibung der Epidemiologie von Rehospitalisierungen für sechs volkswirtschaftlich wichtigen Erkrankungen, insbesondere unter Berücksichtigung zeitlicher Verläufe
- Charakterisierung der Erkrankungspopulationen bezüglich Soziodemographie und Komorbiditäten
- Identifikation von Prädiktoren für Rehospitalisierungen im Rahmen der Arzneimitteltherapie
- Vorhersage von Rehospitalisierungen in sechs Risikopopulationen und Prüfung der Generalisierbarkeit

Übergeordnetes Projektziel ist es demnach, Routinedaten gesetzlicher Krankenkassen für die die Erkennung von Risikopatienten für Rehospitalisierungen nutzbar zu machen, indem insbesondere Prädiktoren für eine adäquate Arzneimitteltherapie berücksichtigt werden. Die folgenden Arbeitshypothesen (und damit assoziierte Forschungsfragen) stehen damit im Einklang:

1. Es ist möglich, in Routinedaten ein Regelwerk zur Definition von Rehospitalisierungen zu entwickeln (methodische Grundlagen in **Kapiteln 5.1-5.3**, Ergebnisse in **Kapitel 6.1**)
2. In der resultierenden Datenstruktur können Kandidatenprädiktoren ermittelt werden und Zielpopulationen (mit entsprechenden Erkrankungen oder Index-Hospitalisierungen) charakterisiert werden (methodische Grundlagen in **Kapiteln 5.6.1-5.6.2**, Ergebnisse in **Kapitel 6.2**)
3. Es lassen sich Risikomodelle für Rehospitalisierungen in einzelnen Modellerkrankungen entwickeln und indikationsübergreifend generalisieren (methodische Grundlagen in **Kapitel 5.6.3-5.6.4**, Ergebnisse in **Kapitel 6.3**).

4. Projektdurchführung

Dieses Projekt nutzte Routinedaten (zur ambulanten und stationären Versorgung) von Patienten der gesetzlichen Krankenversicherungen (GKV), um hoch-spezifisch und hoch-sensitiv Risikopatienten für Rehospitalisierungen zu erkennen und perspektivisch daraus Vorbeugemaßnahmen abzuleiten. In ausgewählten Indikationen von besonderer Relevanz (Modellerkrankungen) wurden Risikomodelle unter Berücksichtigung von Therapieinformationen entwickelt und hinsichtlich ihrer Vorhersagegüte (Performance) untersucht. Zu diesem Zweck wurde das Projekt in seiner Durchführung in folgende Aspekte gegliedert (siehe Abb. 2):

- (1) Falldefinition in Modellerkrankungen: Erstellung eines Regelwerks zur zeitlichen und inhaltlichen Definition der Rehospitalisierung in Routinedaten.
- (2) Klinisch-pharmakologische Identifikation von Kandidatenprädiktoren für Rehospitalisierungen.

- (3) Operationalisierung vielversprechender Kandidatenprädiktoren für die Anwendung in Routinedaten der GKV.
- (4) Entwicklung und Validierung von Risikomodelle in Modellerkrankungen.
- (5) Prüfung der Generalisierbarkeit der Risikobewertung durch Aggregation der erkrankungsspezifischen Modelle.
- (6) Ausblick etwa zur Abschätzung des möglichen Einsparpotenzials in verschiedenen Szenarien aufgrund potenziell vermiedener Rehospitalisierungen.

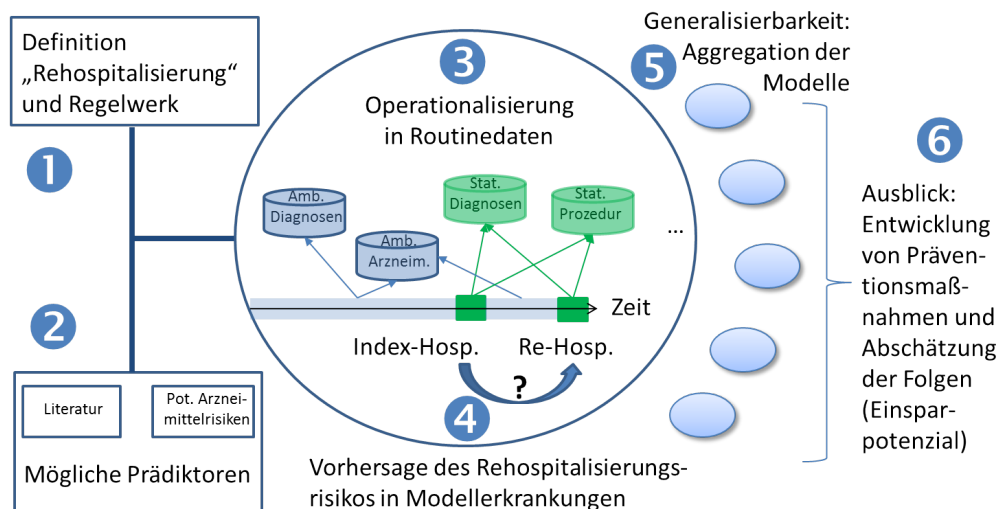


Abb. 2: Schematische Darstellung des Projektablaufs (übernommen aus Antrag). **Amb:** ambulant; **Arzneim:** Arzneimittel; **Hosp:** Hospitalisierung; **Pot:** potenziell; **Stat:** stationär.

In der Durchführung wurde die Projektverlauf um sechs Monate verlängert, um die unter (4) zu entwickelnden Risikomodelle auch für alle Grunderkrankungen hinsichtlich der Vorhersagegüte weitgehend zu optimieren. Dies betrifft im Einzelnen die Vorhersage von Rehospitalisierungen in Zusammenhang mit dem akuten Myokardinfarkt (acute myocardial infarction, AMI) und der Herzinsuffizienz (heart failure, HF). Aus dieser Notwendigkeit sind weiterführende Projektansätze entstanden, deren Ergebnisse in **Kapitel 6.3.1 und 6.3.2** beschrieben sind.

5. Methodik

5.1 Datenbasis

Zur Erstellung der Datenbasis wurde auf Abrechnungsdaten der Jahre 2011–2016 der AOK (Allgemeine Ortskrankenkasse) Baden-Württemberg zurückgegriffen, wobei vollstationäre, über das DRG-System abgerechnete Krankenhausfälle der über 65-Jährigen betrachtet wurden. Als Einschlusskriterien war außerdem gefordert, dass nur Fälle von Versicherten berücksichtigt werden, die im Jahr 2011 mindestens 65 Jahre alt waren und die eine vollständige Versichertenhistorie zwischen 2011 und 2016 aufwiesen. Das Jahr 2011 diente als Vorbeobachtungsperiode, wohingegen die Fälle der Jahre 2012–2016 analysiert wurden.

5.2 Klinische Definition von Index- und Rehospitalisierungsfall

Im Projekt wurden Index-Fälle als Krankenhausfälle definiert, die eindeutig durch eine der sechs Modellerkrankungen (AMI: akuter Myokardinfarkt; COPD: chronisch-obstruktive Lungenerkrankung; DM: Diabetes mellitus Typ 2; HF: Herzinsuffizienz; OS: Osteoporose; S/AF: Schlaganfall, transitorische ischämische Attacke (TIA) oder Vorhofflimmern) ausgelöst wurden. In Bezug auf einen Index-Fall wurde ein Rehospitalisierungsfall als eine zweite,

nachfolgende Krankenhausaufnahme definiert, welche innerhalb eines spezifizierten Zeitraumes (d. h. 30 oder 90 Tage) als Folgefall des Index-Falls auftrat. Trat der zweite Fall innerhalb des vorgegebenen Zeitraumes auf, so wurde dieser als Rehospitalisierungsfall jeglicher Ursache definiert (**Abb. 3**). Stand der zweite Fall zusätzlich klinisch mit der Index-Hospitalisierung bzw. Erkrankung in Beziehung, so wurde dieser Fall auch als spezifischer Rehospitalisierungsfall angesehen. Eine direkte Beziehung zwischen Index-Fall, bzw. -Erkrankung und Rehospitalisierung ist z. B. durch verwandte Symptome, eine typische Komplikation, Exazerbation oder Folgeerkrankung der Index-Erkrankung oder ihrer Behandlung (unerwünschte Arzneimittelwirkung, UAW) gegeben (**Tabelle 1**).

Um als Index-Fall klassifiziert zu werden, musste ein Krankenhausfall innerhalb eines einzigen Krankenhauses stattgefunden haben, keinerlei Zusammenlegungskennzeichen aufweisen, kein Verlegungsfall oder verlegter Fall sein, sowie vollständige Informationen über den Gesundheitszustand des Patienten innerhalb dieses Zeitraums (d. h. Diagnosen und Prozeduren) beinhalten. Aufgrund dieser Festlegung konnte jeder klassifizierte Index-Fall gleichzeitig einen möglichen Rehospitalisierungsfall darstellen. Die sich daraus ergebenden möglichen Fallkonstellationen, wie sie in den Abrechnungsdaten der Krankenkasse erscheinen (siehe hierfür auch **Anhang 1** zur Fallzusammenführung und Rückverlegung im deutschen DRG-System), sowie deren Klassifizierung in dieser Arbeit werden in **Abb. 4** zusammengefasst.

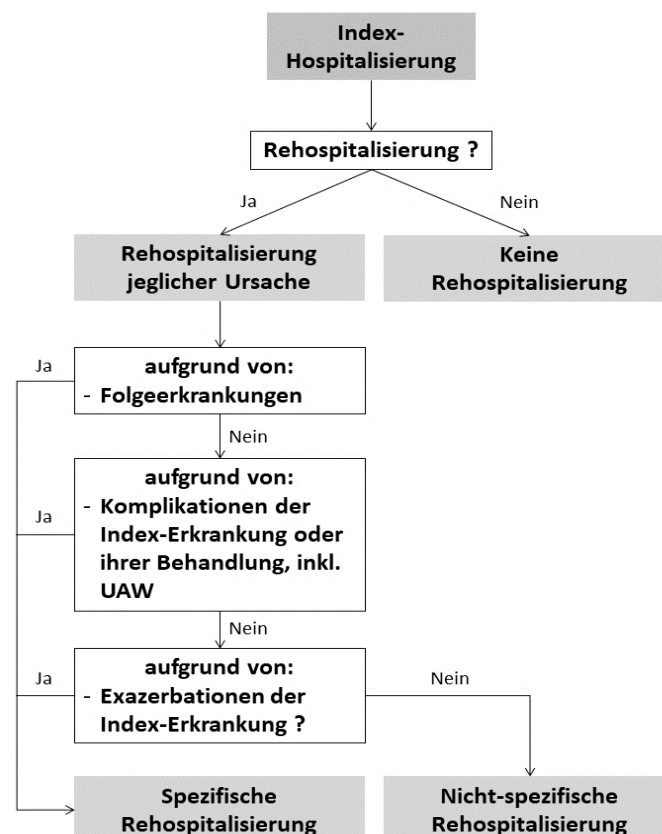


Abb. 3: Klinische Definition eines (nicht-)spezifischen Rehospitalisierungsfall. Eine Rehospitalisierung jeglicher Ursache wurde dann analysiert, wenn sie innerhalb von 30 bzw. 90 Tagen auftrat. Nicht-spezifische Rehospitalisierungen sind die Differenz aus Rehospitalisierungen jeglicher Ursache und spezifischer Rehospitalisierung. Die Bedingungen, unter denen aus einer Rehospitalisierung jeglicher Ursache eine spezifische Rehospitalisierung werden kann (Folgeerkrankungen, Komplikationen aufgrund der Index-Erkrankung oder ihrer Behandlung, inkl. UAW, Exazerbationen der Index-Erkrankung), sind durch ICD-Codes in den Rehospitalisierungs-Code-Sets bestimmt (siehe **S1 Table** aus Ruff et al. 2021). **ICD:** Internationale statistische Klassifikation der Krankheiten; **inkl.:** inklusive; **UAW:** unerwünschte Arzneimittelwirkungen.

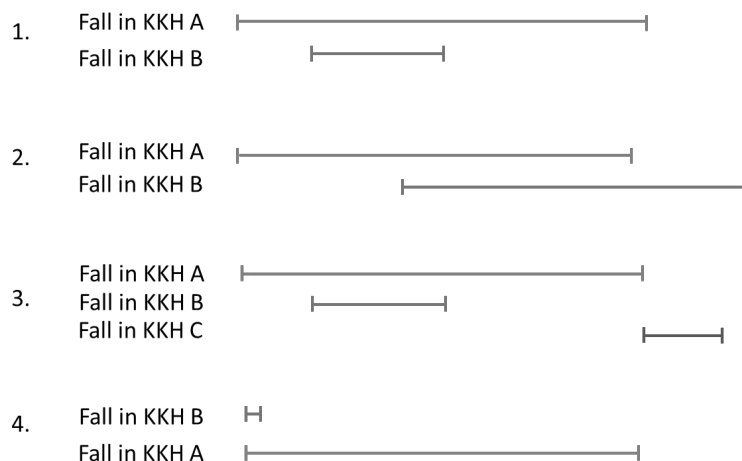


Abb. 4: Spezielle Fallkonstellationen in Krankenkassendaten und deren Klassifizierung für das Projekt. Fälle in Krankenhaus A wurden als mögliche Rehospitalisierungsfälle in die Analyse eingeschlossen. Fälle in Krankenhaus B wurden von der Analyse ausgeschlossen. Fälle in Krankenhaus C (Konstellation 3.) konnten als Index-Fall dienen, wenn der Patient nicht von Krankenhaus A in Krankenhaus C verlegt wurde und der Patient erst am folgenden Tag nach der Entlassung aus Krankenhaus A in Krankenhaus C aufgenommen wurde. Konnte der Fall in Krankenhaus C als Index-Fall dienen, wurde er laut festgelegter Definition ebenfalls als möglicher Rehospitalisierungsfall angesehen. Fälle in Krankenhaus C wurden von der Analyse ausgeschlossen, wenn der Patient von Krankenhaus A in Krankenhaus C verlegt wurden. **KKH:** Krankenhaus.

Tabelle 1. Regelwerk für die Definition von Rehospitalisierungen in sechs Modellerkrankungen.

Erkrankung	Codeset (ICD-10-GM) für Index-Hospitalisierung	Codeset (ICD-10-GM) für nachfolgende Rehospitalisierung
COPD	J44.0-, J44.1-, J44.8-, J44.9-	J44.0-, J44.1-, J44.8-, J44.9-, J41.0, J41.1, J41.8, J42, J43.1, J43.2, J43.8, J43.9, J96.0-, J96.1-, J96.9-, B37.0, C34.0, C34.1, C34.2, C34.3, C34.8, C34.9, I27.9, I50.01, I50.02, I50.03, I50.04, I50.05, R00.0, R06.0, R60.0, R60.1, R60.9, R68.8
OS	M80.0-, M80.1-, M80.2-, M80.3-, M80.4-, M80.5-, M80.8-, M80.9-, M81.0-, M81.1-, M81.2-, M81.3-, M81.4-, M81.5-, M81.6-, M81.8-, M81.9-	M80.0-, M80.1-, M80.2-, M80.3-, M80.4-, M80.5-, M80.8-, M80.9-, M81.0-, M81.1-, M81.2-, M81.3-, M81.4-, M81.5-, M81.6-, M81.8-, M81.9-, S22.0-, S22.1, S32.0-, S32.1, S32.2, S32.3, S32.4, S32.5, S32.7, S32.8-, S42.2-, S42.3, S42.4-, S52.0-, S52.1-, S52.2-, S52.3-, S52.4, S52.5-, S52.6, S52.7, S52.8, S52.9, S72.0-, S72.1-, Z47.0, Z47.8, Z47.9, M84.0-, M84.1-, M84.2-, M87.18, E83.58, E83.59
DM	E11.0-, E11.1-, E11.2-, E11.3-, E11.4-, E11.5-, E11.6-, E11.7-, E11.8-, E11.9-	E11.0-, E11.1-, E11.2-, E11.3-, E11.4-, E11.5-, E11.6-, E11.7-, E11.8-, E11.9-, E16.0, E16.1, E16.2, T38.3, E87.2, B37.3, B37.4, T88.6
HF	I50.0-, I50.1-, I50.9, I11.0-, I13.0-, I13.2-, I25.5, I27.9, I42.0	I50.0-, I50.1-, I50.9, I11.0-, I13.0-, I13.2-, I25.5, I27.9, I42.0, I34.0, I36.1, R57.0, R60.0, R60.1, R60.9, I46.0, I46.1, I46.9, J90, J94.8, R06.0, I47.1, I47.2, I47.9, R00.0, I49.0, I49.3, I44.0, I44.1, I44.2, R00.1, H53.5, Z45.01, Z45.02, Z45.08, R16.0, K76.1, E87.7, E87.1, E87.5, I95.0, I95.1, I95.2, I95.8, I95.9, I20.0, I20.8, I20.9, I21.0, I21.1, I21.2, I21.3, I21.4, I21.9, I22.0, I22.1, I22.8, I22.9, T46.0, T46.4, T44.7, T50.0, T50.1, T50.2
AMI	I21.0, I21.1, I21.2, I21.3, I21.4, I21.9, I22.0, I22.1, I22.8, I22.9, I23.0, I23.1, I23.2, I23.3, I23.4, I23.5, I23.6, I23.8, I20.0, I20.1, I20.8, I20.9, I50.0-, I50.1-, I50.9, I25.0, I25.1-, I25.3, I25.5, I25.8, I25.9, I24.1, I24.8, I24.9, I49.0, R00.1, I95.-, I46.0, I46.1, R57.0, G72.0, E87.5, R04.0, R31, K06.8, K92.0, K92.1, K92.2, K29.0, K25.0, K25.1, K25.2, K25.3, K25.4, K25.5, K25.6, K25.7, K25.9, R23.2, T46.1, T46.3, T46.4, T44.7, T50.0	

S/AF	I63.0, I63.1, I63.2, I63.3, I63.4, I63.5, I63.6, I63.8, I63.9, I64, G45.0-, G45.1-, G45.2-, G45.3-, G45.4-, G45.8-, G45.9-, I69.3, I69.4, G81.0, G81.1, G81.9, G82.0-, G82.1-, G82.2-, G82.3-, G82.4-, G82.5-, G83.1, G83.2, G83.3, G83.6, G40.1, G40.2, H53.2, H53.3, H53.4, R27.0, R29.5, R47.0, R47.1, R48.0, R48.1, R48.2, R48.8, R13.0, R13.1, R13.9, J69.0, F01.0, F01.1, F01.2, F01.3, F01.8, F01.9, I65.0, I65.1, I65.2, I65.3, I65.8, I65.9, I66.0, I66.1, I66.2, I66.3, I66.4, I66.8, I66.9, R04.0, R31, K06.8, K92.0, K92.1, K92.2, K29.0, K25.0, K25.1, K25.2, K25.3, K25.4, K25.5, K25.6, K25.7, K25.9, S06.4, I31.2, R04.1, R04.2, R04.8, R04.9, I85.0, K26.0, K26.2, K26.4, K26.6, K27.0, K27.2, K27.4, K27.6, K28.0, K28.2, K28.4, K28.6, K29.0, K62.5, N93.0, N93.8, N93.9, I60.0, I60.1, I60.2, I60.3, I60.4, I60.5, I60.6, I60.7, I60.8, I60.9, I61.0, I61.1, I61.2, I61.3, I61.4, I61.5, I61.6, I61.8, I61.9, I62.0-, I62.1, I62.9, K22.81, D68.33, D68.34, D68.35, R58, N92.0, N92.1, N92.4, G72.0, E87.1, E87.5, I95.-, T46.4, T50.0, T50.1, T50.2
------	--

AMI: akuter Myokardinfarkt; **COPD:** chronisch-obstruktive Lungenerkrankung; **DM:** Diabetes mellitus Typ 2; **HF:** Herzinsuffizienz; **OS:** Osteoporose; **S/AF:** Schlaganfall, transitorische ischämische Attacke (**TIA**) oder Vorhofflimmern; **ICD-10-GM:** International Classification of Diseases, 10th revision, German Modification.

5.3 Definition des Outcomes (Re-)Hospitalisierung

Für jede der Modellerkrankungen wurde ein Index-Code-Set und ein Rehospitalisierungs-Code-Set von ICD-10-GM-Codes (International Classification of Diseases, 10th revision, German Modification) definiert, mit denen die finalen Index- und Rehospitalisierungsfälle ermittelt wurden, die die technischen und die klinischen Voraussetzungen erfüllten (siehe [S1 Table](#) aus Ruff et al. 2021). Für das Index-Code-Set wurden ICD-10-GM-Codes auf Basis von bereits publizierten Codes ausgewählt. Das Rehospitalisierungs-Code-Set beinhaltet Diagnosen, die eine spezifische Rehospitalisierung widerspiegeln, d. h. Folgeerkrankungen, Komplikationen der Index-Erkrankung oder ihrer Behandlung oder Exazerbationen der Index-Erkrankung darstellen. Dafür wurden ICD-10-GM-Codes auf Basis von publizierten Codes, Leitlinien (Folgeerkrankungen oder Komplikationen) und Fachinformationen der in der Arzneimitteltherapie der Erkrankung eingesetzten Arzneimittel (UAW) ausgewählt unter der Maßgabe, dass diese spezifisch für die entsprechende Erkrankung bzw. Therapie waren (Ruff et al., 2021). Zwei Ärzte überprüften die Code-Sets für HF, AMI bzw. COPD und die Code-Sets wurden entsprechend ihren Vorschlägen angepasst.

War die Hauptdiagnose identisch mit einem aufgeführten ICD-Code des entsprechenden Index-Code-Sets, und erfüllte die Hospitalisierung die Voraussetzungen für eine technische Klassifizierung als Index-Fall, wurde eine Hospitalisierung als Index-Hospitalisierung gewertet. Stimmte die Hauptdiagnose oder eine der Aufnahmediagnosen mit einem aufgeführten ICD-Code des jeweiligen Rehospitalisierungs-Code-Sets überein, und erfüllte die Hospitalisierung die Voraussetzungen zur technischen Klassifizierung als Rehospitalisierungsfall, wurde eine Hospitalisierung als spezifische Rehospitalisierung gewertet (**Abb. 5**).

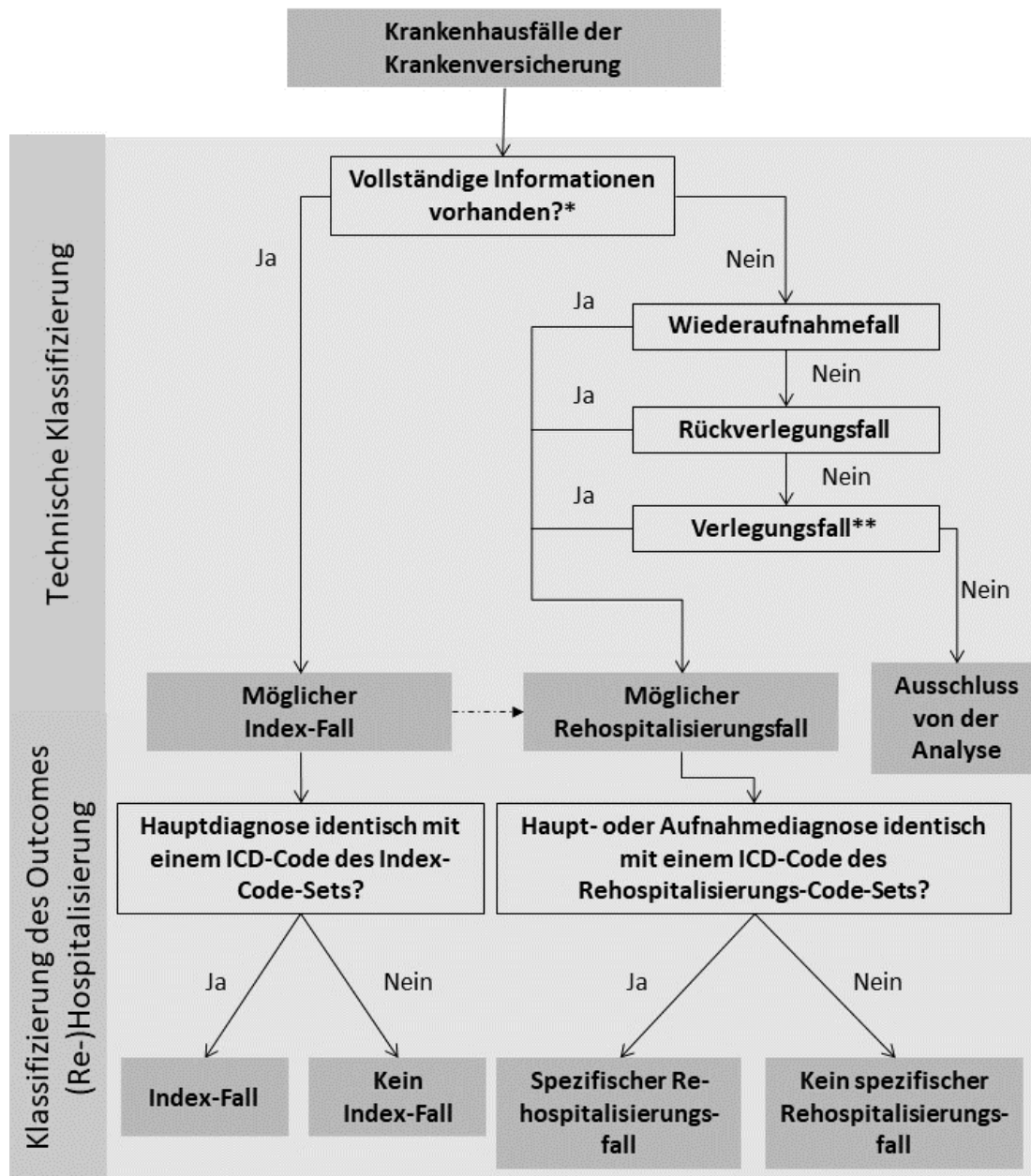


Abb. 5: Vollständige Klassifizierung der Krankenhausfälle in Index- und spezifische Rehospitalisierungsfälle. Im ersten Schritt erfolgte die technische Klassifizierung mit der Einteilung der Fälle in mögliche Index- oder Rehospitalisierungsfälle, welche die Grundlage für die Klassifizierung des Outcomes (Re-)Hospitalisierung darstellten. Nach der technischen Klassifizierung konnte jeder mögliche Index-Fall gleichzeitig einen möglichen Rehospitalisierungsfall darstellen (gestrichelte Verbindungslinie).

*Vollständige Informationen bedeutet: Der Fall musste innerhalb eines einzigen Krankenhauses stattgefunden haben; er durfte keinerlei Zusammenlegungskennzeichen aufweisen, d. h. er durfte kein Rückverlegungs- oder Wiederaufnahmefall sein; er durfte kein Verlegungsfall (= Fall des verlegenden Krankenhauses) oder verlegter Fall (= Fall des aufnehmenden Krankenhauses) sein; er musste alle Informationen über den Gesundheitszustand des Patienten innerhalb des stationären Zeitraums (d. h. Diagnosen und Prozeduren) beinhalten.

**Verlegungsfälle (= Fall des verlegenden Krankenhauses) wurden als mögliche Rehospitalisierungsfälle klassifiziert, außer wenn es sich um einen „Stundenfall“ handelte, d. h., wenn der Patient noch innerhalb des gleichen Tages in ein anderes Krankenhaus verlegt wurde. Dann wurde der Fall des verlegenden Krankenhauses von der Analyse ausgeschlossen und der sich anschließende verlegte Fall als Rehospitalisierungsfall gewertet.

ICD: Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme.

5.4 Medikations-basierte Analysen von Rehospitalisierungen bei Herzinsuffizienz und Herzinfarkt

Ausreichende Adhärenz zu Arzneimitteln bestimmt auch bei den Modellerkrankungen AMI oder HF das Risiko für Rehospitalisierungen. In diesen Modellerkrankungen wurden aus einschlägigen Leitlinien die jeweiligen Behandlungsschemata mit Arzneimittel identifiziert, nach deren Gebrauch die Teildatensätze von AMI und HF so gefiltert wurden, dass nur Patienten mit den jeweiligen Verschreibungen betrachtet wurden. Die individuellen Verordnungsdauern der zu diesem Zeitpunkt in den Leitlinien empfohlenen Medikamente (AMI: Betablocker, Angiotensin-Konversionsenzym-Inhibitoren und AT1-Rezeptorblocker (Renin-Angiotensin-System-Inhibitoren, RASI), Statine und Antithrombotika; HF: Betablocker, RASI, Diuretika und Aldosteronantagonisten) wurden, beginnend ein Jahr vor der Indexhospitalisierung und endend mit dem Zensierungsdatum, abgeleitet. Das Zensierungsdatum wurde entweder durch das Datum einer bestimmten Rehospitalisierung, den Tod des Patienten oder das Ende der Nachbeobachtung definiert, je nachdem, was zuerst eintrat. Konzeptionell wurde für jeden Patienten und jede Arzneimittelklasse ermittelt, ob die Arzneimittelklasse während des Indexkrankenhausaufenthalts abgesetzt wurde (keine Verschreibung innerhalb von 90 Tagen nach der Entlassung). Nur Patienten, die während des Beobachtungszeitraums (erste Verschreibung bis zum Zensierungsdatum) mindestens drei Verschreibungen in jeder Arzneimittelklasse hatten oder die Behandlung per Definition abgebrochen hatten, wurden zur Schätzung der Adhärenz mit der CMA6-Methode aus dem R-Paket *AdhereR* einbezogen. Daraus ergaben sich Schätzungen des Anteils der abgedeckten Tage (*Proportion-Of-Days-Covered*) für die entsprechende Arzneimittelklasse während des Beobachtungszeitraums (im Falle eines Behandlungsabbruchs wurde die Adhärenzschätzung auf 0 gesetzt). In Vorhersagemodellen wurde als binäres Ergebnis eine bestimmte Rehospitalisierung innerhalb von 180 Tagen (AMI) bzw. 90 Tagen (HF) definiert. Die leitlinienadhärente Medikation und die in der Literatur akzeptierten Kovariaten waren feste Terme in den logistischen Regressionsmodellen, während zusätzliche Komorbiditäten in einem Rückwärtsselektionsverfahren getestet wurden, bevor die endgültigen Modelle durch eine LASSO-Regression mit Interaktionstermen der Adhärenzschätzungen für jedes Medikament angepasst wurden. Bei der Entwicklung und Validierung des Modells wurde das Split-Sample-Verfahren angewandt, bei dem jeder dritte Patient nach dem Zufallsprinzip der Testgruppe zugewiesen wurde, um zu untersuchen, wie gut die Adhärenz zu einzelnen Arzneimittelklassen und verschiedenen Arzneimittelkombinationen das Risiko für bestimmte Wiederaufnahmen vorhersagte.

5.5 Epidemiologie von Rehospitalisierungen und Charakterisierung der Erkrankungspopulationen

Die Identifikation der häufigsten Diagnosen, der Vergleich mit externer Referenz (Dharmarajan et al., 2013), die Identifikation von Sequenzen von Index-Erkrankung und Gründen für Rehospitalisierung, die Berechnung der Anteile der Rehospitalisierungsgründe an jeder Index-Erkrankung, die Identifikation von Gründen nicht-spezifischer Wiederaufnahmen innerhalb von 90 Tagen sowie die deskriptive statistische Analyse der Rehospitalisierungen sind detailliert in der Publikation von Ruff et al. 2021 beschrieben.

Zur deskriptiven Beschreibung der Versichertenpopulation wurde jeder Patient einmalig bei der ersten Index-Hospitalisierung charakterisiert. Als soziodemographische Variablen wurden das durchschnittliche Alter (in ganzen Jahren), sowie das Geschlecht identifiziert. Zur Ermittlung von Komorbiditäten wurden die Komorbiditätsgruppen aus dem Elixhauser-Comorbidity-Index genutzt (Elixhauser, Steiner, Harris, & Coffey, 1998; Quan et al., 2005).

5.6 Vorhersage von Rehospitalisierungen in sechs Modellerkrankungen und Prüfung der Generalisierbarkeit

5.6.1 Kandidatenprädiktoren und deren Operationalisierung

Die Identifikation von Prädiktoren für Rehospitalisierungen im Rahmen der Arzneimitteltherapie erfolgte in Übereinstimmung mit **Abbildung 2**; demnach wurden die Prädiktoren nach Evidenz aus der Fachliteratur mit nachfolgender Experteneinschätzung ausgewählt. Aus den in der Literatur identifizierten Originalarbeiten wurden von einem Projektmitglied deskriptiv Informationen zu Prädiktoren extrahiert und deren Relevanz sowie Operationalisierbarkeit im Projektrahmen vom gesamten Projektteam diskutiert. Dabei wurde die Expertise im Projektteam auf den Feldern Kardiologie und Altersmedizin (potenziell arzneimittelinduzierte sturz-assoziierte Frakturen) zusätzlich von einer Expertin der Pneumologie ergänzt. So lieferte die indikationsübergreifende, freie Literaturrecherche eine heterogene Erstausswahl mit hoher Trefferanzahl, bei der beispielsweise Alter (26 gekennzeichnete Publikationen) und Komorbiditäten (29 gekennzeichnete Publikationen) den weitaus größten Anteil hatten und weitere relevante Prädiktoren vermeintlich weniger häufig untersucht wurden. Um Prädiktoren mit Arzneimittelbezug besser kategorisieren zu können wurde daher auch auf ein Einzelfallbeispiel zurückgegriffen, in dem eine systematische Literaturrecherche exemplarisch am Beispiel der direkten oralen Antikoagulantien bei Patienten mit Vorhofflimmern durchgeführt wurde (Ruff et al., 2019). Dieses lieferte die Diskussionsgrundlage für die im Folgenden betrachteten Kandidatenprädiktoren. Zusammenfassend aus den Beratungen ist als maßgebliches Kriterium für die Experteneinschätzung hervorzuheben, dass die Arzneimittelexposition kontextbezogen unter Berücksichtigung der jeweiligen Patientensituation zu bewerten sein sollte und dass entsprechende Situationen und Komorbiditäten in Routinedaten abzubilden sein sollten. Für beide Kriterien konnte auf etablierte Konstrukte zurückgegriffen werden. Zunächst wurden das durchschnittliche Alter (als stetige Variable in ganzen Jahren) und das Geschlecht (als kategorielle Variable) wurden als soziodemographische Variablen ermittelt. Als weiteres Charakteristikum wurde die Anzahl der vorausgegangenen Hospitalisierungen (Anzahl als stetige Variable), die innerhalb eines Jahres vor einem Index-Aufenthalt stattfanden, identifiziert, da diese Variable in bereits publizierten Modellen prädiktiv für eine Rehospitalisierung war (Lee et al., 2019; McCoy, Herrin, Lipska, & Shah, 2018; Serra-Picamal et al., 2018). Zur Ermittlung der Komorbiditäten wurden die 31 Elixhauser-Gruppen (Elixhauser, Steiner, Harris, & Coffey, 1998; Quan et al., 2005) verwendet, wie sie im R-Paket *comorbidities* implementiert sind. Im Unterschied zur Vorgehensweise bei der deskriptiven Beschreibung der Population (siehe **Abschnitt 5.4**) wurden die Komorbiditäten für 31 Elixhauser-Gruppen für jeden einzelnen Index-Fall ermittelt und als kategorielle Variable berücksichtigt.

Bezüglich der Arzneimittelexposition erfolgte die klinisch-pharmakologische Identifikation von Kandidatenprädiktoren für Rehospitalisierungen“ ebendieser Experteneinschätzung, wonach Kontextabhängigkeit und Operationalisierbarkeit maßgeblich waren. Die Entscheidung fiel somit auf Kriterien der Demografie (Alter, Geschlecht), Grundrisiko gemessen an früheren Hospitalisierungen, Komorbiditäten und kontextabhängigen Kriterien für potenziell inadäquate Arzneimittelverordnungen (siehe auch **Anlage 2**).

Für die finale Entscheidung abgewogen wurde neben klinischer Erfahrung die Evidenz aus recherchierten Originalarbeiten und umfassende Kriterienlisten, wozu auch die STOPP-/START-Kriterien (START: Screening-Tool-to-Alert-to-Right-Treatment; STOPP: Screening-Tool-of-Older-People's-Prescriptions) zählen. Deren nach Indikation gegliederte Kriterien deckten sich umfänglich mit der klinischen Erwartungshaltung im Rahmen von Rehospitalisierungen, so dass auf diese umfangreiche Liste zurückgegriffen wurde. Als Grundlage zur Operationalisierung der STOPP-/START-Kriterien dienten Vorarbeiten in deutschen Routinedaten (Meid et al., 2018), welche auf den STOPP-/START-Kriterien der Version 2

basieren (O'Mahony et al., 2015). Dementsprechend wurden im Projekt temporäre Diagnosen aus stationären Diagnose-Daten (in Haupt- oder Nebendiagnosen des stationären Index-Falls kodiert) ermittelt. Chronische Diagnosen wurden aus ambulanten Diagnose-Daten (in mindestens zwei der vier Quartale vor dem Beginn der Index-Hospitalisierung als „gesicherte“ Diagnose kodiert) oder aus stationären Diagnose-Daten (in Haupt- oder Nebendiagnosen des stationären Index-Falls kodiert) identifiziert. Die zum jeweiligen STOPP-/ START-Kriterium gehörende Arzneistoff-Variable wurde dann als vorliegend angesehen, wenn eine Verordnung im Intervall beginnend 90 Tagen vor dem Aufnahmedatum der Index-Hospitalisierung und bis zehn Tagen nach Entlassungsdatum der Index-Hospitalisierung eingelöst wurde. Das Intervallende zehn Tage nach Entlassungsdatum wurde im Projekt verwendet, da hierdurch auch im Krankenhaus neu angesetzte und daraufhin ambulant in Apotheken bezogene Arzneimittel identifiziert werden konnten. **Anhang 1** listet den zugehörigen Analysecode, welcher vollumfänglich die Operationalisierung der Variablen beschreibt.

Weiterhin wurden die operationalisierten STOPP-/ START-Kriterien für eine bessere Übersichtlichkeit und Interpretierbarkeit in indikationsspezifische und sonstige STOPP-/ START-Kriterien eingeteilt, wobei ein STOPP-/ START-Kriterium unter folgenden Bedingungen als indikationsspezifisch gewertet wurde:

- bei Nennung einer Index-Erkrankung als Diagnose in einem STOPP-/ START-Kriterium oder
- bei Nennung einer Exazerbation einer Index-Erkrankung als erwartetes Outcome in den STOPP-/ START-Kriterien

Dies führte zu einer auf Experteneinschätzung basierenden Auswahl an STOPP-Kriterien, welche als indikationsspezifische Kriterien klassifiziert wurden (**Tabelle 2**). In einem analogen Vorgehen wurde dies ebenfalls bei den START-Kriterien durchgeführt (**Tabelle 3**).

5.6.2 Beschreibung der Strata von rehospitalisierten und nicht-rehospitalisierten Patienten

Um die Strata von rehospitalisierten und nicht-rehospitalisierten Patienten für jede Erkrankung zu beschreiben, wurde die Ausprägung der Kandidatenprädiktoren für jeden Index-Fall ermittelt; trug der Patient mehr als einen Index-Fall zur Datengrundlage bei, so wurde er für jeden Index-Fall neu anhand der Variablen charakterisiert, sodass eine vollständige Charakterisierung aller Index-Fälle stattfand. Die Anzahl vorausgegangener Hospitalisierungen innerhalb eines Jahres ging ebenfalls als eigenständiger Kandidatenprädiktor ein. Die Fälle wurden anhand ihres Rehospitalisierungsstatus eingeteilt, sodass für jede Erkrankung unterschieden werden konnte, ob ein Kriterium für Index-Fälle mit oder ohne spezifische Rehospitalisierung vorlag. Zur statistischen Analyse wurden deskriptive Standardverfahren angewendet, um relative Anteile von Index-Fällen zu bestimmen, auf die eine Prädiktorvariable zutraf.

Tabelle 2: Indikationsspezifische STOPP-Kriterien.

STOPP-Kriterium	Inhalt des STOPP-Kriteriums	Index-Erkrankung	Grund für Einteilung als indikationsspezifisches Kriterium
B13	PDE-5-Hemmer bei HF/Nitrat-Therapie für Angina pectoris	HF	Nennung der Index-Erkrankung als Diagnose
C5	ASS und VKA/Antikoagulanzen bei AF	S/AF	Nennung der Index-Erkrankung als Diagnose
C6	TAH und VKA/Antikoagulanzen bei stabiler koronarer, zerebrovaskulärer oder peripher arterieller Erkrankung	S/AF	Nennung der Index-Erkrankung als Diagnose
G1	Theophyllin als Monotherapie für COPD	COPD	Nennung der Index-Erkrankung als Diagnose
G2	Systemische GC als Erhaltungstherapie bei COPD	COPD	Nennung der Index-Erkrankung als Diagnose
H7	COX-2-selektive NSAR bei kardiovaskulärer Erkrankung	AMI und S/AF	Nennung einer Exazerbation einer Index-Erkrankung als erwartetes Outcome
J1	Sulfonylharnstoffe mit langer Wirkdauer bei DM	DM	Nennung der Index-Erkrankung als Diagnose und Nennung einer Exazerbation einer Index-Erkrankung als erwartetes Outcome
J2	Thiazolidendione bei HF	HF	Nennung der Index-Erkrankung als Diagnose und Nennung einer Exazerbation einer Index-Erkrankung als erwartetes Outcome
J3	Beta-Blocker bei DM mit häufigen hypoglykämischen Episoden	DM	Nennung der Index-Erkrankung als Diagnose

AF: Vorhofflimmern. **AMI:** Akuter Myokardinfarkt. **ASS:** Acetylsalicylsäure. **COPD:** Chronisch-obstruktive Lungenerkrankung. **COX-2:** Cyclooxygenase-2. **DM:** Diabetes mellitus Typ 2. **GC:** Glucocorticoide. **HF:** Herzinsuffizienz. **NSAR:** Nicht-steroidale Antirheumatika. **PDE-5:** Phosphodiesterase-5. **TAH:** Thrombozytenaggregationshemmer. **VKA:** Vitamin K-Antagonisten. **S/AF:** Schlaganfall, transitorische ischämische Attacke und Vorhofflimmern.

Tabelle 3: Indikationsspezifische START-Kriterien.

START-Kriterium	Inhalt des START-Kriteriums	Index-Erkrankung	Grund für Einteilung als indikationsspezifisches Kriterium
A1	Antikoagulation bei chronischem AF	S/AF	Nennung der Index-Erkrankung als Diagnose
A2	ASS bei chronischem AF wegen Kontraindikation	S/AF	Nennung der Index-Erkrankung als Diagnose
A3	TAH bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung	AMI und S/AF	Nennung der Index-Erkrankung als Diagnose
A5	Statine bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung	AMI und S/AF	Nennung der Index-Erkrankung als Diagnose
A6	ACE-Hemmer bei HF/ KHK	HF	Nennung der Index-Erkrankung als Diagnose
A7	Beta-Blocker bei ischämischer Herzkrankheit	AMI	Nennung der Index-Erkrankung als Diagnose
A8	geeigneter Beta-Blocker bei systolischer HF	HF	Nennung der Index-Erkrankung als Diagnose
E4	Antiresorptive oder knochen-anabole Therapie bei OS	OS	Nennung der Index-Erkrankung als Diagnose
F1	ACE-Hemmer oder ARB bei DM mit renaler Erkrankung	DM	Nennung der Index-Erkrankung als Diagnose

ACE: Angiotensin-Converting-Enzym. **AF:** Vorhofflimmern. **AMI:** Akuter Myokardinfarkt. **ARB:** Angiotensin-II-Rezeptorblocker. **ASS:** Acetylsalicylsäure. **COPD:** Chronisch-obstruktive Lungenerkrankung. **COX-2:** Cyclooxygenase-2. **DM:** Diabetes mellitus Typ 2. **GC:** Glucocorticoide. **HF:** Herzinsuffizienz. **KHK:** Koronare Herzkrankheit. **NSAR:** Nicht-steroidale Antirheumatika. **OS:** Osteoporose. **PDE-5:** Phosphodiesterase-5. **TAH:** Thrombozytenaggregationshemmer. **VKA:** Vitamin K-Antagonisten. **S/AF:** Schlaganfall, transitorische ischämische Attacke und Vorhofflimmern.

5.6.3 Entwicklung der Prädiktionsmodelle

Die Modellentwicklungen erfolgten in einer ersten Stufe mit parametrischen Verfahren, wobei generalisierte lineare (Regressions-) Modelle (GLM bzw. logistische Regression) verwendet wurden. Für die Hauptanalyse wurde der Einfluss von Prädiktorvariablen auf das Risiko einer Rehospitalisierung innerhalb von 90 Tagen (binäre Zielvariable) geschätzt um als lineare Prädiktoren die Rehospitalisierungswahrscheinlichkeiten in einzelnen Fällen zu berechnen. Für die Auswahl an Prädiktoren (Variablenselektion) aus allen Kandidaten wurden zwei alternative Ansätze verfolgt: Die Modellierung eines Regressionsmodells ohne Variablenselektion (für ein so genanntes „volles Modell“ mit allen Prädiktoren) und als Hauptanalyse ein regularisiertes Regressionsmodell, bei welchem durch Anwendung des so genannten LASSO-Verfahrens (*Least-Absolute-Shrinkage-and-Selection-Operator*) nur eine Selektion aus allen möglichen Kandidatenprädiktoren aufgenommen wurde. Der Einfluss der nicht-aufgenommenen Variablen wird dabei auf null korrigiert, was dem Entfernen der entsprechenden Variable entspricht und zu Modellen mit höherer (externer) Vorhersagekraft und gleichzeitig kleineren und daher besser interpretierbaren Modellen führt. Der dafür notwendige so genannte Tuning-Parameter λ wurde in fünffacher Kreuzvalidierung bestimmt. Die Modellgüte (Performance) der einzelnen Modelle wurde in einer zehnfachen Kreuzvalidierung ermittelt und über die c-Statistik definiert, welche bei logistischen Regressionsmodellen der Fläche unter der ROC-Kurve (*Area-Under-the-Curve* der *Receiver-Operator-Characteristics-Kurve*) entspricht. Damit wird die Fähigkeit zur Diskrimination des Modells dahingehend beschrieben, inwieweit das Modell über die vorhergesagten Risiken die Patienten mit und ohne Rehospitalisierung zu trennen vermag. Zur Ermittlung der ROC-Kurve für verschiedene Schwellenwerte (gemäß Youden-Index (Youden, 1950)) der vorhergesagten Risiken wurde separat für jede Erkrankung ein unabhängiger Testdatensatz erstellt, welcher durch zufälliges Auftrennen aller Index-Fälle in einen Entwicklungsdatensatz (enthielt 70 % der Index-Fälle) und einen Testdatensatz (enthielt die restlichen 30 % der Index-Fälle) erhalten wurde. Zur Prüfung der Generalisierbarkeit der Prädiktionsmodelle wurden alle einzelnen, in den verschiedenen Modellerkrankungen entwickelten Regressionsmodelle mittels eines *Stacked-Regression*-Ansatzes aggregiert (Debray et al., 2014; Martin, Mamas, Peek, Buchan, & Sperrin, 2017). Damit kann gemessen werden, welchen Einfluss die krankheitsspezifischen Modelle auf eine Rehospitalisierung jeglicher Ursache (*All-Cause-Readmission*) haben. Die Basis bildeten somit die aus den krankheitsspezifischen LASSO-Modellen ermittelten Rehospitalisierungsrisiken (lineare Prädiktoren), welche dann ihrerseits als Kandidatenprädiktoren im Modell für Rehospitalisierungen jeglicher Ursache verwendet wurden. Auf diese aus den Einzelmodellen abgeleiteten Variablen wurde eine unregularisierte logistische Regression gefittet. Letztlich wurde ein *stacked* Regressionsmodell erstellt, also ein GLM, welches die unabhängigen Variablen nutzte, die zuvor per LASSO bestimmt wurden.

5.6.4 Modellvergleiche von Verfahren aus dem Bereich künstliche Intelligenz und Machine-Learning

Als Benchmark-Modell wurde in dieser weiterführenden Analyse ein einfaches generalisiertes lineares Modell (GLM) gewählt, welches nur die grundlegenden Informationen über die Patienten enthält, ohne dass die verfügbaren Rohdaten vorverarbeitet werden müssen. Dieses Basismodell enthält grundlegende demografische Daten und Informationen über die Entlassung der Patienten. Für den gesamten Datensatz wurde nun ein einfaches GLM, klassische Methoden der Regularisierung (LASSO; RIDGE: Ridge Regression; relaxed-LASSO), klassische Methoden des maschinellen Lernens (kNN: k-nearest-neighbors; CART: Classification and Regression Tree; RF: Random Forest) und klassische Boosting-Methoden (GLMBOOST: Boosting Variante für GLMs; XGBOOST: eXtreme Gradient Boosting) verglichen. Außerdem wurde die so genannte Variable-Importance der Random Forests betrachtet, um unsere Variablen nach Wichtigkeit zu ordnen. Diese Metrik wurde dazu genutzt, eine Teilmenge von Merkmalen auszuwählen und ein grundlegendes GLM oder einen

grundlegenden CART mit dieser Teilmenge von Merkmalen zu berechnen (ähnlich der Idee von relaxed-LASSO). Für diesen Prozess wurden sowohl die GINI-Impurity-Decrease-Variable-Importance als auch die Variable-Importance aus einem Permutationsverfahren verwendet, was zu vier Paaren führt (GINI GLM: GLM nach Variablen Selection mit GINI-Impurity; GINI CART: CART nach Variablen Selection mit GINI-Impurity; PERM GLM: GLM nach Variablen Selection mit Permutationsverfahren; PERM CART: CART nach Variablen Selection mit Permutations-Verfahren).

5.7 Verwendete Software

Die Vorbereitung der Daten und der Aufbau der Datenbasis wurde mit Microsoft Structured Query Language (MS SQL) Server 2017 (Microsoft Corporation, Redmond, WA, USA) durchgeführt. Alle statistischen Analysen zur Beschreibung der Epidemiologie von Rehospitalisierungen wurden mittels der Software R in Version 3.6.0 (R Foundation for Statistical Computing, Vienna, Austria) durchgeführt. Alle statistischen Analysen zur Charakterisierung der Population wurden mittels der Software R in Version 4.0.3 (R Foundation for Statistical Computing, Vienna, Austria) durchgeführt. Alle statistischen Analysen zur Entwicklung der Prädiktionsmodelle wurden mittels der Software R in Version 4.0.3 (R Foundation for Statistical Computing, Vienna, Austria) mit den Schlüsselpaketen glmnet (Version 4.0-2) und pROC (Version 1.16.2) durchgeführt.

6. Projektergebnisse

Alle wesentlichen Projektergebnisse sind in erfolgten Veröffentlichungen (siehe **Abschnitt 9**) detailliert beschrieben, so dass insbesondere die zentralen Erkenntnisse in diesem Abschnitt wiedergegeben werden. Im Einklang mit den Arbeitshypothesen (**Kapitel 3.2**) betrifft dies die Beschreibung der Populationen, die aus dem in Routinedaten entwickelten Regelwerk zur Definition von Rehospitalisierungen hervorgingen (**Kapitel 6.1**), daneben die Charakterisierung dieser Populationen anhand der betrachteten Kandidatenprädiktoren (**Kapitel 6.2**) und schließlich die Beschreibung der damit entwickelten Vorhersagemodelle für Rehospitalisierungen in einzelnen Modellerkrankungen und deren indikationsübergreifende Generalisierung (**Kapitel 6.3**). Als Bewertungskriterien gelten hierbei deskriptive Häufigkeitsmaße (**Kapitel 6.1-6.2**) und statistische Kennzahlen für die Modellperformance. Beispielsweise wird die Verteilung von Unterversorgung, Über- und Fehlversorgung mit Arzneimitteln (gemessen an den STOPP-/START-Kriterien) deskriptiv beschrieben (**Kapitel 6.2**), wonach ihr prädiktiver Effekt quantifiziert wird (**Kapitel 6.3, Anhang 4**). Die Generalisierbarkeit wird dagegen mit der statistischen Kenngröße der c-Statistik bewertet, nachdem die Rehospitalisierungswahrscheinlichkeiten aus den krankheitsspezifischen Modellen in ein übergreifendes, verallgemeinerndes Modell („stacked“) integriert wurden (**Kapitel 6.3**).

6.1 Rehospitalisierungen in deutschen Abrechnungsdaten und Epidemiologie

Im Beobachtungszeitraum von 2012–2016 wurden insgesamt 1 841 877 Versicherte identifiziert, für welche 5 039 570 Krankenhausfälle dokumentiert waren. Nach der Durchführung der Datenaufbereitung verblieben 1 689 019 Fälle, welche von 569 912 Versicherten über 65 Jahren mit durchgängigem Versichertenstatus erzeugt wurden (**Abb. 6**).

Bei der Betrachtung von Hospitalisierungen jeglicher Ursache wurden von den Patienten mit Index-Hospitalisierung rund 14-22 % innerhalb von 30 und rund 27-41 % innerhalb von 90 Tagen wiederaufgenommen (**Tabelle 4**). Bei Betrachtung von spezifischen Rehospitalisierungen (gleicher oder damit zusammenhängender, nahe verwandter Grund) wurden je nach Index-Erkrankung bis zu 13 % der Index-Fälle innerhalb von 30 Tagen und 25 % innerhalb von 90 Tagen rehospitalisiert. Die meisten Index-Fälle gingen auf HF (4,96 %) und

S/AF (4,99 %) zurück und HF und COPD lieferten die meisten Rehospitalisierungen jeglicher Ursache innerhalb von 30 (21,6 %, bzw. 21,0 %) und 90 Tagen (41,0 %, bzw. 41,2 %).

Die häufigsten Diagnosen, die innerhalb von 30 und 90 Tagen zu Rehospitalisierungen führten, sind in [S2 Table von Ruff et al. 2021](#) dargestellt. HF war hierbei insgesamt die häufigste Hauptdiagnose. Die Häufigkeitsverteilung über die Zeit der Rehospitalisierungen jeglicher Ursache in der Baden-Württembergischen AOK-Population war mit der Verteilung der Rehospitalisierungen der US-amerikanischen Medicare-Population vergleichbar ([Fig. 1 aus Ruff et al. 2021](#)). Die Patienten mit Wiederaufnahme binnen 90 Tagen wurden am häufigsten für andere als die in der Analyse betrachteten Gründe („andere Gründe“) rehospitalisiert ([Fig. 2 aus Ruff et al. 2021](#)). Der zweithäufigste Grund für eine Rehospitalisierung war meist die Index-Erkrankung. Insgesamt stellten Rehospitalisierungen für HF den häufigsten Grund für eine Wiederaufnahme dar. Die Anteile der sechs Erkrankungen unterschieden sich je nach Rehospitalisierungsgrund stark voneinander ([Abb. 7a-7c](#)). Bei Betrachtung der genannten Index-Erkrankungen wurden die größten Anteile für eine Rehospitalisierung erwartungsgemäß von der zugrunde liegenden Index-Erkrankung beigetragen. So wurden beispielsweise 9,96 % der DM-Index-Fälle für DM rehospitalisiert, aber nur 0,63 – 1,39 % der Index-Fälle der anderen Index-Erkrankungen. Auffällig war außerdem, dass verwandte Erkrankungen in der Regel die höheren Anteile als nicht-verwandte Erkrankungen lieferten (z. B. HF-Index-Fälle für Rehospitalisierung für AMI, COPD- und AMI-Index-Fälle für Rehospitalisierung für HF).

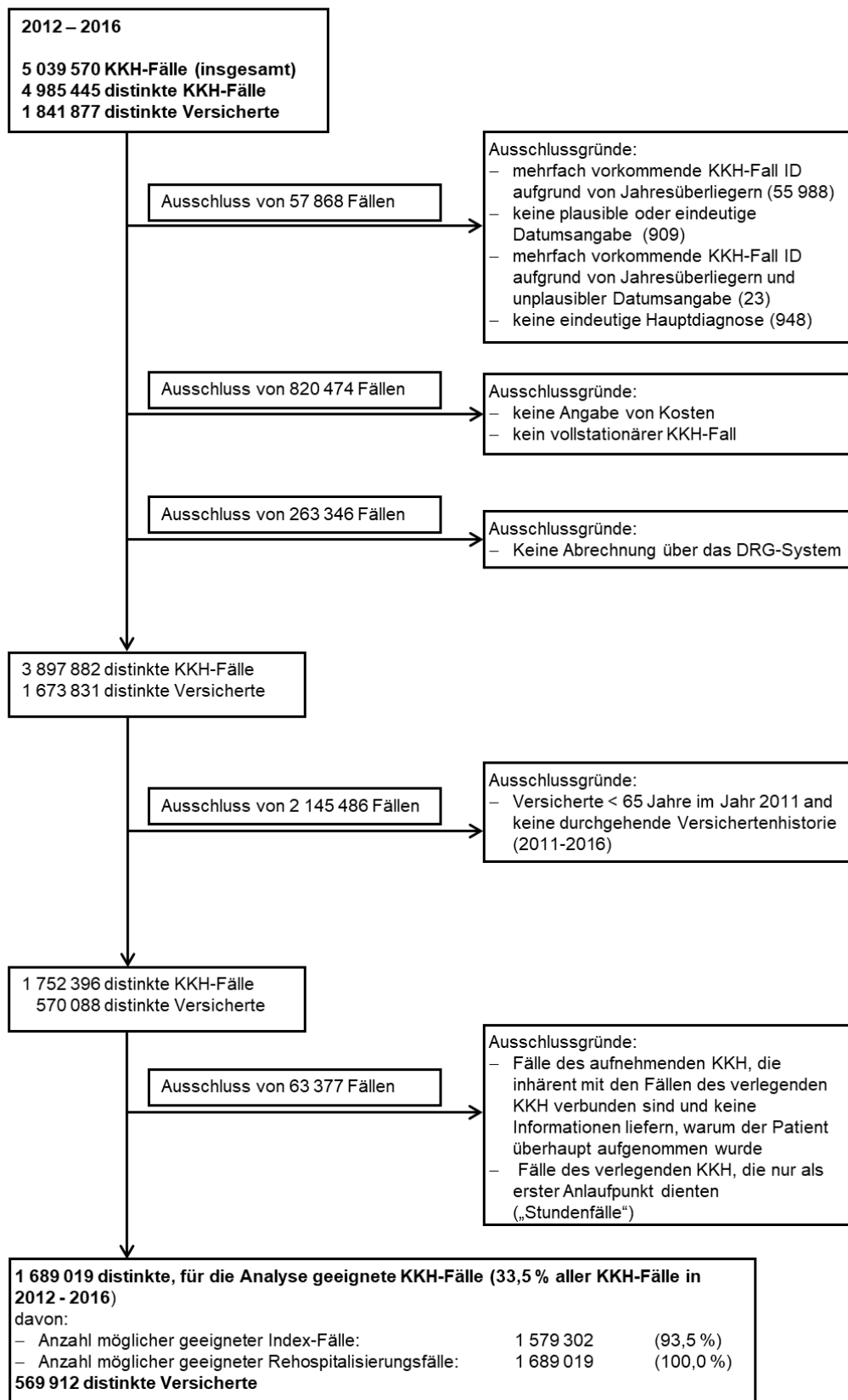


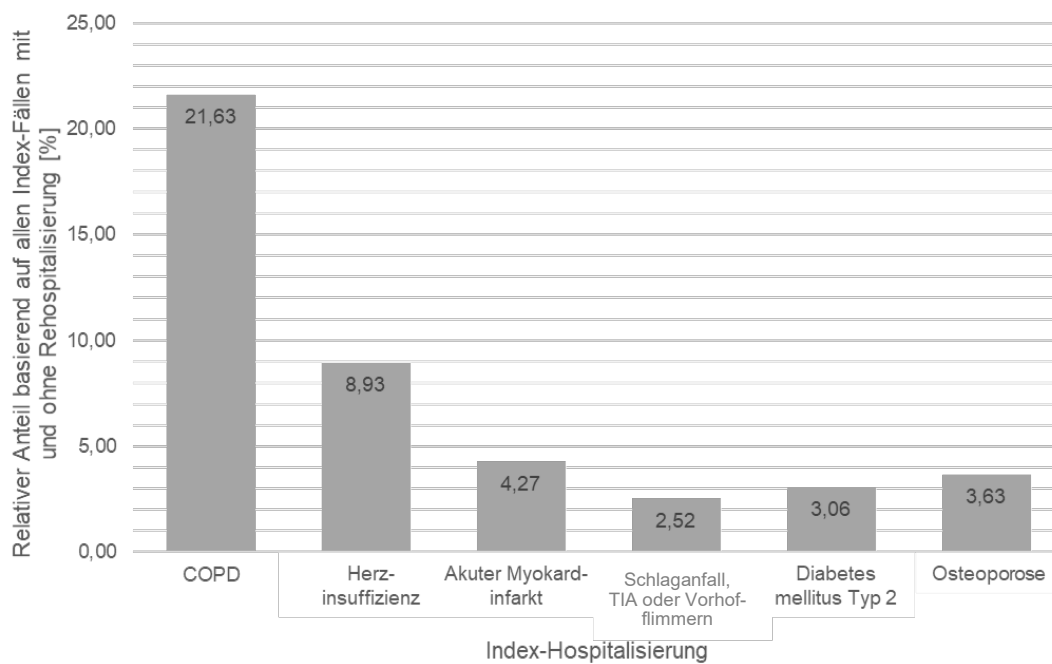
Abb. 6: Flussdiagramm mit Darstellung der Datenbereinigung und Ergebnissen der Fallklassifizierung (adaptiert von Ruff et al. 2021). **DRG:** Diagnosis-Related-Group; **ID:** Identifikationsnummer; **KKH:** Krankenhaus

Tabelle 4: Anzahl und Anteile von (Re-) Hospitalisierungen der Jahre 2012-2016 (adaptiert von Ruff et al. 2021).

Hospitalisierungsart	Anteil der Fälle (%*) mit Wiederaufnahme innerhalb von	
	30 Tagen (Anteil (%) und absolute Anzahl)	90 Tagen (Anteil (%) und absolute Anzahl)
COPD (chronisch-obstruktive Lungenerkrankung)		
Index-Fall	100,0 % (29323)	100,0 % (28446)
Keine Rehospitalisierung	79,0 % (23165)	58,8 % (16715)
Rehospitalisierung jeglicher Ursache	21,0 % (6158)	41,2 % (11731)
Spezifische Rehospitalisierung	12,9 % (3794)	25,4 % (7217)
Nicht-spezifische Rehospitalisierung	8,1 % (2364)	15,9 % (4514)
Osteoporose		
Index-Fall	100,0 % (6315)	100,0 % (6111)
Keine Rehospitalisierung	80,8 % (5100)	64,7 % (3951)
Rehospitalisierung jeglicher Ursache	19,2 % (1215)	35,3 % (2160)
Spezifische Rehospitalisierung	5,6 % (352)	10,9 % (664)
Nicht-spezifische Rehospitalisierung	13,7 % (863)	24,5 % (1496)
Diabetes mellitus Typ 2		
Index-Fall	100,0 % (24338)	100,0 % (23665)
Keine Rehospitalisierung	82,8 % (20164)	66,0 % (15623)
Rehospitalisierung jeglicher Ursache	17,2 % (4174)	34,0 % (8042)
Spezifische Rehospitalisierung	5,3 % (1290)	11,1 % (2621)
Nicht-spezifische Rehospitalisierung	11,8 % (2884)	22,9 % (5421)
Herzinsuffizienz		
Index-Fall	100,0 % (83814)	100,0 % (80870)
Keine Rehospitalisierung	78,3 % (65689)	59,0 % (47678)
Rehospitalisierung jeglicher Ursache	21,6 % (18125)	41,0 % (33192)
Spezifische Rehospitalisierung	10,8 % (9087)	20,7 % (16749)
Nicht-spezifische Rehospitalisierung	10,8 % (9038)	20,3 % (16443)
Akuter Myokardinfarkt		
Index-Fall	100,0 % (19519)	100,0 % (18893)
Keine Rehospitalisierung	80,8 % (15764)	61,5 % (11620)
Rehospitalisierung jeglicher Ursache	19,2 % (3755)	38,5 % (7273)
Spezifische Rehospitalisierung	10,5 % (2042)	21,8 % (4116)
Nicht-spezifische Rehospitalisierung	8,8 % (1713)	16,7 % (3157)
Schlaganfall, transitorische ischämische Attacke (TIA) oder Vorhofflimmern		
Index-Fall	100,0 % (84326)	100,0 % (81632)
Keine Rehospitalisierung	86,5 % (72952)	72,6 % (59294)
Rehospitalisierung jeglicher Ursache	13,5 % (11374)	27,4 % (22338)
Spezifische Rehospitalisierung	2,9 % (2458)	5,4 % (4397)
Nicht-spezifische Rehospitalisierung	10,6 % (8916)	22,0 % (17941)
Gesamt		
Index-Fall	100,0 % (247635)	100,0 % (239617)
Keine Rehospitalisierung	81,9 % (202834)	64,6 % (154881)
Rehospitalisierung jeglicher Ursache	18,1 % (44801)	35,4 % (84736)
Spezifische Rehospitalisierung	7,7 % (19023)	14,9 % (35764)
Nicht-spezifische Rehospitalisierung	10,4 % (25778)	20,4 % (48972)

*Summe kann aufgrund von Rundung von 100 % abweichen.

Rehospitalisierung für COPD



Rehospitalisierung für Herzinsuffizienz

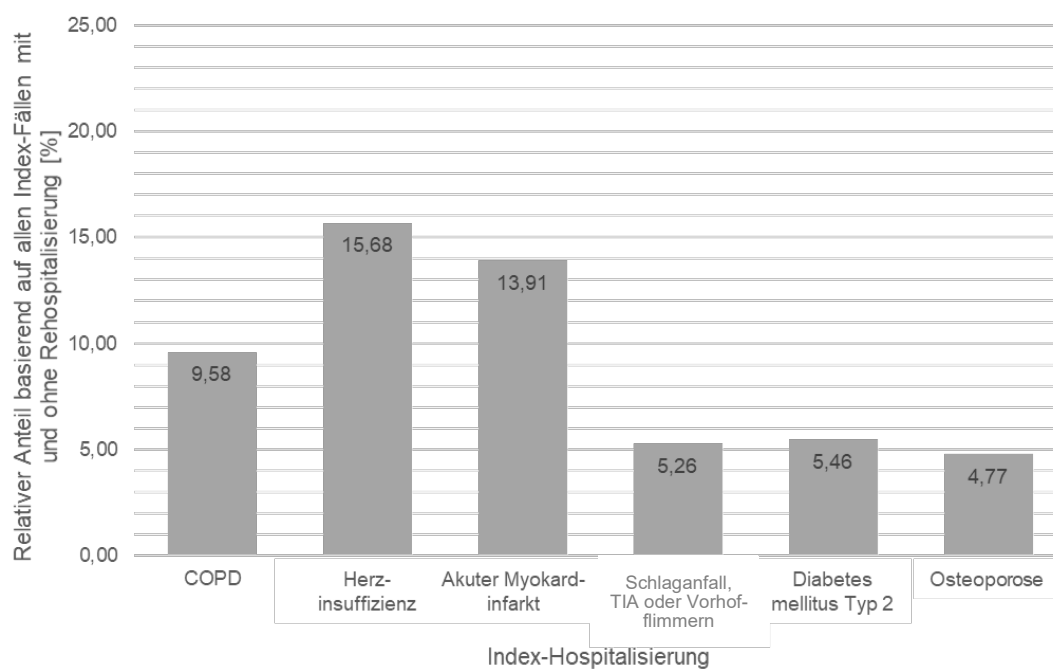
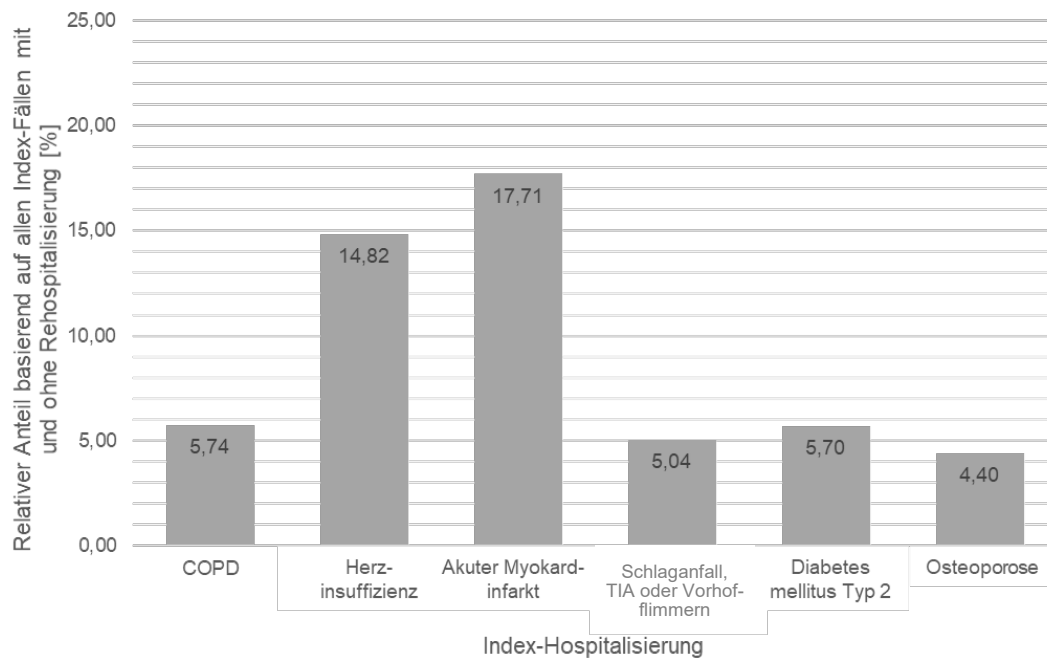


Abb. 7a: Anteile von Index-Hospitalisierungen, welche für COPD bzw. Herzinsuffizienz rehospitalisiert wurden. Angegeben werden die Anteile der Index-Hospitalisierungen (y-Achse) pro Index-Erkrankung (x-Achse), welche für COPD (oberes Panel) bzw. Herzinsuffizienz (unteres Panel) rehospitalisiert wurden. Zum Beispiel wurden 4,77 % der Osteoporose-Index-Fälle und 5,46 % der Diabetes mellitus Typ 2-Index-Fälle für Herzinsuffizienz rehospitalisiert. **COPD:** Chronisch-obstruktive Lungenerkrankung; **TIA:** Transitorische ischämische Attacke.

Rehospitalisierung für Akuten Myokardinfarkt



Rehospitalisierung für Diabetes mellitus Typ 2

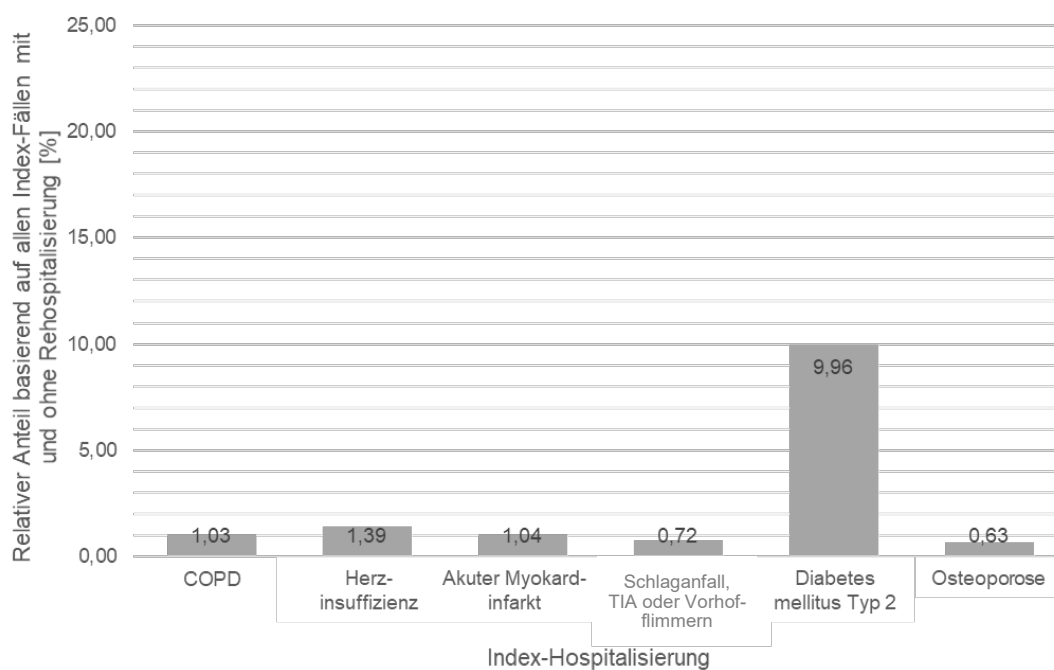
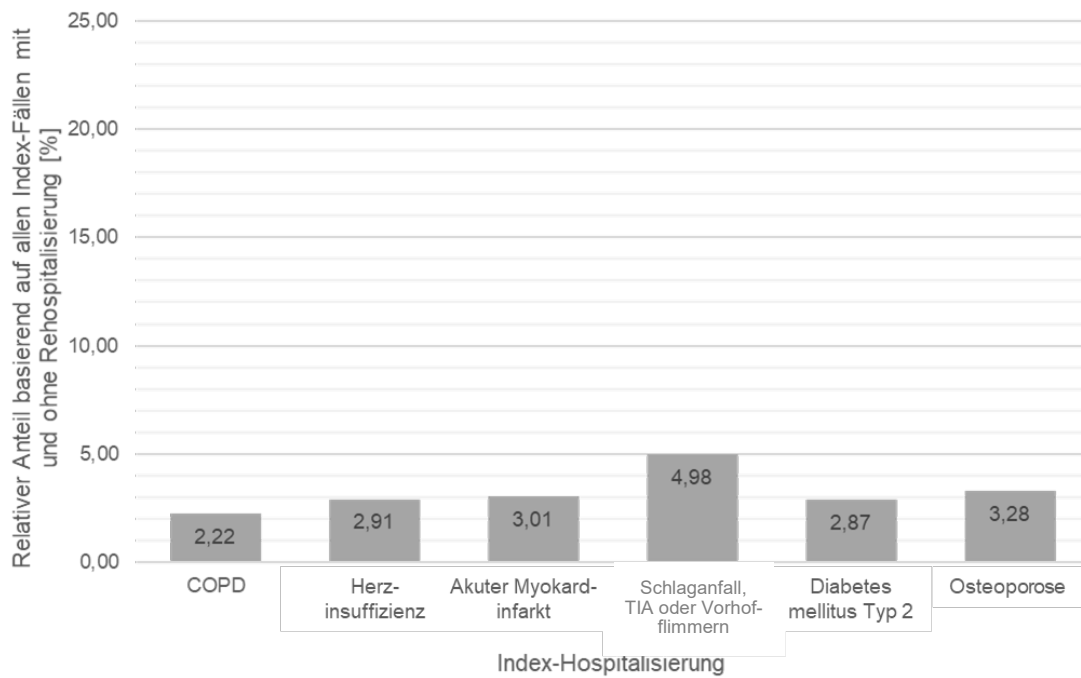


Abb. 7b: Anteile von Index-Hospitalisierungen, welche für akuten Myokardinfarkt bzw. Diabetes mellitus Typ 2 rehospitalisiert wurden. Angegeben werden die Anteile der Index-Hospitalisierungen (y-Achse) pro Index-Erkrankung (x-Achse), welche für akuten Myokardinfarkt (oberes Panel) bzw. Diabetes mellitus Typ 2 (unteres Panel) rehospitalisiert wurden. Zum Beispiel wurden 0,63 % der Osteoporose-Index-Fälle und 9,96 % der Diabetes mellitus Typ 2-Index-Fälle für Diabetes mellitus Typ 2 rehospitalisiert. **COPD:** Chronisch-obstruktive Lungenerkrankung; **TIA:** Transitorische ischämische Attacke.

Rehospitalisierung für Schlaganfall, TIA oder Vorhofflimmern



Rehospitalisierung für Osteoporose

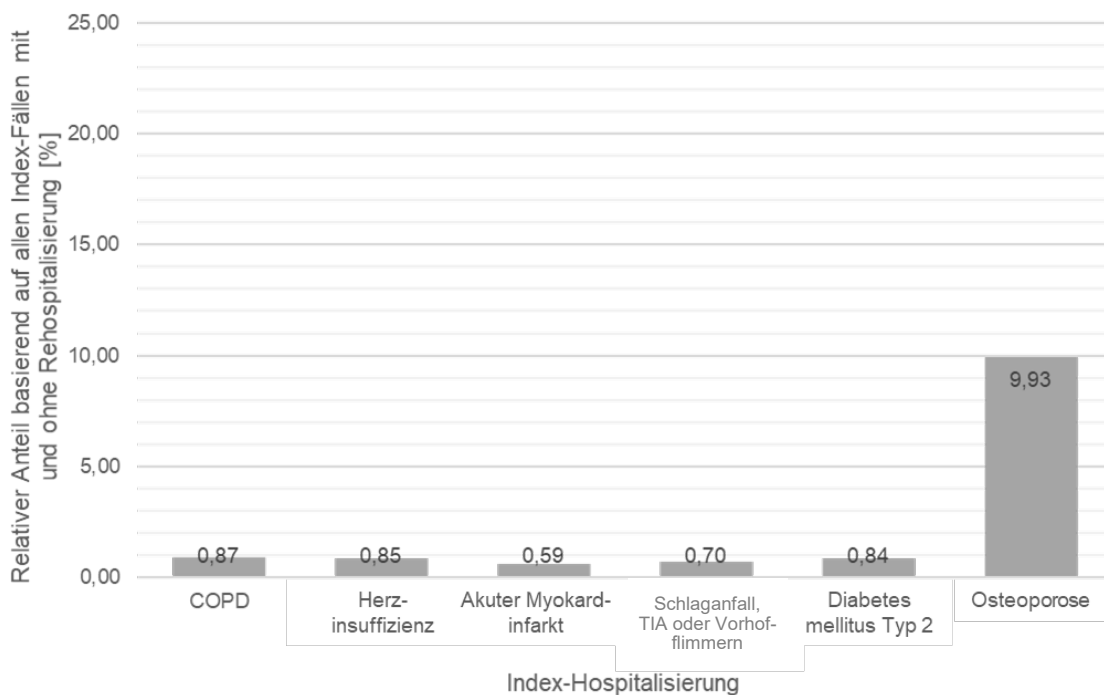


Abb. 7c: Anteile von Index-Hospitalisierungen, welche für Schlaganfall, TIA oder Vorhofflimmern bzw. Osteoporose rehospitalisiert wurden. Angegeben werden die Anteile der Index-Hospitalisierungen (y-Achse) pro Index-Erkrankung (x-Achse), welche für Schlaganfall, TIA oder Vorhofflimmern (oberes Panel) bzw. Osteoporose (unters Panel) rehospitalisiert wurden. Zum Beispiel wurden 9,93 % der Osteoporose-Index-Fälle und 0,84 % der Diabetes mellitus Typ 2-Index-Fälle für Osteoporose rehospitalisiert. **COPD:** Chronisch-obstruktive Lungenerkrankung; **TIA:** Transitorische ischämische Attacke.

Die Verteilungen und Kurvenverläufe der Rehospitalisierungen jeglicher Ursache und der spezifischen Rehospitalisierungen zeigten krankheitsspezifische Muster mit jeweils unterschiedlichen Maxima ([Fig. 3](#) in Ruff et al. 2021). Nach Einteilung der Hauptdiagnosen der nicht-spezifischen Wiederaufnahmen zu ihren zugehörigen ICD-Kapiteln lieferten zwei ICD-Kapitel unabhängig der analysierten Index-Erkrankung jeweils mindestens einmal im analysierten Zeitraum 5 % aller Rehospitalisierungen: Kapitel XI (Krankheiten des Verdauungssystems) und Kapitel IX (Krankheiten des Kreislaufsystems). Kapitel IX spiegelte dabei für jede Erkrankung die meisten Rehospitalisierungen wieder. Dies war auch für HF und AMI der Fall, obwohl für HF und AMI bereits kardiovaskuläre Erkrankungen als spezifische Rehospitalisierung klassifiziert wurden, die in den unspezifischen Wiederaufnahmen nicht mehr vorkamen ([S2 Fig](#) in Ruff et al 2021).

6.2 Charakterisierung der Erkrankungspopulationen bezüglich Soziodemographie und Komorbiditäten

In Anbetracht der Rolle von Komorbiditäten und insbesondere Herz-Kreislauf-Erkrankungen als Gründe für Rehospitalisierungen nach einer Index-Erkrankung wurde die Komorbiditätslast und Soziodemographie zum Zeitpunkt der ersten Index-Hospitalisierung anhand des Elixhauser-Komorbiditäts-Scores analysiert. In den Populationen der Modellerkrankungen unterschieden sich die soziodemographischen Eigenschaften in wenigen markanten Variablen ([Tabelle 5](#)). Neben unterschiedlichen Hospitalisierungshäufigkeiten fällt auf, dass beispielsweise jeder COPD-Patient etwa zwei Index-Fälle lieferte, wohingegen OS-Patienten und Patienten mit AMI durchschnittlich etwa einen Index-Fall beitrugen. Die Anzahl an männlichen Patienten beim ersten Index-Fall unterschied sich erwartungsgemäß je nach Erkrankung erheblich, mit dem geringsten Anteil innerhalb der OS-Gruppe und dem höchsten Anteil innerhalb der COPD-Population. Das durchschnittliche Alter war dagegen vergleichbar.

Tabelle 5: Soziodemographische Charakterisierung der Erkrankungspopulationen anhand ihrer Index-Fälle.

	COPD	Herz-insuffizienz	Akuter Myokardinfarkt	Osteoporose	Diabetes mellitus Typ 2	Schlaganfall, TIA oder Vorhofflimmern
Anzahl Index-Fälle	28 446	80 870	18 893	6 111	23 665	81 632
Anzahl Patienten (gesamt)	15 778	52 890	16 991	5 321	17 530	65 750
Anzahl männlicher Patienten (%)	8 688 (55,06)	22 151 (41,88)	8 965 (52,76)	767 (14,41)	8 049 (45,92)	26 974 (41,03)
Alter* (Mittelwert ± Standardabweichung)	77,98 ± 6,43	81,72 ± 6,69	79,54 ± 6,72	81,25 ± 6,4	78,79 ± 6,47	79,51 ± 6,6

COPD: chronisch obstruktive Lungenerkrankung; **TIA:** Transitorische ischämische Attacke. *Ein Patient konnte mehrere Index-Fälle beitragen. Die in diesem Kapitel durchgeführte Analyse sollte jedoch die Patientenpopulation bestmöglich charakterisieren. Um die Versichertenpopulation für jede Erkrankung beschreiben zu können, wurde festgelegt, dass jeder Patient einmalig, und zwar zum Zeitpunkt des ersten Auftretens im jeweiligen Erkrankungs-Datensatz, d. h. zum Zeitpunkt des ersten Index-Falls, charakterisiert wird. Durch die so ermittelten Baseline-Charakteristika zum Eintrittszeitpunkt konnten die unterschiedlichen Erkrankungspopulationen bestmöglich miteinander verglichen werden.

Die Populationen unterschieden sich hinsichtlich der aus den Nebendiagnosen ermittelten Elixhauser-Komorbiditäten erheblich ([Abb. 8](#)), wobei zwei Modellerkrankungen herausstachen: Patienten mit HF hatten überdurchschnittlich viele Komorbiditäten, Patienten mit OS deutlich weniger als die anderen Patientenpopulationen. Betrachtet man die Patienten mit COPD, HF oder AMI, so ähnelten sich diese in ihren Mustern von Komorbiditäten (die vollständige Auflistung der Anteile aller untersuchten Komorbiditäten pro Erkrankungspopulation findet sich in [Anhang 3](#)). Zusammenfassend litten die einzelnen Erkrankungspopulationen vor allem unter erwarteten Komorbiditäten, die den typischen Komorbiditätsmustern in der Bevölkerung entsprachen, und sich häufig als kardiovaskuläre Komorbidität äußerten.

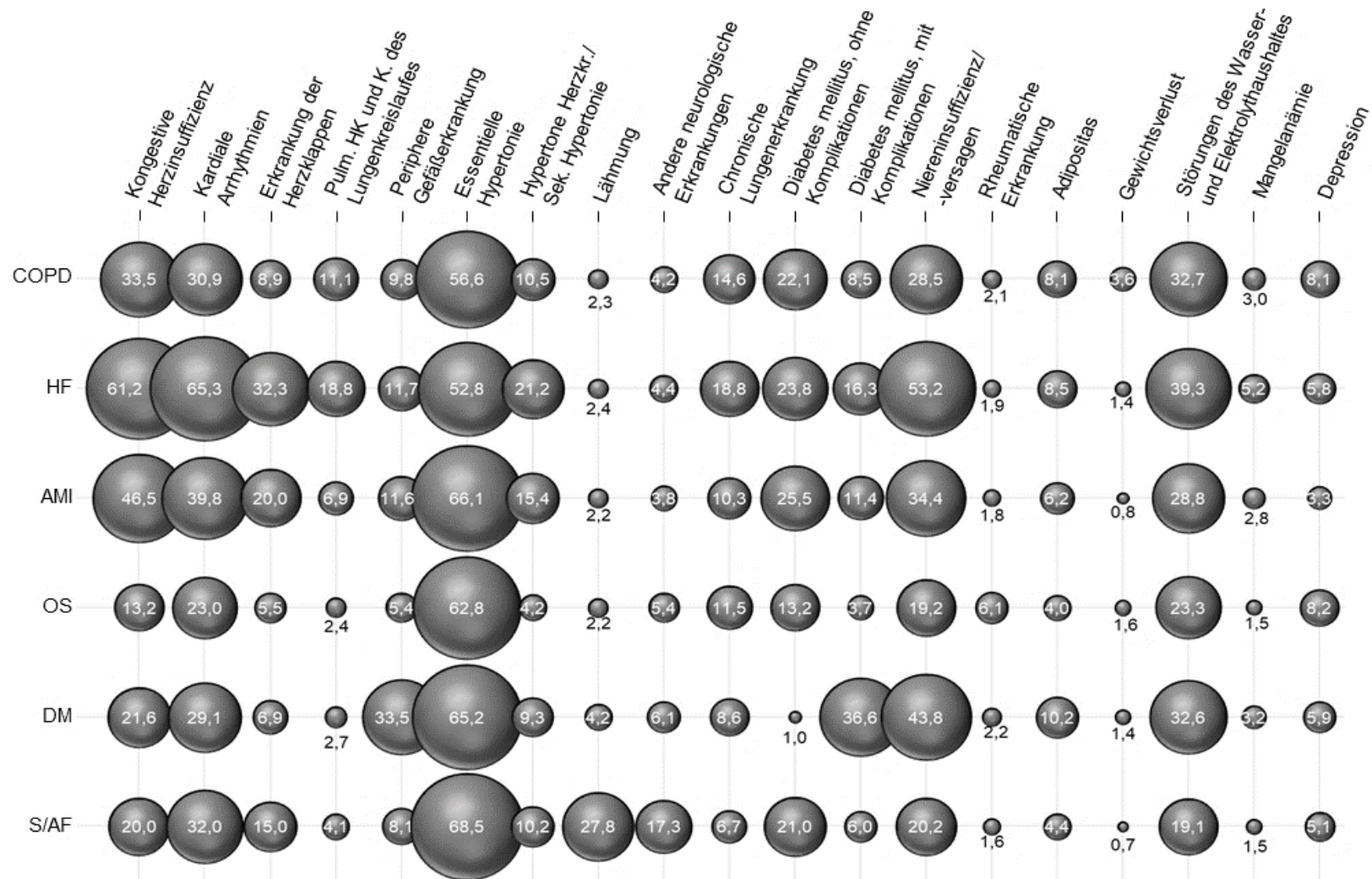


Abb. 8: Anteile der Index-Patientenpopulationen mit Komorbiditäten nach den Elixhauser Gruppen (aus Nebendiagnosen; Auswahl). Dargestellt ist der Anteil an Patienten mit Index-Erkrankung (Zeile), welche gleichzeitig auch an der aufgeführten Komorbidität (Spalte) litten. Index-Erkrankungen wurden durch die Hauptdiagnose des Index-Falls bestimmt. **AMI:** Akuter Myokardinfarkt; **COPD:** Chronisch obstruktive Lungenerkrankung; **DM:** Diabetes mellitus Typ 2; **HF:** Herzinsuffizienz; **HK:** Herzkrankheit; **Kr.:** Krankheiten; **OS:** Osteoporose; **Pulm.:** Pulmonale; **S/AF:** Schlaganfall, transitorische ischämische Attacke oder Vorhofflimmern; **Sek.:** Sekundäre.

6.3 Vorhersage von Rehospitalisierungen für sechs Modellerkrankungen und Prüfung der Generalisierbarkeit

Als explorativer Schritt vor der Identifikation von Risikopatienten für spezifische Rehospitalisierungen mithilfe von Prädiktionsmodellen wurden zunächst Unterschiede in Kandidatenprädiktoren zwischen spezifisch rehospitalisierten und nicht-rehospitalisierten Patienten untersucht. Zwischen diesen nach Outcome stratifizierten Gruppen unterschied sich die Anzahl an Index-Fällen zwischen den einzelnen Erkrankungen deutlich (**Anhang 3**). So war der Unterschied bei COPD-Patienten am größten (etwa 1,7 Index-Fälle mit spezifischer Rehospitalisierung und 1,4 Index-Fälle ohne spezifische Rehospitalisierung). Dies bedeutet, dass mit der entwickelten Klassifikation COPD-Patienten vorrangig wegen ihrer Grunderkrankung rehospitalisiert werden, was klinisch gut nachvollziehbar erscheint. Das durchschnittliche Alter und das Geschlechterverhältnis war dagegen über die Modellerkrankungen hinweg vergleichbar. Erwartungsgemäß lag die Anzahl der Hospitalisierungen im vorherigen Jahr für die Index-Fälle mit spezifischer Rehospitalisierung immer über der Anzahl der Hospitalisierungen im vorherigen Jahr für die Index-Fälle ohne Rehospitalisierung (bis zu 1,13 Hospitalisierungen mehr). COPD- und HF-Patienten hatten die höchste Anzahl an Hospitalisierungen im Jahr vor dem Index-Fall. In den aus den Nebendiagnosen der Index-Fälle bestimmten Komorbiditäten gab es große Unterschiede zwischen Index-Fällen mit oder ohne Rehospitalisierung.

Vorrangig projektrelevant waren jedoch potenziell beeinflussbare Risikofaktoren, etwa die Arzneimitteltherapie abgebildet über die STOPP-/ START-Kriterien. Die Prävalenzen der verschiedenen START-Kriterien waren sehr unterschiedlich verteilt, wobei über alle Erkrankungen hinweg kardiovaskuläre START-Kriterien (Sektion START „A“) erwartungsgemäß mit hohen Prävalenzen vertreten waren (**Abb. 9**). Das indikationsspezifische START-Kriterium E4 (Antiresorptive oder osteoanabole Therapie bei OS) wies erwartungsgemäß die höchsten Prävalenzen in der OS-Population auf (70,33 % der Index-Fälle für OS mit spezifischer Rehospitalisierung). Erwartungsgemäß könnte das Fehlen einer Therapie für eine bestimmte Indikation und somit das Vorliegen eines START-Kriteriums gesundheitliche Nachteile (und damit auch Rehospitalisierungen) haben. Index-Fälle mit nachfolgender spezifischer Rehospitalisierung müssten zu höheren Anteilen ein indikationsspezifisches START-Kriterium aufweisen als Index-Fälle ohne spezifische Rehospitalisierung (z. B. müsste START-Kriterium A3 – Antithrombotische Therapie bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung – zu einem größeren Anteil bei Index-Fällen mit spezifischer Rehospitalisierung für AMI als bei Index-Fällen ohne spezifische Rehospitalisierung für AMI auftreten). Hinsichtlich ihrer Prävalenz zwischen Index-Fällen mit und Index-Fällen ohne Rehospitalisierung unterschieden sich START-Kriterien jedoch nicht oder sehr gering (**Abb. 9**).

Von den 49 operationalisierten STOPP-Kriterien lieferten 30 Kriterien Verschreibungsanteile von mindestens 1 % in mindestens einer der 6 Krankheitspopulationen. In allen Fallpopulationen lag der Anteil von elf STOPP-Kriterien über 1 % (B8, B9, B12, C3, C10, C11, E3, E4, E6, F3, H5), von drei STOPP-Kriterien über 5 % (B8, C3, H5), so dass diese Kriterien unabhängig von der Index-Erkrankung relevant erschienen. Die Erfüllung eines STOPP-Kriteriums, d. h. das Vorliegen einer Erkrankung zusammen mit einem ungeeigneten Arzneimittel oder einer ungeeigneten Arzneimittelkombination, sollte erwartungsgemäß vermehrt zu gesundheitlichen Ereignissen (z. B. Rehospitalisierungen) führen. Indikations-spezifische STOPP-Kriterien (z. B. STOPP G2 – systemische Glucocorticoide als Erhaltungstherapie bei COPD) müssten daher bei Index-Fällen mit spezifischer Rehospitalisierung zu einem höheren Anteil vertreten sein als bei Index-Fällen ohne spezifische Rehospitalisierung (**Abb. 10**). Solche Assoziationen könnten auch für die folgenden Modelle relevant sein.

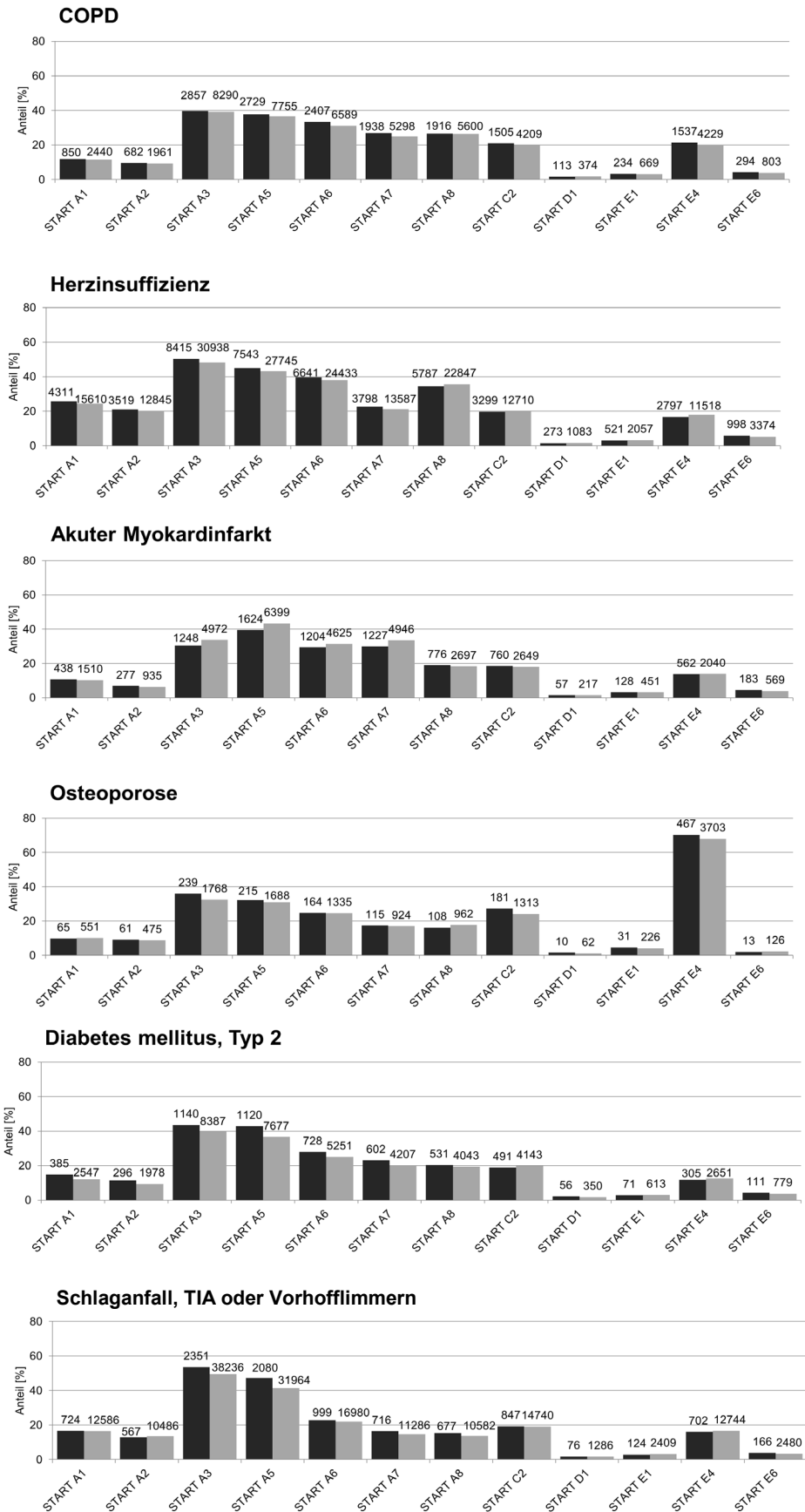


Abb. 9: Anteil der Index-Fälle mit Exposition zu START-Kriterien (Auswahl; schwarz: Anteil an Index-Fällen mit spezifischer Rehospitalisierung; grau: ohne spezifische Rehospitalisierung) (absolute Fallzahlen sind über den zugehörigen Balken angegeben).

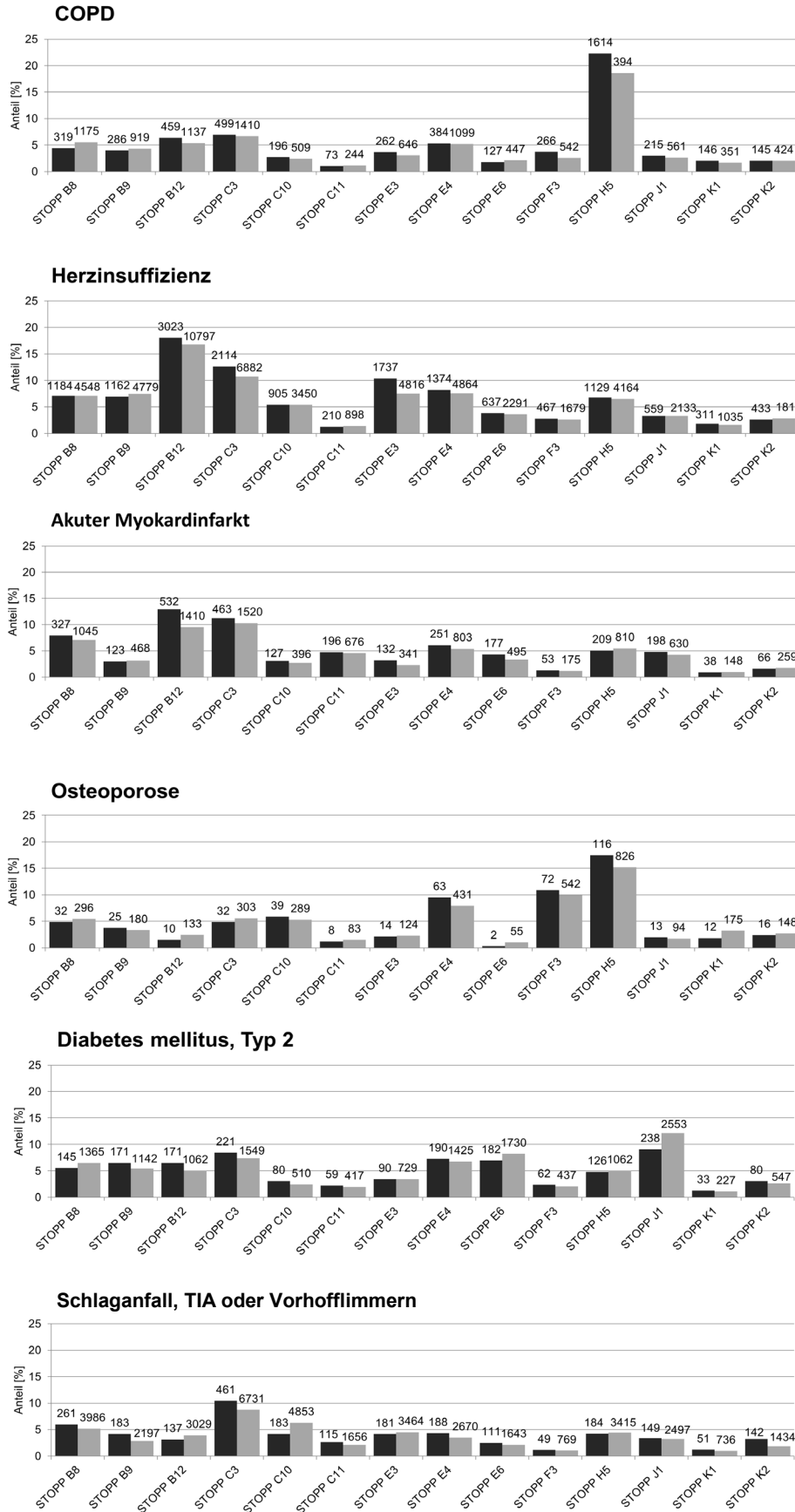


Abb. 10: Anteil der Index-Fälle mit Exposition zu STOPP-Kriterien (Auswahl; schwarz: Anteil an Index-Fällen mit spezifischer Rehospitalisierung; grau: ohne spezifische Rehospitalisierung) (absolute Fallzahlen sind über den zugehörigen Balken angegeben).

Für jede Erkrankung wurde je ein volles und ein regularisiertes Regressionsmodell für das Outcome „spezifische Rehospitalisierung“ entwickelt, so dass auch der Einfluss einzelner Kandidatenprädiktoren vergleichend beurteilt werden konnte (Gerharz et al., 2022) (**Anhang 4** listet die letztlich selektierten Variablen mit Parameterschätzern auf). Die prädiktive Modellierung ergab verschiedene Modelle unterschiedlicher Größe (von fünf Prädiktoren aus der STOPP-/ START-Liste im DM-Modell bis zu 32 Prädiktoren im HF-Modell (**Tabelle S6** und **Abbildung S1** in Gerharz et al. 2022)). Keines der operationalisierten STOPP-/START-Kriterien war in allen (regularisierten) Regressionsmodellen über alle Erkrankungen hinweg prädiktiv für eine spezifische Rehospitalisierung. Außerdem gab es Kriterien, die in keinem Modell prädiktiv waren bzw. nicht durch das LASSO-Verfahren selektiert wurden. Hierbei handelte es sich um die START-Kriterien START A1, START A6 und START E1 sowie um die STOPP-Kriterien STOPP E1 und STOPP J2. START-Kriterien waren unter den Prädiktoren für eine Rehospitalisierung binnen 90 Tagen weitaus seltener als STOPP-Kriterien. Um die Güte der Voraussagen zu vergleichen, wurden c-Statistiken berechnet, welche im Median in zehnfacher Kreuzvalidierung bis maximal 0,67 (LASSO-Modell für DM) reichten. Kleinere Modelle zeigten einen größeren Nutzen der regularisierten LASSO-Modelle (**Abbildung 11A**), und die CV-Performance zeigte die beste Diskriminationsfähigkeit für das DM-Modell und das COPD-Modell (mediane CV-c-Statistik von 0,67 bzw. 0,65). Der niedrigste Median wurde für die spezifische Rehospitalisierung für OS erreicht. Dieses Muster war darüber hinaus bei der Split-Sample-Validierung konsistent (**Abbildung 11B**). Der inkrementelle Vorhersagewert von Arzneimittelprädiktoren gegenüber einem Basismodell für Komorbidität war jedoch bei HF und COPD am höchsten, während sie nur wenig zur Vorhersage von Wiederaufnahmen im Zusammenhang mit DM beitragen (**Abbildung S2** in Gerharz et al. 2022).

Um als Ausblick gezielt Interventionen zur Behebung der pharmakotherapeutischen Risikofaktoren zu initiieren und in ihrer Wirksamkeit zu beurteilen helfen die Sensitivitäts- und Spezifitätswerte beim jeweiligen Schwellenwert t (**Abb. 11B**). Daraus lässt sich der Anteil richtig ausgewählter Interventionen an den Gesamtinterventionen ermitteln. Für ein gegebenes Szenario mit Kenntnis einer Intervention (Erfolgsrate und Kosten) sowie den Fallkosten für eine Rehospitalisierung lassen sich erwartete Einsparungen durch die Interventionsstrategie wie folgt berechnen:

$$\begin{aligned} \text{Einsparungen} &= \text{Erwarteter Nutzen} - \text{Kosten} \\ &= \left(\frac{\text{Anzahl richtig-ausgewählte Interventionen}}{\text{Interventionen}} \right) \times \left(\frac{\text{Interventions-}}{\text{Erfolgsrate}} \right) \\ &\quad \times \left(\frac{\text{Rehosp.-}}{\text{kosten}} \right) - \left(\frac{\text{Anzahl}}{\text{Gesamtinterventionen}} \right) \times \left(\frac{\text{Interventions-}}{\text{kosten}} \right) \end{aligned}$$

Damit lassen sich Einsparpotenziale auch für lokale Gegebenheiten (verfügbare Interventionsmöglichkeiten, spezifische Rehospitalisierungskosten) generieren. Auch kann erörtert werden, welche Intervention in einer spezifischen Erkrankung als kosteneffizient gelten kann.

Verallgemeinernd über die Modellerkrankungen hinweg wurde ermittelt, wie gut die Risiken aus krankheitsspezifischen Modellen für eine spezifische Rehospitalisierung auch eine Rehospitalisierung jeglicher Ursache vorhersagen können (Prüfung der Generalisierbarkeit). Hierfür wurde der Ansatz der *gestackten* Modelle erfolgreich durchgeführt, indem die Rehospitalisierungs-wahrscheinlichkeiten aus den krankheitsspezifischen Modellen in einem („stacked“) GLM integriert wurden. Dieses sagte die Rehospitalisierung jeglicher Ursache mit einer Split-Sample-C-Statistik von 0,63 [0,63-0,64] voraus (**Abbildung S3** in Gerharz et al. 2022).

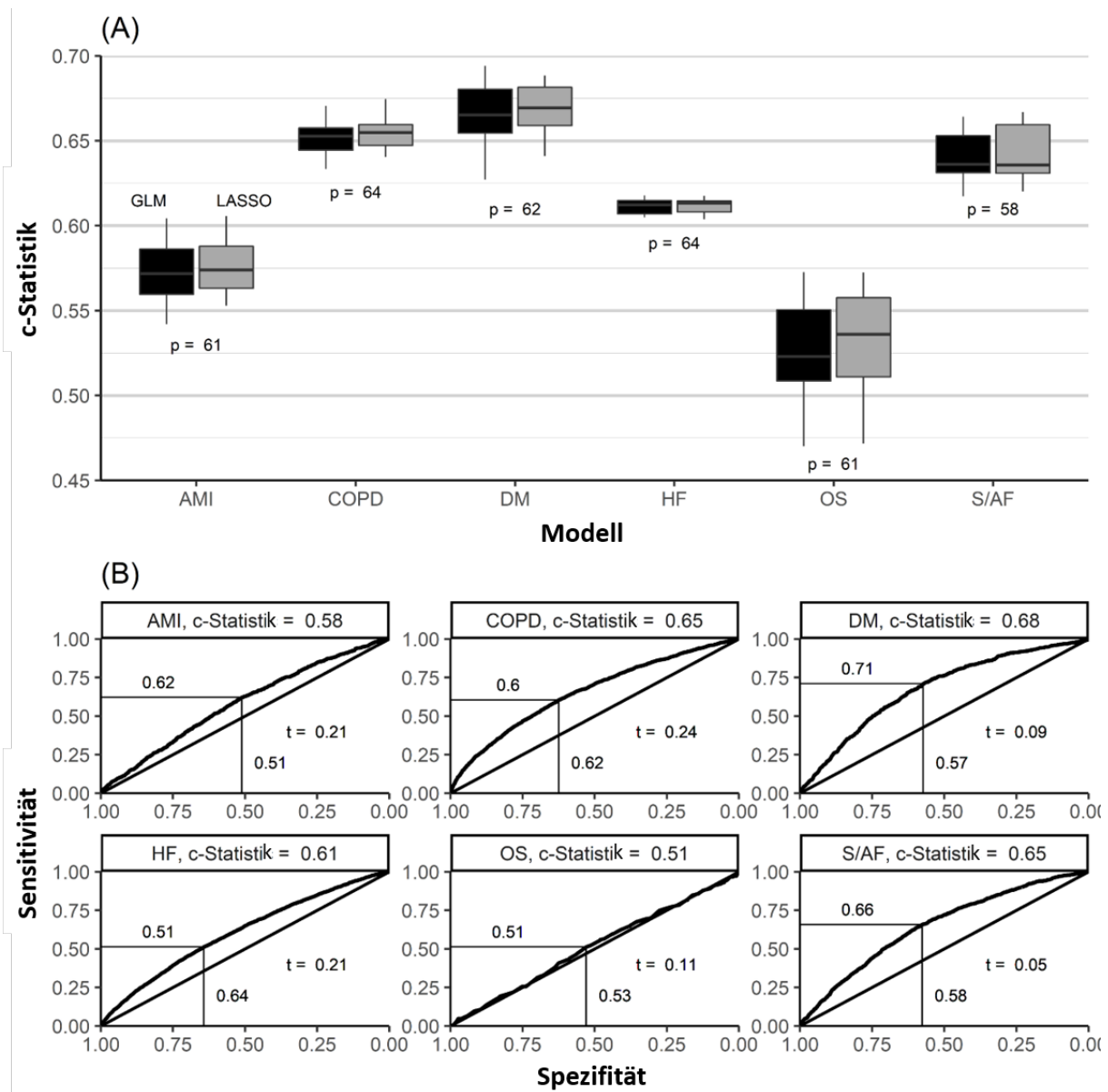


Abb. 11: Performance der entwickelten Vorhersagemodelle für spezifische Rehospitalisierungen bei sechs Modellerkrankungen: (A) Ein Boxplot zeigt Performanceschätzungen in Form von c-Statistiken, die aus einer 10-fachen Kreuzvalidierung gewonnen wurden, wobei p die Anzahl der Kandidatenvariablen für Prognosemodelle auf der Grundlage des LASSO (Hauptanalyse, grau gefüllte Kästchen) und eines Standard-GLM (Sekundäranalyse, schwarz gefüllte Kästchen) bezeichnet. (B) Receiver-Operator-Charakteristik-Kurven wurden aus einer Split-Sample-Validierung erhalten, die t als besten Schwellenwert auf der Grundlage der höchsten Schätzungen für Sensitivität und Spezifität neben der jeweiligen Achse (Youden-Index) angibt. Die entsprechenden c-Statistiken waren 0,58 [95 % Konfidenzintervall [0,56-0,60]] für AMI (akuter Myokardinfarkt), 0,65 [0,64-0,67] für COPD (chronisch obstruktive Lungenerkrankung), 0,68 [0,66-0,70] für DM (Typ-2-Diabetes mellitus), 0,61 [0,60-0,62] für HF (Herzinsuffizienz), 0,51 [0,47-0,56] für OS (Osteoporose) und 0,65 [0,63-0,66] für S/AF (eine Kombination aus Schlaganfall, transitorischer ischämischer Attacke oder Vorhofflimmern).

6.3.1 Modellvergleiche von Verfahren aus dem Bereich künstliche Intelligenz und Machine-Learning

In **Tabelle 6** sind die Ränge der mittleren Performannewerte der Methoden für jede Modellerkrankung angegeben. Osteoporose stellte einen schwierigen Sonderfall für die Vorhersage dar, so dass in dieser Tabelle für jede Methode ein durchschnittlicher Rang für alle Krankheiten und ein durchschnittlicher Rang für alle Krankheiten außer Osteoporose angezeigt wird. Hier zeigte sich, dass insbesondere Methoden mit einem Variablenselektionsschritt sehr gut geeignet waren. Die beste Performance wurde erreicht, indem die Variance-Importance aus einem Permutationsverfahren bei einem Random-Forest verwendet wurde, um alle Variablen in eine Rangfolge zu bringen. Daraufaufgehend wurde eine Hyperparameter-Tuning angewendet, das bewertet, wie viele Variablen in das aufeinanderfolgende GLM aufgenommen werden sollten, beginnen mit der Wichtigsten. Die Variablenauswahl mit Hilfe der GINI-Impurity-Decrease-Variable-Importance und einem anschließenden GLM schnitt ebenfalls gut ab, aber etwas schlechter als die Variable-Importance aus einem Permutationsverfahren. Die nächstbeste Methode war das LASSO, das zu den klassischen Regularisierungsmethoden gehört, die ebenfalls eine Variablenauswahl beinhalten. Besonders interessant ist, dass die beiden Krankheiten, bei denen das Modell am besten abschnitt, auch diejenigen mit der größten Fallzahl waren. RIDGE schnitt ebenfalls gut ab, aber insgesamt etwas schlechter als LASSO. Im Bereich des maschinellen Lernens und der Boosting-Methoden schnitten beide Boosting-Methoden besser ab als die anderen Methoden. GLMBOOST und XGBOOST beinhalten eine intrinsische Variablenauswahl, da in jedem Schritt des Modellbildungsprozesses nur die besten Variablen verwendet werden. Insbesondere GLMBOOST schnitt sehr gut ab und sollte als mögliche Methode für diese Modellierungsaufgaben in Betracht gezogen werden, während XGBOOST nur für eine der Krankheiten wirklich gut abgeschnitten hat und für die anderen Krankheiten im Durchschnitt liegt. Obwohl CART und RF auch eine intrinsische Variablenauswahl beinhalten, schnitten diese Methoden in diesem Vergleich nicht gut ab. Da CART, RF, XGBOOST und die CARTs, die auf der Variablenwichtigkeit basierende Variablenselektion verwenden, alle zu den baumbasierten Methoden zählen und nicht gut abschnitten, ist dies ein Indikator dafür, dass baumbasierte Methoden für diese Modellierungsaufgabe möglicherweise keine gute Wahl sind.

6.3.2 Einfluss der Adhärenz mit leitliniengerechter Medikation in medikationsbasierten Analysen von Rehospitalisierungen bei Herzinsuffizienz und Herzinfarkt

Es gab positive lineare Korrelationen zwischen den Adhärenzschätzungen der Medikamentenklassen innerhalb der Kohorten, wobei die Adhärenzeffekte im Allgemeinen bei HF-Patienten stärker ausgeprägt waren als bei MI-Patienten. Eine gute Vorhersageleistung wurde durch große Flächen unter der Empfängerbetriebskennlinie (ROC) von 0,763 (MI) und 0,868 (HF) indiziert. Explorative Analysen ergaben, dass die Einbeziehung der Medikamententreue in die Vorhersagemodelle deren Leistung deutlich verbesserte (**Abb. 12**).

Tabelle 6: Relatives Abschneiden im Methodenvergleich der prädiktiven Performance als Rang unter verschiedenen Verfahren in sechs Modellerkrankungen.

	AMI	HF	S/AF	COPD	DM	OS	ØRang	ØRang (ohne OS)
Baseline	15	15	15	15	15	9	14,00	15,0
GLM	9	2	7	5	6	3	5,33	5,8
LASSO	2	1	1	3	4	6	2,83	2,2
RIDGE	7	7	3	7	2	1	4,50	5,2
relaxed-LASSO	8	4	6	8	10	2	6,33	7,2
kNN	13	11	14	14	14	15	13,50	13,2
CART	11	14	13	12	12	13	12,50	12,4
RF	10	10	10	9	9	10	9,67	9,6
NEUR NET	4	6	9	6	7	7	6,50	6,4
GLMBOOST	5	8	4	4	3	11	5,83	4,8
XGBOOST	1	9	8	10	8	8	7,33	7,2
GINI GLM	6	5	5	2	5	5	4,67	4,6
GINI CART	14	12	12	13	13	14	13,00	12,8
PERM GLM	3	3	2	1	1	4	2,33	2,0
PERM CART	12	13	11	11	11	12	11,67	11,6

AMI: Akuter Myokardinfarkt; **HF:** Herzinsuffizienz; **S/AF:** Schlaganfall, transitorische ischämische Attacke oder Vorhofflimmern; **COPD:** Chronisch obstruktive Lungenerkrankungen; **DM:** Diabetes mellitus Typ 2; **OS:** Osteoporose; **GLM:** Generalisiertes Lineares Modell; **LASSO:** least absolute shrinkage and selection operator Regression; **RIDGE:** Ridge Regression; **relaxed-LASSO:** Sonderform LASSO; **kNN:** k-nearest-neighbors; **CART:** Classification and Regression Tree; **RF:** Random Forest; **NEUR NET:** Neuronales Netz; **GLMBOOST:** Boosting Variante für GLMs; **XGBOOST:** eXtreme Gradient Boosting; **GINI GLM:** GLM nach Variablenselektion mit GINI-Impurity; **GINI CART:** CART nach Variablenselektion mit GINI-Impurity; **PERM GLM:** GLM nach Variablenselektion mit Permutationsverfahren; **PERM CART:** CART nach Variablenselektion mit Permutations-Verfahren.

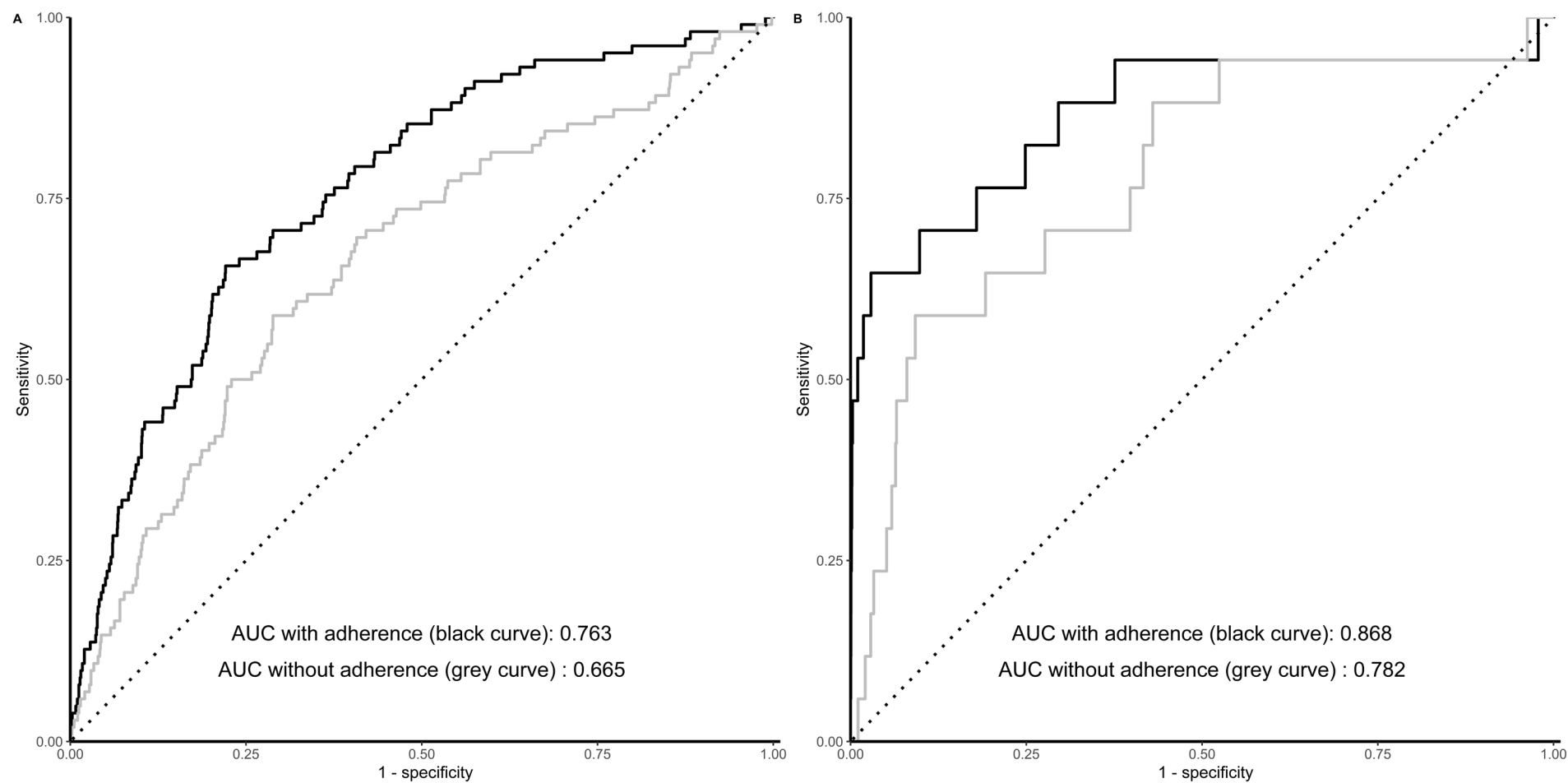


Abb. 12: ROC-Kurven der Vorhersagemodelle für Rehospitalisierungen nach akutem Myokardinfarkt (A) und Herzinsuffizienz (B) mit und ohne Betrachtung der Adhärenzprädiktoren.

7. Diskussion der Projektergebnisse

Ergebnisse und Einordnung in die Literatur

Die ermittelten Rehospitalisierungsraten innerhalb von 30 und 90 Tagen lassen sich gut bei Betrachtung der Hospitalisierung aufgrund jeglicher Ursache in die Literatur einordnen; in bereits publizierten Arbeiten wurden vergleichbare Schätzer veröffentlicht, so z. B. für Rehospitalisierungen innerhalb von 30 Tagen für HF (Bottle, Honeyford, Chowdhury, Bell, & Aylin, 2018; Jencks, Williams, & Coleman, 2009), AMI (Dodson et al., 2019; Tung, Chang, Chang, & Yu, 2017), COPD (Bottle et al., 2018; Jencks et al., 2009) und DM (Enomoto, Shrestha, Rosenthal, Hollenbeak, & Gabbay, 2017; Karunakaran, Zhao, & Rubin, 2018; Ostling et al., 2017; Png et al., 2018). Das Verhältnis von spezifischen zu nicht-spezifischen Rehospitalisierungen war innerhalb der Beobachtungsperiode von 90 Tagen konstant. Kardiovaskuläre Komorbiditäten waren erwartungsgemäß besonders prominent in der AMI-Erkrankungspopulation anzutreffen (Thygesen et al., 2018), was gleichzeitig zu den Gründen für Rehospitalisierung beigetragen haben kann. Die Kontrolle und Vermeidung von Risikofaktoren für einen AMI sind in diesem Fall womöglich am vielversprechendsten für die Prävention, um das kardiovaskuläre Risiko zu senken und somit auch (Re-)Hospitalisierungen zu vermeiden (Deutsche Gesellschaft für Allgemeinmedizin und Familienmedizin (DEGAM), 2017a). Auch nachvollziehbar war die (kardiovaskuläre) Komorbiditätslast der COPD-Patienten mit kardiovaskulären (HF, Hypertonie) und metabolischen (DM) Erkrankungen (Bundesärztekammer, 2020). DM führte nur selten zu spezifischen Rehospitalisierungen, sondern eher zu Rehospitalisierungen, die durch kardiovaskuläre Erkrankungen ausgelöst wurden, wie beispielsweise HF. Es scheint daher geboten, DM-Patienten auf kardiovaskuläre Komorbiditäten zu screenen und auch die dazugehörige Medikation zu überwachen, damit Rehospitalisierungen potenziell vermieden werden können. Die hier ebenfalls häufig beobachteten Nierenerkrankungen sind auch aus der Literatur zu erwarten (Braunwald, 2019; Doshi & Friedman, 2017; Dumbreck et al., 2015) und könnten (z. B. über das ICD-Kapitel „Krankheiten des Urogenitalsystems“) häufig zu Rehospitalisierungen geführt haben. Somit erscheint eine adäquate Therapie des DM zur Vermeidung von renalen Folgeerkrankungen ebenfalls wichtig. Die spezifische Rehospitalisierungsrate lag bei HF (mit COPD und AMI) am höchsten. Hieraus kann man als Vermeidungsstrategien die Befolgung geeigneter Leitlinien, multidisziplinäre Versorgung des Patienten und die Durchführung von Interventionen ableiten, welche ein Case-Management beinhalten (Takeda, Martin, Taylor, & Taylor, 2019). Aus dem erwarteten Komorbiditätsmuster der HF-Patienten (kardiovaskulär, metabolisch, renale Erkrankungen) (Angermann, 2009; Dumbreck et al., 2015) und der Tatsache, dass es sich bei der HF um eine Systemerkrankung handelt, lässt sich die Empfehlung bekräftigen, multidisziplinäre Ansätze zur Versorgung dieser Patienten zu nutzen, um häufige Hospitalisierungen aufgrund von HF zu vermeiden (Shadmi et al., 2015).

Die Ergebnisse der deskriptiven Beschreibungen im Projekt lassen bereits zwei wichtige Strategien zur Vermeidung von Rehospitalisierungen erkennen: Zum einen scheinen krankheitsindividuelle Maßnahmen erforderlich, da Rehospitalisierungen häufig für die Index-Erkrankung (oder eine spezifische damit zusammenhängende Diagnose) stattfinden. Aufgrund der ausgeprägten Komorbiditätslast und vielen Rehospitalisierungen ohne direkten Bezug zur Index-Erkrankung scheint daneben ein ganzheitlicher Behandlungsansatz notwendig (Deutsche Gesellschaft für Allgemeinmedizin und Familienmedizin (DEGAM), 2017b). Zum anderen legen die Erkenntnisse nahe, dass ein allgemeines Screening der Patienten > 65 Jahre auf die „Volkskrankheiten“ und typische alters-assoziierte Erkrankungen essentiell sein sollte. Damit könnten rechtzeitig Arzneimitteltherapien angepasst werden (etwa aufgrund von verminderter Nierenfunktion) und letztlich mögliche Folgen der Volkskrankheiten (z. B. Myokardinfarkt, Schlaganfall als Folgen von Hypertonie und anderen vaskulären Risikofaktoren) verhindert werden.

In der explorativen Datenanalyse vor Entwicklung der Vorhersagemodelle war die Anzahl an Hospitalisierungen im Jahr vor dem Index-Fall für COPD- und HF-Patienten am höchsten. Diese Erkenntnisse bestätigen die bisherige Literatur, in welcher COPD und HF chronische Erkrankungen mit häufigen Exazerbationen sind; häufige Krankenhausaufenthalte sind die Folge davon (Shadmi et al., 2015). Die Variable „Anzahl der Hospitalisierungen im Jahr vor dem Index-Fall“ gilt als starker und häufig verwendeter Prädiktor in Vorhersagemodellen (Allen et al., 2010; Amarasingham et al., 2010; Cubbon et al., 2014; Garcia-Aymerich et al., 2003; Hebert et al., 2014; Krumholz et al., 2000; Krumholz et al., 2011; Lau, Yam, & Poon, 2001; Rana et al., 2014; Robbins & Webb, 2006; Yu et al., 2015). Die in der vorliegenden Arbeit gefundene durchschnittliche Anzahl von 2,62 Hospitalisierungen im Jahr vor der Index-Hospitalisierung für COPD liegt höher als in der Literatur beschrieben, weil dort nur spezifische Hospitalisierungen berücksichtigt wurden (Gudmundsson et al., 2005). Mit diesem Vorgehen könnte zukünftig auch eine Population hinsichtlich des Schweregrads der betreffenden Erkrankung besser charakterisiert werden.

Betrachtet man die Kandidatenprädiktoren für die Vorhersagemodelle, so traten die STOPP-/START-Kriterien sehr prävalent auf. Dies traf besonders auf die kardiovaskulären STOPP- / START-Kriterien zu, was wiederum mit dem Komorbiditätsmuster verknüpft sein mag. Ein Screening wie in diesem Projekt könnte einen ersten Schritt zur Prävention von Unterversorgung und Non-Adhärenz darstellen (Hess et al., 2014; Wilke et al., 2015). Für die finalen Vorhersagemodelle wurden START-Kriterien dagegen selten ausgewählt. Dies konnte mit dem Wissensstand aus der Literatur so erwartet werden, wo START-Kriterien manchmal nicht mit einem erhöhten Hospitalisierungsrisiko assoziiert waren (Brunetti et al., 2019; van der Stelt et al., 2016). Man muss dies jedoch vor dem Hintergrund betrachten, dass sich präventive Effekte von vieler Arzneimitteltherapien oft erst nach Monaten bis Jahren zeigen (Black et al., 2020) und die untersuchten Zeiträume von 90 Tagen (dieses Projekt) bzw. sechs Monaten (Brunetti et al., 2019) dies nicht abbilden können. Dagegen waren (indikationsspezifische) STOPP-Kriterien in Fällen mit spezifischer Rehospitalisierung oft zu statistisch signifikant höheren Anteilen vertreten als in Fällen ohne spezifische Rehospitalisierung. Weil dies aber inhaltlich auch über die Grunderkrankung hinausging, wird abermals deutlich, dass eine ganzheitliche und dem Krankheitsstatus angepasste Behandlung des Patienten essentiell scheint, um eine möglichst gute Versorgung sicherzustellen und schädliche Ereignisse abzuwenden. Viele in den Modellen ausgewählte Kriterien sind jedoch zur Grunderkrankung passend und bestätigen einschlägige Leitlinien. Ein Beispiel hierfür ist etwa das Risiko durch systemische Glucocorticoide bei COPD, wo stattdessen eine inhalative Therapie zu bevorzugen ist (Bundesärztekammer, 2020). Effekte von STOPP-/START-Kriterien, die für alle Prädiktionsmodelle durchgehend gleichermaßen gültig waren, lagen nicht vor. Generell zeichnet sich auch ein heterogenes Bild in der Literatur. Während die STOPP-/START-Kriterien prädiktiv für Hospitalisierungen waren (Brown, Hutchison, Li, Painter, & Martin, 2016), ist die Datenlage bezüglich Rehospitalisierungen weniger eindeutig (Hyttinen, Jyrkka, & Valtonen, 2016).

Die Vorhersagegüte der Modelle lag mit den erreichten AUC auf einem zufriedenstellenden Niveau, welches so auch in der Literatur bereits für Rehospitalisierungen jeglicher Ursache innerhalb von 90 Tagen beschrieben ist (Ena et al., 2018; Formiga, Masip, Chivite, & Corbella, 2017; Sawhney et al., 2017). Für HF und DM sind auch bereits Ergebnisse mit einer besseren Diskriminationsfähigkeit zu finden (Kitamura et al., 2017; Xu et al., 2019), welche jedoch aus einer deutlich stärker selektierten HF-Population stammen (Ausschluss von Herzschrittmacheroperation, Entlassung nicht nach Hause) und teilweise Variablen nutzen, welche in Abrechnungsdaten nicht zugänglich sind (z. B. Hämoglobinwert). Insgesamt sind bei weitem nicht so viele Prädiktionsmodelle für eine Rehospitalisierung innerhalb von 90 Tagen publiziert (Artetxe, Beristain, & Grana, 2018) und generell ist die Vergleichbarkeit durch unterschiedliche Ansätze eingeschränkt.

Limitationen und Ausblick

Die eben genannte limitierte Performance kann als nicht ausreichend erachtet werden, wenn ein solcher routinedatenbasierter Ansatz mit der weitaus besseren Performance von Modellen mit Krankenhausdaten verglichen wird. Insofern haben die entwickelten Modelle noch immer Entwicklungscharakter und sind weiter zu optimieren. Dann sind auch Kennzahlen zur Modellkalibrierung von Interesse, sobald die Entwicklung abgeschlossen ist und als zufriedenstellend für eine nachfolgende Implementierung betrachtet wird. Kennzahlen zur Modellkalibrierung wurden aus diesem Grund bisher noch nicht berechnet. Auch weitere Limitationen der vorliegenden Arbeit können als Ansatzpunkte für darauf aufbauende Projekte dienen. Die in dieser Arbeit neu entwickelte Methodik und die damit erstellte Datenbasis konnten (noch) nicht validiert werden. Retrospektive Vergleiche sind nicht zielführend (Dorning, Bitzer, Hofmann, & Schwartz, 1995), da während dieses Zeitraums eine ganz andere Gesundheitsversorgung mit anderen Bedingungen herrschte. Auch die Abrechnungsdaten eines einzelnen Krankenhauses (Inhouse-Daten (Dormann et al., 2004)) würden nicht ausreichen, wenn das Ziel ein generalisierter Ansatz von Rehospitalisierungen in jeglichem möglichen Krankenhaus ist.

Wünschenswert wäre daher ein Datensatz mit allen Fällen eines Patienten vor möglicherweise erfolgter Fallzusammenführung, was alle Behandlungsperioden des Patienten in jeglichen Krankenhäusern einschließen würde. Aufgrund der Abrechnungsbestimmungen kann dies jedoch nicht ohne Weiteres vorliegen. So legt § 301 Abs. 3 SGB V Durchführungshinweise zur Vereinbarung der Datenübermittlung fest, wonach jedes Krankenhaus der jeweiligen Krankenversicherung die Aufnahme eines Versicherten bekanntgeben muss und innerhalb von drei Arbeitstagen nach Entlassung oder Verlegung eine Entlassungsanzeige an die Krankenversicherung übermitteln muss (GKV-Spitzenverband, 2018). Falls ein übermittelndes Krankenhaus einen Fall jedoch neu einstufen muss (wenn z. B. die Bedingungen einer Fallzusammenführung zutreffen), dann wird der erste übermittelte Fall storniert und ein neuer zusammengeführter Fall an die Krankenversicherung übermittelt (GKV-Spitzenverband, 2018). Somit beinhalten die Abrechnungsdaten der Krankenkassen nicht die stornierten Fälle, was zu einer Verzerrung in der Datengrundlage führt. Jedoch sind gerade hier die Fälle zu nennen, welche eine Wiederaufnahme nach Komplikationen beim ersten Fall darstellen können. Dabei könnten genau diese Fälle entscheidende Gründe für Wiederaufnahmen liefern bzw. auf präventive Maßnahmen hindeuten, um die Versorgung zu verbessern. Als Lösungs- bzw. Validierungsansatz könnten Krankenhausdaten mit Abrechnungsdaten der Krankenkassen verknüpft werden. Damit könnten für einzelne Krankenhäuser alle Krankenhausfälle ausgewertet werden (ohne die zuvor genannte Verzerrung der Fallzusammenführung). Gleichzeitig würde sich der Vorteil ergeben, dass alle anderen Fälle auf Basis der Krankenkassendaten zusätzlich erfasst wären.

Im Projekt PREMISE wurden nur die Daten einer einzelnen Krankenversicherung ausgewertet. Daraus ergibt sich die offene Frage, ob die Projektergebnisse auch auf andere Krankenkassen und damit andere Populationen übertragbar sind. Ganz unmittelbar betrifft dies beispielsweise Privatpatienten, die sich von der GKV-Population unterscheiden mögen (Hoffmann & Icks, 2012; Huber & Mielck, 2010). Ein Lösungsansatz wäre die Zusammenführung mehrerer Quellen von Routinedaten, wie es aktuell beim Forschungsdatenzentrum Gesundheit angestrebt wird. In PREMISE konnten dagegen bisher nur die Daten der regionalen Krankenversicherung AOK Baden-Württemberg analysiert werden, was regional einer großen Datenquelle aus den meisten Menschen in der Region entspricht und damit als weitgehend repräsentativ angesehen werden kann (AOK Baden-Württemberg (Hauptverwaltung), 2017).

Bei der Auswahl von Index- und Rehospitalisierungs-Code-Sets kann nicht ausgeschlossen werden, relevante ICD-Codes ausgeschlossen oder irrelevante Codes fälschlicherweise eingeschlossen zu haben. Dem wurde dadurch begegnet, dass mehrere Informationsquellen

aus Expertenmeinung und veröffentlichter Evidenz berücksichtigt wurden, z. B. in Validierungsstudien (Ando et al., 2018; Chen, Khan, Walker, & Quan, 2010). Außerdem konnte nicht zwischen geplanten, ungeplanten, vermeidbaren und nicht vermeidbaren Rehospitalisierungen unterschieden werden. Mit einer solchen Unterscheidung könnten präventive Maßnahmen weitaus zielgerichteter zugewiesen werden. Während die Literatur hierfür viele Konzepte zur Definition und Ermittlung von vermeidbaren (Re)Hospitalisierungen bereithält (Mongkhon, Ashcroft, Scholfield, & Kongkaew, 2018; van Walraven, Bennett, Jennings, Austin, & Forster, 2011), mag bezweifelt werden, ob dies allein an ICD-Codes erkennbar ist (Amelung et al., 2022). Vielmehr würde eine solche Einschätzung alle verfügbaren klinischen und lebenssituativen Informationen eines Patienten erfordern (van Walraven et al., 2011).

8. Verwendung der Ergebnisse nach Ende der Förderung

Unter den stetig steigenden Gesundheitsausgaben der letzten Jahre machen die Ausgaben im stationären Bereich fortwährend den größten Teil aus. Dies gilt national wie international, wo zahlreiche Initiativen die Wiederaufnahmen von Patienten in Krankenhäuser vermeiden sollen. Demgegenüber ist der Wissensstand zu Rehospitalisierungen in Deutschland noch sehr gering. Folglich fällt es schwer, zielgerichtete Handlungsalternativen und Maßnahmen zu entwickeln und zu implementieren. Hierfür hat PREMISE wichtige Grundlagen gelegt, denn selbst in GKV-Abrechnungsdaten gab es bisher nur vereinzelte Analysen (Stahmeyer, Stubenrauch, Geyer, Weissenborn, & Eberhard, 2019; Swanson, Vogt, Sundmacher, Hagen, & Moger, 2018). Die fehlenden Standards zur Klassifikation eines erneuten Aufenthaltes sind noch nicht etabliert, wofür das Vorgehen aus PREMISE verwendet und weiterentwickelt werden kann. Dies wäre äußerst wichtig für spezifische Rehospitalisierungen, doch über die Abfolge solcher Rehospitalisierungen ist wenig bekannt (etwa in welchem Zeitraum sie stattfinden). Die abrechnungstechnischen Besonderheiten in Routinedaten waren auch für PREMISE herausfordernd, wofür ein Regelwerk erstellt wurde, um dieser Herausforderung adäquat zu begegnen. Die auf dieser Grundlage entwickelten Modelle sind in ihrer Vorhersagekraft bereits zufriedenstellend, können aber durchaus weiter verbessert werden um Risikopatienten eindeutig erkennen zu können.

Prädiktionsmodelle wie in PREMISE (aus longitudinalen Kassendaten) sind zunehmend populär. Ein Grund hierfür mag sein, dass sie sich zur Überwachung einer medizinischen Behandlung eignen und auf dieser Grundlage auch interveniert werden kann, um klinische und gesundheitsökonomische Folgen zu reduzieren. Optimierungsmöglichkeiten ergeben sich auch in der kosten-effizienten Gestaltung des Gesundheitswesens, denn adäquate Arzneimittelverschreibungen tragen maßgeblich zur Wirtschaftlichkeit, Zweckmäßigkeit und Qualität der GKV-Versorgung bei. Inadäquate Verschreibungen dagegen gefährden die Patientensicherheit und das Erreichen medizinischer (patientenorientierter) Outcomes in den Alltagsbedingungen der Versorgung. Somit lassen sich durch deren Vermeidung mit großer Wahrscheinlichkeit unerwünschte Folgen vermindern und Versorgungsabläufe und Outcomes beträchtlich verbessern. Der Einfluss von Fehl-, Unter- und Überversorgung auf Rehospitalisierungen wurde in PREMISE mittels STOPP-/ START-Kriterien untersucht, eine international anerkannte und in Sekundärdaten erfolgreich anwendbare Kriterienliste (de Groot et al., 2014; Meid et al., 2018) mit nachgewiesenem prädiktiven Wert für die Vorhersage von Hospitalisierungen (Brown et al., 2016; Meid et al., 2018). Die aus PREMISE erhaltenen Erkenntnisse können um weitere arzneimittelbedingte Risikofaktoren für Rehospitalisierungen erweitert werden, etwa um darauf präventive Maßnahmen aufzubauen und bisher bekannte Risikofaktoren zu bestätigen (Rosen, Fridman, Rosen, Shane, & Pevnick, 2017).

Die Prädiktionsmodelle von PREMISE können bei der Erkennung von Hochrisikopopulationen helfen, die von einer mitunter aufwendigen Intervention tatsächlich profitieren können

(Hesselink et al., 2012). Unter solchen finden sich vor der Entlassung beispielsweise eine intensive pharmazeutische Betreuung (Al-Rashed, Wright, Roebuck, Sunter, & Chrystyn, 2002), elektronischen Entscheidungshilfen bis hin zur web-basierten Informationsübertragung an weiterbehandelnde niedergelassene Ärzte (Hesselink et al., 2012). Ambulante Nachverfolgungsprogramme (Cavanaugh, Lindsey, Shilliday, & Ratner, 2015) im Speziellen können von telefonischer Betreuung (Jack et al., 2009) bis hin zum Hausbesuch reichen (Hansen, Young, Hinami, Leung, & Williams, 2011). In künftigen Anwendungen könnten die aus Routinedaten identifizierten Risikofaktoren (automatisiert entdeckt) dabei helfen, Ressourcen adäquat zuzuteilen und die passende Intervention auszuwählen, um letztlich Anzahl und Schwere von Komplikationen zu reduzieren, die zu Rehospitalisierungen führen.

Für die Weiterentwicklung bestehender Versorgungskonzepte wäre es vielversprechend, wenn ambulante Routinedaten mit Informationen aus übermittelten stationären Routinedaten verknüpft würden. Auch könnten gänzlich neue Programme zur koordinierten Betreuung zwischen dem Leistungserbringer Krankenhaus und dem Leistungsträger Krankenkasse entwickelt werden. Folglich müssten Interventionen sich nicht nur auf das Setting vor oder nach der Entlassung (Pre-/Post-Discharge (Jenq, Doyle, Belton, Herrin, & Horwitz, 2016; Kripalani, Theobald, Anctil, & Vasilevskis, 2014)) beziehen, sondern es kämen auch sektoren-übergreifende Ansätze in Betracht (Hesselink et al., 2012). Je nach Intervention kann eine Priorisierung nach dem Rehospitalisierungsrisiko bei Entlassung vorgenommen werden, was in der Fläche funktionieren und beträchtliche Effekte erzielen kann (Halfon et al., 2006). Fundamental wichtig ist es, Risikopatienten für Rehospitalisierungen zum Zeitpunkt ihrer Entlassung zuverlässig zu erkennen und individuell geeignete Präventionsmaßnahmen einzuleiten (Casas et al., 2006; Takeda et al., 2019). Modelle aus PREMISE müssten dem entlassenden Krankenhaus zur Verfügung stehen, wie auch die im Krankenhaus generierten Daten zu Diagnosen und Prozeduren und ambulante Patientendaten einschließlich Diagnosen und Arzneimittelverordnungen. All dies muss einrichtungsübergreifend erfolgen, was bei derzeitigen Rahmenbedingungen zur Nutzung von Sozialdaten (nur zweckgebunden und für definierte Zeiträume nach Bewilligung eines entsprechenden Antrages) noch nicht möglich erscheint (Bundesamt für Justiz, 1980). Außerdem geschieht die Datenübermittlung aktuell nicht in Echtzeit, was die Nutzung der Modelle noch nicht möglich macht. Es ist daher eine Verzahnung der einzelnen Akteure im Gesundheitswesen nötig, was am effizientesten mit der elektronischen Gesundheitskarte (eGK) denkbar wäre, aber auch hier muss erst der rechtliche Rahmen geschaffen werden.

Nachdem Risikopatienten erkannt wurden, könnten individuelle Präventionsmaßnahmen ergriffen werden. Dazu könnte eine gezielte Analyse der aktuellen Pharmakotherapie (z. B. als Prüfung vor Entlassung), entscheidend beitragen. Beeinflussbare Risikofaktoren der Unter-, Fehl- oder Überversorgung könnten mit dieser Methode eliminiert (Angel et al., 2019; Komagamine, 2018) und therapeutische Alternativen mit geringerem Risiko oder Modifikationen des Therapieregimes (z. B. Dosisanpassung) vorgeschlagen werden. Hierfür kann auch ein elektronisches Entscheidungsunterstützungssystem verwendet werden (Rogerio-Blanco et al., 2020). Auch die Entlassbriefschreibung könnte entsprechende Risikofaktoren berücksichtigen und Anpassungen vorschlagen, etwa in Stile einer intensiven pharmazeutischen Betreuung vor der Entlassung (Al-Rashed et al., 2002) oder über gezieltere Hinweise für die nachbehandelnden Heilberufler (Morath, Wien, Hoppe-Tichy, Haefeli, & Seidling, 2019). Dabei ist es effektiver die Präventionsmaßnahmen bereits vor Entlassung zu beginnen und danach weiterzuführen als erst nach der Entlassung zu beginnen (Braet, Weltens, & Sermeus, 2016). Ambulante Nachverfolgungsprogramme sind also vielversprechende, sich anschließende Maßnahmen (Cavanaugh et al., 2015). Hier können auch die telefonische Betreuung oder Hausbesuche angebracht sein (Hansen et al., 2011; Jack et al., 2009; Wu et al., 2006). Darüber hinaus lassen sich ihre aggregierten Rehospitalisierungsrisiken auch als fortlaufender Qualitätsindikator nutzen (Fischer et al., 2014).

9. Erfolgte bzw. geplante Veröffentlichungen

Erfolgte Veröffentlichungen

Ruff C, Gerharz A, Groll A, Stoll F, Wirbka L, Haefeli WE, Meid AD. Disease-dependent variations in the timing and causes of readmissions in Germany: A claims data analysis for six different conditions. [PLoS One 2021;16:e0250298](#).

Gerharz A*, Ruff C*, Wirbka L, Stoll F, Haefeli WE, Groll A, Meid AD. Predicting hospital readmissions from health insurance claims data: a modeling study targeting potentially inappropriate prescribing. *Methods Inf Med* 2022;61:55-60. (*equal contribution)

Wirbka L, Ruff C, Haefeli WE, Meid AD. A prediction model for nonpersistence or nonadherence to direct oral anticoagulants in hospitalized patients with atrial fibrillation. [J Manag Care Spec Pharm 2022;28:1161-1172](#).

Abstracts und Kongressbeiträge

Meid AD, Wirbka L, Ruff C, Haefeli WE. Differential effects of beta-blocker exposures in predicting specific readmissions for heart failure and myocardial infarction. Annual conference of Gesellschaft für Arzneimittelwandlungsforschung und Arzneimittel-epidemiologie (GAA) 2021; [28th conference: abstract 09](#)

Wirbka L, Ruff C, Haefeli WE, Meid AD. The role of adherence to multiple guideline-directed medicines in predicting specific readmissions for heart failure and myocardial infarction. Annual conference of Gesellschaft für Arzneimittelwandlungsforschung und Arzneimittel-epidemiologie (GAA) 2021; [28th conference: abstract 17](#)

Gerharz A, Ruff C, Groll A, Meid AD. Model Selection for Predicting Readmissions with Health Insurance Data. [36th International Workshop on Statistical Modelling \(IWSM\), Trieste, Italy, 18–22 July 2022](#).

Geplante Veröffentlichungen

Levinson R, Paul C, Meid AD, Schultz J, Wild B. Using a machine learning approach to identify predictors of readmission in heart failure patients from a statutory health insurance database. *(eingereicht bei Journal of Medical Internet Research)*

Gerharz A, Ruff C, Haefeli WE, Groll A, Meid AD. Comparison of different machine learning approaches for predicting readmissions. *(Einreichung geplant)*

10. Literaturverzeichnis

Al-Rashed, S. A., Wright, D. J., Roebuck, N., Sunter, W., & Chrystyn, H. (2002). The value of inpatient pharmaceutical counselling to elderly patients prior to discharge. *Br J Clin Pharmacol*, 54(6), 657-664. doi:10.1046/j.1365-2125.2002.01707.x

Allen, N. B., Holford, T. R., Bracken, M. B., Goldstein, L. B., Howard, G., Wang, Y., & Lichtman, J. H. (2010). Geographic variation in one-year recurrent ischemic stroke rates for elderly Medicare beneficiaries in the USA. *Neuroepidemiology*, 34(2), 123-129. doi:10.1159/000274804

Amarasingham, R., Moore, B. J., Tabak, Y. P., Drazner, M. H., Clark, C. A., Zhang, S., . . . Halm, E. A. (2010). An automated model to identify heart failure patients at risk for 30-day

- readmission or death using electronic medical record data. *Med Care*, 48(11), 981-988. doi:10.1097/MLR.0b013e3181ef60d9
- Amelung, S., Czock, D., Thalheimer, M., Hoppe-Tichy, T., Haefeli, W. E., & Seidling, H. M. (2022). Shortcomings of administrative data to derive preventive strategies for in-hospital drug-induced acute kidney failure - insights from patient record analysis. *J Clin Med*, 11(15). doi:10.3390/jcm11154285
- Ando, T., Ooba, N., Mochizuki, M., Koide, D., Kimura, K., Lee, S. L., . . . Kubota, K. (2018). Positive predictive value of ICD-10 codes for acute myocardial infarction in Japan: a validation study at a single center. *BMC Health Serv Res*, 18(1), 895. doi:10.1186/s12913-018-3727-0
- Angel, Y., Zeltser, D., Berliner, S., Ingbir, M., Shapira, I., Shenhar-Tsarfaty, S., & Rogowski, O. (2019). Hospitalization as an opportunity to correct errors in anticoagulant treatment in patients with atrial fibrillation. *Br J Clin Pharmacol*, 85(12), 2838-2847. doi:10.1111/bcp.14116
- Angermann, C. E. (2009). Comorbidities in heart failure: a key issue. *European Journal of Heart Failure Supplements*, 8, i5-i10.
- AOK Baden-Württemberg (Hauptverwaltung). (2017). DAS GEHT. Unternehmensbericht AOK Baden-Württemberg Ausgabe 2017. Retrieved from https://aok-bw-presse.de/fileadmin/mediathek/dokumente/unternehmensbericht_aok_bw_2017.pdf
- Artetxe, A., Beristain, A., & Grana, M. (2018). Predictive models for hospital readmission risk: A systematic review of methods. *Comput Methods Programs Biomed*, 164, 49-64. doi:10.1016/j.cmpb.2018.06.006
- Billings, J., & Mijanovich, T. (2007). Improving the management of care for high-cost Medicaid patients. *Health Aff (Millwood)*, 26(6), 1643-1654. doi:10.1377/hlthaff.26.6.1643
- Black, D. M., Geiger, E. J., Eastell, R., Vittinghoff, E., Li, B. H., Ryan, D. S., . . . Adams, A. L. (2020). Atypical Femur Fracture Risk versus Fragility Fracture Prevention with Bisphosphonates. *N Engl J Med*, 383(8), 743-753. doi:10.1056/NEJMoa1916525
- Bottle, A., Honeyford, K., Chowdhury, F., Bell, D., & Aylin, P. (2018). In *Factors associated with hospital emergency readmission and mortality rates in patients with heart failure or chronic obstructive pulmonary disease: a national observational study*. Southampton (UK).
- Braet, A., Weltens, C., & Sermeus, W. (2016). Effectiveness of discharge interventions from hospital to home on hospital readmissions: a systematic review. *JBIS Database System Rev Implement Rep*, 14(2), 106-173. doi:10.11124/jbisrir-2016-2381
- Braunwald, E. (2019). Diabetes, heart failure, and renal dysfunction: The vicious circles. *Prog Cardiovasc Dis*, 62(4), 298-302. doi:10.1016/j.pcad.2019.07.003
- Brown, J. D., Hutchison, L. C., Li, C., Painter, J. T., & Martin, B. C. (2016). Predictive Validity of the Beers and Screening Tool of Older Persons' Potentially Inappropriate Prescriptions (STOPP) Criteria to Detect Adverse Drug Events, Hospitalizations, and Emergency Department Visits in the United States. *J Am Geriatr Soc*, 64(1), 22-30. doi:10.1111/jgs.13884

- Brunetti, E., Aurucci, M. L., Boietti, E., Gibello, M., Sappa, M., Falcone, Y., . . . Bo, M. (2019). Clinical Implications of Potentially Inappropriate Prescribing According to STOPP/START Version 2 Criteria in Older Polymorbid Patients Discharged From Geriatric and Internal Medicine Wards: A Prospective Observational Multicenter Study. *J Am Med Dir Assoc*, 20(11), 1476 e1471-1476 e1410. doi:10.1016/j.jamda.2019.03.023
- Bundesamt für Justiz. (1980). Zehntes Buch Sozialgesetzbuch - Sozialverwaltungsverfahren und Sozialdatenschutz - (SGB X). Retrieved from https://www.gesetze-im-internet.de/sgb_10/index.html
- Bundesärztekammer, K. B., Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften (2020),. (2020). Nationale Versorgungsleitlinie COPD (AWMF-Register-Nr.: nvl-003). Retrieved from <https://www.leitlinien.de/themen/copd>
- Bundesministerium für Gesundheit und Soziale Sicherung (Referat 216). (2004). Leitsätze zur Anwendung der Wiederaufnahmeregelung nach § 2 KFPV 2004. Retrieved from <https://nkgev.info/abrechnung.html>
- Casas, A., Troosters, T., Garcia-Aymerich, J., Roca, J., Hernandez, C., Alonso, A., . . . members of the, C. P. (2006). Integrated care prevents hospitalisations for exacerbations in COPD patients. *Eur Respir J*, 28(1), 123-130. doi:10.1183/09031936.06.00063205
- Cavanaugh, J. J., Lindsey, K. N., Shilliday, B. B., & Ratner, S. P. (2015). Pharmacist-coordinated multidisciplinary hospital follow-up visits improve patient outcomes. *J Manag Care Spec Pharm*, 21(3), 256-260. doi:10.18553/jmcp.2015.21.3.256
- Chen, G., Khan, N., Walker, R., & Quan, H. (2010). Validating ICD coding algorithms for diabetes mellitus from administrative data. *Diabetes Res Clin Pract*, 89(2), 189-195. doi:10.1016/j.diabres.2010.03.007
- Cruden, N. L., Din, J. N., Janssen, C., Smith, R., Hilton, J. D., Klinke, W. P., . . . Siega, A. D. (2014). Delay in filling first clopidogrel prescription after coronary stenting is associated with an increased risk of death and myocardial infarction. *J Am Heart Assoc*, 3(3), e000669. doi:10.1161/JAHA.113.000669
- Cubbon, R. M., Woolston, A., Adams, B., Gale, C. P., Gilthorpe, M. S., Baxter, P. D., . . . Kearney, M. T. (2014). Prospective development and validation of a model to predict heart failure hospitalisation. *Heart*, 100(12), 923-929. doi:10.1136/heartjnl-2013-305294
- de Groot, D. A., de Vries, M., Joling, K. J., van Campen, J. P., Hugtenburg, J. G., van Marum, R. J., . . . van Hout, H. P. (2014). Specifying ICD9, ICPC and ATC codes for the STOPP/START criteria: a multidisciplinary consensus panel. *Age Ageing*, 43(6), 773-778. doi:10.1093/ageing/afu075
- Debray, T. P., Koffijberg, H., Nieboer, D., Vergouwe, Y., Steyerberg, E. W., & Moons, K. G. (2014). Meta-analysis and aggregation of multiple published prediction models. *Stat Med*, 33(14), 2341-2362. doi:10.1002/sim.6080
- DESTATIS Statistisches Bundesamt. (2020a). Kosten der Krankenhäuser nach Bundesländern im Jahr 2018. Retrieved from <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Krankenhaeuser/Tabellen/kosten-krankenhaeuser-bl.html>
- DESTATIS Statistisches Bundesamt. (2020b). Krankenhäuser – Einrichtungen, Betten und Patientenbewegungen. Retrieved from

<https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Krankenhaeuser/Tabellen/gd-krankenhaeuser-jahre.html>

- Deutsche Gesellschaft für Allgemeinmedizin und Familienmedizin (DEGAM). (2017a). Hausärztliche Risikoberatung zur kardiovaskulären Prävention. S3-Leitlinie. Retrieved from https://www.awmf.org/uploads/tx_szleitlinien/053-024l_S3_Hausaerztliche_Risikoberat_kardiovask_Praevention_2018-09.pdf
- Deutsche Gesellschaft für Allgemeinmedizin und Familienmedizin (DEGAM). (2017b). Multimorbidität. S3-Leitlinie. Retrieved from https://www.degam.de/files/Inhalte/Leitlinien-Inhalte/Dokumente/DEGAM-S3-Leitlinien/053-047_Multimorbiditaet/053-047l_%20Multimorbiditaet_redakt_24-1-18.pdf
- Dharmarajan, K., Hsieh, A. F., Lin, Z., Bueno, H., Ross, J. S., Horwitz, L. I., . . . Krumholz, H. M. (2013). Diagnoses and timing of 30-day readmissions after hospitalization for heart failure, acute myocardial infarction, or pneumonia. *JAMA*, *309*(4), 355-363. doi:10.1001/jama.2012.216476
- Dodson, J. A., Hajduk, A. M., Murphy, T. E., Geda, M., Krumholz, H. M., Tsang, S., . . . Chaudhry, S. I. (2019). Thirty-day readmission risk model for older adults hospitalized with acute myocardial infarction. *Circ Cardiovasc Qual Outcomes*, *12*(5), e005320. doi:10.1161/CIRCOUTCOMES.118.005320
- Donze, J. D., Williams, M. V., Robinson, E. J., Zimlichman, E., Aujesky, D., Vasilevskis, E. E., . . . Schnipper, J. L. (2016). International validity of the HOSPITAL score to predict 30-day potentially avoidable hospital readmissions. *JAMA Intern Med*, *176*(4), 496-502. doi:10.1001/jamainternmed.2015.8462
- Dormann, H., Neubert, A., Criegee-Rieck, M., Egger, T., Radespiel-Troger, M., Azaz-Livshits, T., . . . Hahn, E. G. (2004). Readmissions and adverse drug reactions in internal medicine: the economic impact. *J Intern Med*, *255*(6), 653-663. doi:10.1111/j.1365-2796.2004.01326.x
- Dorning, H., Bitzer, E. M., Hofmann, W., & Schwartz, F. W. (1995). [Longitudinal analysis of utilization of inpatient treatment based on insurance data of mandatory health insurance]. *Gesundheitswesen*, *57*(10), 631-637.
- Doshi, S. M., & Friedman, A. N. (2017). Diagnosis and Management of Type 2 Diabetic Kidney Disease. *Clin J Am Soc Nephrol*, *12*(8), 1366-1373. doi:10.2215/CJN.11111016
- Dumbreck, S., Flynn, A., Nairn, M., Wilson, M., Treweek, S., Mercer, S. W., . . . Guthrie, B. (2015). Drug-disease and drug-drug interactions: systematic examination of recommendations in 12 UK national clinical guidelines. *Bmj*, *350*, h949. doi:10.1136/bmj.h949
- Elixhauser, A., Steiner, C., Harris, D. R., & Coffey, R. M. (1998). Comorbidity measures for use with administrative data. *Med Care*, *36*(1), 8-27. doi:10.1097/00005650-199801000-00004
- Ena, J., Gomez-Huelgas, R., Gracia-Tello, B. C., Vazquez-Rodriguez, P., Alcalá-Pedrajas, J. N., Carrasco-Sanchez, F. J., . . . Grupo de Diabetes, O. y. N. d. I. S. E. d. M. I. (2018). Derivation and validation of a predictive model for the readmission of patients with

- diabetes mellitus treated in internal medicine departments. *Rev Clin Esp (Barc)*, 218(6), 271-278. doi:10.1016/j.rce.2018.03.010
- Enomoto, L. M., Shrestha, D. P., Rosenthal, M. B., Hollenbeak, C. S., & Gabbay, R. A. (2017). Risk factors associated with 30-day readmission and length of stay in patients with type 2 diabetes. *J Diabetes Complications*, 31(1), 122-127. doi:10.1016/j.jdiacomp.2016.10.021
- Fischer, C., Lingsma, H. F., Marang-van de Mheen, P. J., Kringos, D. S., Klazinga, N. S., & Steyerberg, E. W. (2014). Is the readmission rate a valid quality indicator? A review of the evidence. *PLoS One*, 9(11), e112282. doi:10.1371/journal.pone.0112282
- Formiga, F., Masip, J., Chivite, D., & Corbella, X. (2017). Applicability of the heart failure Readmission Risk score: A first European study. *Int J Cardiol*, 236, 304-309. doi:10.1016/j.ijcard.2017.02.024
- Freund, T., Campbell, S. M., Geissler, S., Kunz, C. U., Mahler, C., Peters-Klimm, F., & Szecsenyi, J. (2013). Strategies for reducing potentially avoidable hospitalizations for ambulatory care-sensitive conditions. *Ann Fam Med*, 11(4), 363-370. doi:10.1370/afm.1498
- Garcia-Aymerich, J., Farrero, E., Felez, M. A., Izquierdo, J., Marrades, R. M., Anto, J. M., & Estudi del Factors de Risc d'Aguditzacio de la, M. i. (2003). Risk factors of readmission to hospital for a COPD exacerbation: a prospective study. *Thorax*, 58(2), 100-105. doi:10.1136/thorax.58.2.100
- Gerharz, A., Ruff, C., Wirbka, L., Stoll, F., Haefeli, W. E., Groll, A., & Meid, A. D. (2022). Predicting hospital readmissions from health insurance claims data: A modeling study targeting potentially inappropriate prescribing. *Methods Inf Med*, 61(1-02), 55-60. doi:10.1055/s-0042-1742671
- GKV-Spitzenverband. (2018). Elektronischer Datenaustausch in der gesetzlichen Krankenversicherung. Anlage 5 zur § 301-Vereinbarung - Durchführungshinweise - Version 66. Retrieved from https://www.gkv-datenaustausch.de/leistungserbringer/krankenhaeuser/technische_anlagen_kh_archiv.jsp
- GKV-Spitzenverband, Verband der Privaten Krankenversicherung, & Deutsche Krankenhausgesellschaft. (2018). Vereinbarung zum Fallpauschalensystem für Krankenhäuser für das Jahr 2019 (Fallpauschalenvereinbarung 2019 – FPV 2019). Retrieved from https://www.g-drg.de/G-DRG-System_2019/Abrechnungsbestimmungen/FPV_2019
- Gudmundsson, G., Gislason, T., Janson, C., Lindberg, E., Hallin, R., Ulrik, C. S., . . . Bakke, P. (2005). Risk factors for rehospitalisation in COPD: role of health status, anxiety and depression. *Eur Respir J*, 26(3), 414-419. doi:10.1183/09031936.05.00078504
- Halfon, P., Egli, Y., Pretre-Rohrbach, I., Meylan, D., Marazzi, A., & Burnand, B. (2006). Validation of the potentially avoidable hospital readmission rate as a routine indicator of the quality of hospital care. *Med Care*, 44(11), 972-981. doi:10.1097/01.mlr.0000228002.43688.c2
- Hansen, L. O., Young, R. S., Hinami, K., Leung, A., & Williams, M. V. (2011). Interventions to reduce 30-day rehospitalization: a systematic review. *Ann Intern Med*, 155(8), 520-528. doi:10.7326/0003-4819-155-8-201110180-00008

- Hebert, C., Shivade, C., Foraker, R., Wasserman, J., Roth, C., Mekhjian, H., . . . Embi, P. (2014). Diagnosis-specific readmission risk prediction using electronic health data: a retrospective cohort study. *BMC Med Inform Decis Mak*, *14*, 65. doi:10.1186/1472-6947-14-65
- Hess, P. L., Mirro, M. J., Diener, H. C., Eikelboom, J. W., Al-Khatib, S. M., Hylek, E. M., . . . Atrial Fibrillation Think-Tank, P. (2014). Addressing barriers to optimal oral anticoagulation use and persistence among patients with atrial fibrillation: Proceedings, Washington, DC, December 3-4, 2012. *Am Heart J*, *168*(3), 239-247 e231. doi:10.1016/j.ahj.2014.04.007
- Hesselink, G., Schoonhoven, L., Barach, P., Spijker, A., Gademan, P., Kalkman, C., . . . Wollersheim, H. (2012). Improving patient handovers from hospital to primary care: a systematic review. *Ann Intern Med*, *157*(6), 417-428. doi:10.7326/0003-4819-157-6-201209180-00006
- Ho, P. M., Magid, D. J., Shetterly, S. M., Olson, K. L., Maddox, T. M., Peterson, P. N., . . . Rumsfeld, J. S. (2008). Medication nonadherence is associated with a broad range of adverse outcomes in patients with coronary artery disease. *Am Heart J*, *155*(4), 772-779. doi:10.1016/j.ahj.2007.12.011
- Hoffmann, F., & Icks, A. (2012). [Structural differences between health insurance funds and their impact on health services research: results from the Bertelsmann Health-Care Monitor]. *Gesundheitswesen*, *74*(5), 291-297. doi:10.1055/s-0031-1275711
- Huber, J., & Mielck, A. (2010). [Morbidity and healthcare differences between insured in the statutory ("GKV") and private health insurance ("PKV") in Germany. Review of empirical studies]. *Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz*, *53*(9), 925-938. doi:10.1007/s00103-010-1119-7
- Hyttinen, V., Jyrkka, J., & Valtonen, H. (2016). A systematic review of the impact of potentially inappropriate medication on health care utilization and costs among older adults. *Med Care*, *54*(10), 950-964. doi:10.1097/MLR.0000000000000587
- Institut für das Entgeltsystem im Krankenhaus GmbH (InEK). (2019). DEUTSCHE KODIERRICHTLINIEN, Allgemeine und spezielle Kodierrichtlinien für die Verschlüsselung von Krankheiten und Prozeduren (Version 2019).
- Jack, B. W., Chetty, V. K., Anthony, D., Greenwald, J. L., Sanchez, G. M., Johnson, A. E., . . . Culpepper, L. (2009). A reengineered hospital discharge program to decrease rehospitalization: a randomized trial. *Ann Intern Med*, *150*(3), 178-187. doi:10.7326/0003-4819-150-3-200902030-00007
- Jencks, S. F., Williams, M. V., & Coleman, E. A. (2009). Rehospitalizations among patients in the Medicare fee-for-service program. *N Engl J Med*, *360*(14), 1418-1428. doi:10.1056/NEJMs0803563
- Jenq, G. Y., Doyle, M. M., Belton, B. M., Herrin, J., & Horwitz, L. I. (2016). Quasi-experimental evaluation of the effectiveness of a large-scale readmission reduction program. *JAMA Intern Med*, *176*(5), 681-690. doi:10.1001/jamainternmed.2016.0833
- Karunakaran, A., Zhao, H., & Rubin, D. J. (2018). PredischARGE and postdischarge risk factors for hospital readmission among patients with diabetes. *Med Care*, *56*(7), 634-642. doi:10.1097/MLR.0000000000000931

- Kitamura, M., Izawa, K. P., Taniue, H., Mimura, Y., Imamura, K., Nagashima, H., & Brubaker, P. H. (2017). Relationship between activities of daily living and readmission within 90 days in hospitalized elderly patients with heart failure. *Biomed Res Int*, 2017, 7420738. doi:10.1155/2017/7420738
- Komagamine, J. (2018). Prevalence of potentially inappropriate medications at admission and discharge among hospitalised elderly patients with acute medical illness at a single centre in Japan: a retrospective cross-sectional study. *BMJ Open*, 8(7), e021152. doi:10.1136/bmjopen-2017-021152
- Kostev, K., & Jacob, L. (2018). Multimorbidity and polypharmacy among elderly people followed in general practices in Germany. *Eur J Intern Med*, 55, 66-68. doi:10.1016/j.ejim.2018.07.014
- Kripalani, S., Theobald, C. N., Anctil, B., & Vasilevskis, E. E. (2014). Reducing hospital readmission rates: current strategies and future directions. *Annu Rev Med*, 65, 471-485. doi:10.1146/annurev-med-022613-090415
- Krumholz, H. M., Chen, Y. T., Wang, Y., Vaccarino, V., Radford, M. J., & Horwitz, R. I. (2000). Predictors of readmission among elderly survivors of admission with heart failure. *Am Heart J*, 139(1 Pt 1), 72-77. doi:10.1016/s0002-8703(00)90311-9
- Krumholz, H. M., Lin, Z., Drye, E. E., Desai, M. M., Han, L. F., Rapp, M. T., . . . Normand, S. L. (2011). An administrative claims measure suitable for profiling hospital performance based on 30-day all-cause readmission rates among patients with acute myocardial infarction. *Circ Cardiovasc Qual Outcomes*, 4(2), 243-252. doi:10.1161/CIRCOUTCOMES.110.957498
- Lau, A. C., Yam, L. Y., & Poon, E. (2001). Hospital re-admission in patients with acute exacerbation of chronic obstructive pulmonary disease. *Respir Med*, 95(11), 876-884. doi:10.1053/rmed.2001.1180
- Lee, J. D., Lee, T. H., Huang, Y. C., Lee, M., Kuo, Y. W., Huang, Y. C., & Hu, Y. H. (2019). Prediction model of early return to hospital after discharge following acute ischemic stroke. *Curr Neurovasc Res*, 16(4), 348-357. doi:10.2174/1567202616666190911125951
- Lewis, G. H. (2010). "Impactability models": identifying the subgroup of high-risk patients most amenable to hospital-avoidance programs. *Milbank Q*, 88(2), 240-255. doi:10.1111/j.1468-0009.2010.00597.x
- Martin, G. P., Mamas, M. A., Peek, N., Buchan, I., & Sperrin, M. (2017). Clinical prediction in defined populations: a simulation study investigating when and how to aggregate existing models. *BMC Med Res Methodol*, 17(1), 1. doi:10.1186/s12874-016-0277-1
- Mazzaglia, G., Roti, L., Corsini, G., Colombini, A., Maciocco, G., Marchionni, N., . . . Di Bari, M. (2007). Screening of older community-dwelling people at risk for death and hospitalization: the Assistenza Socio-Sanitaria in Italia project. *J Am Geriatr Soc*, 55(12), 1955-1960. doi:10.1111/j.1532-5415.2007.01446.x
- McCoy, R. G., Herrin, J., Lipska, K. J., & Shah, N. D. (2018). Recurrent hospitalizations for severe hypoglycemia and hyperglycemia among U.S. adults with diabetes. *J Diabetes Complications*, 32(7), 693-701. doi:10.1016/j.jdiacomp.2018.04.007

- Meid, A. D., Groll, A., Heider, D., Mächler, S., Adler, J. B., Günster, C., . . . Haefeli, W. E. (2018). Prediction of Drug-Related Risks Using Clinical Context Information in Longitudinal Claims Data. *Value Health, 21*(12), 1390-1398. doi:10.1016/j.jval.2018.05.007
- Meid, A. D., Groll, A., Schieborr, U., Walker, J., & Haefeli, W. E. (2017). How can we define and analyse drug exposure more precisely to improve the prediction of hospitalizations in longitudinal (claims) data? *Eur J Clin Pharmacol, 73*(3), 373-380. doi:10.1007/s00228-016-2184-0
- Meid, A. D., & Haefeli, W. E. (2016). Age-dependent impact of medication underuse and strategies for improvement. *Gerontology, 62*(5), 491-499. doi:10.1159/000443477
- Mongkhon, P., Ashcroft, D. M., Scholfield, C. N., & Kongkaew, C. (2018). Hospital admissions associated with medication non-adherence: a systematic review of prospective observational studies. *BMJ Qual Saf, 27*(11), 902-914. doi:10.1136/bmjqs-2017-007453
- Morath, B., Wien, K., Hoppe-Tichy, T., Haefeli, W. E., & Seidling, H. M. (2019). Structure and content of drug monitoring advices included in discharge letters at interfaces of care: exploratory analysis preceding database development. *JMIR Med Inform, 7*(2), e10832. doi:10.2196/10832
- O'Mahony, D., O'Sullivan, D., Byrne, S., O'Connor, M. N., Ryan, C., & Gallagher, P. (2015). STOPP/START criteria for potentially inappropriate prescribing in older people: version 2. *Age Ageing, 44*(2), 213-218. doi:10.1093/ageing/afu145
- Ostling, S., Wyckoff, J., Ciarkowski, S. L., Pai, C. W., Choe, H. M., Bahl, V., & Gianchandani, R. (2017). The relationship between diabetes mellitus and 30-day readmission rates. *Clin Diabetes Endocrinol, 3*, 3. doi:10.1186/s40842-016-0040-x
- Png, M. E., Yoong, J., Chen, C., Tan, C. S., Tai, E. S., Khoo, E. Y. H., & Wee, H. L. (2018). Risk factors and direct medical cost of early versus late unplanned readmissions among diabetes patients at a tertiary hospital in Singapore. *Curr Med Res Opin, 34*(6), 1071-1080. doi:10.1080/03007995.2018.1431617
- Puth, M. T., Weckbecker, K., Schmid, M., & Munster, E. (2017). Prevalence of multimorbidity in Germany: impact of age and educational level in a cross-sectional study on 19,294 adults. *BMC Public Health, 17*(1), 826. doi:10.1186/s12889-017-4833-3
- Quan, H., Sundararajan, V., Halfon, P., Fong, A., Burnand, B., Luthi, J. C., . . . Ghali, W. A. (2005). Coding algorithms for defining comorbidities in ICD-9-CM and ICD-10 administrative data. *Med Care, 43*(11), 1130-1139. doi:10.1097/01.mlr.0000182534.19832.83
- Rana, S., Tran, T., Luo, W., Phung, D., Kennedy, R. L., & Venkatesh, S. (2014). Predicting unplanned readmission after myocardial infarction from routinely collected administrative hospital data. *Aust Health Rev, 38*(4), 377-382. doi:10.1071/AH14059
- Robbins, J. M., & Webb, D. A. (2006). Diagnosing diabetes and preventing rehospitalizations: the urban diabetes study. *Med Care, 44*(3), 292-296. doi:10.1097/01.mlr.0000199639.20342.87
- Rogero-Blanco, E., Lopez-Rodriguez, J. A., Sanz-Cuesta, T., Aza-Pascual-Salcedo, M., Bujalance-Zafra, M. J., Cura-Gonzalez, I., & Multi, P. A. P. G. (2020). Use of an electronic clinical decision support system in primary care to assess inappropriate polypharmacy in

- young seniors with multimorbidity: observational, descriptive, cross-sectional study. *JMIR Med Inform*, 8(2), e14130. doi:10.2196/14130
- Rosen, O. Z., Fridman, R., Rosen, B. T., Shane, R., & Pevnick, J. M. (2017). Medication adherence as a predictor of 30-day hospital readmissions. *Patient Prefer Adherence*, 11, 801-810. doi:10.2147/PPA.S125672
- Ruff, C., Gerharz, A., Groll, A., Stoll, F., Wirbka, L., Haefeli, W. E., & Meid, A. D. (2021). Disease-dependent variations in the timing and causes of readmissions in Germany: A claims data analysis for six different conditions. *PLoS One*, 16(4), e0250298. doi:10.1371/journal.pone.0250298
- Ruff, C., Koukalova, L., Haefeli, W.E., Meid, A.D (2019). The Role of Adherence Thresholds for Development and Performance Aspects of a Prediction Model for Direct Oral Anticoagulation Adherence. *Front Pharmacol*, 10, 113. doi: 10.3389/fphar.2019.00113.
- Sawhney, S., Marks, A., Fluck, N., McLernon, D. J., Prescott, G. J., & Black, C. (2017). Acute kidney injury as an independent risk factor for unplanned 90-day hospital readmissions. *BMC Nephrol*, 18(1), 9. doi:10.1186/s12882-016-0430-4
- Serra-Picamal, X., Roman, R., Escarrabill, J., Garcia-Altes, A., Argimon, J. M., Soler, N., . . . Agusti, A. (2018). Hospitalizations due to exacerbations of COPD: A big data perspective. *Respir Med*, 145, 219-225. doi:10.1016/j.rmed.2018.01.008
- Shadmi, E., Flaks-Manov, N., Hoshen, M., Goldman, O., Bitterman, H., & Balicer, R. D. (2015). Predicting 30-day readmissions with preadmission electronic health record data. *Med Care*, 53(3), 283-289. doi:10.1097/MLR.0000000000000315
- Smith, R. W., Kuluski, K., Costa, A. P., Sinha, S. K., Glazier, R. H., Forster, A., & Jeffs, L. (2017). Investigating the effect of sociodemographic factors on 30-day hospital readmission among medical patients in Toronto, Canada: a prospective cohort study. *BMJ Open*, 7(12), e017956. doi:10.1136/bmjopen-2017-017956
- Stahmeyer, J. T., Stubenrauch, S., Geyer, S., Weissenborn, K., & Eberhard, S. (2019). The frequency and timing of recurrent stroke: an analysis of routine health insurance data. *Dtsch Arztebl Int*, 116(42), 711-717. doi:10.3238/arztebl.2019.0711
- Statistisches Bundesamt. (2006). Krankheitskosten nach Alter und Geschlecht Kapitel 5.3.2 [Gesundheit in Deutschland, 2006]. Retrieved from https://www.gbe-bund.de/gbe/abrechnung.prc_abr_test_logon?p_uid=gast&p_aid=0&p_knoten=FID&p_sprache=D&p_suchstring=10913
- Swanson, J. O., Vogt, V., Sundmacher, L., Hagen, T. P., & Moger, T. A. (2018). Continuity of care and its effect on readmissions for COPD patients: A comparative study of Norway and Germany. *Health Policy*, 122(7), 737-745. doi:10.1016/j.healthpol.2018.05.013
- Swart, E. (2005). [What do we conclude from re-admissions about quality of inpatient care?]. *Gesundheitswesen*, 67(2), 101-106. doi:10.1055/s-2005-857880
- Takeda, A., Martin, N., Taylor, R. S., & Taylor, S. J. (2019). Disease management interventions for heart failure. *Cochrane Database Syst Rev*, 1, CD002752. doi:10.1002/14651858.CD002752.pub4
- Thygesen, K., Alpert, J. S., Jaffe, A. S., Chaitman, B. R., Bax, J. J., Morrow, D. A., . . . Executive Group on behalf of the Joint European Society of Cardiology /American College of Cardiology /American Heart Association /World Heart Federation Task Force for the

- Universal Definition of Myocardial, I. (2018). Fourth Universal Definition of Myocardial Infarction (2018). *Circulation*, 138(20), e618-e651. doi:10.1161/CIR.0000000000000617
- Tung, Y. C., Chang, G. M., Chang, H. Y., & Yu, T. H. (2017). Relationship between Early Physician Follow-Up and 30-Day Readmission after acute myocardial infarction and heart failure. *PLoS One*, 12(1), e0170061. doi:10.1371/journal.pone.0170061
- van der Stelt, C. A., Vermeulen Windsant-van den Tweel, A. M., Egberts, A. C., van den Bemt, P. M., Leendertse, A. J., Hermens, W. A., . . . Derijks, H. J. (2016). The Association Between Potentially Inappropriate Prescribing and Medication-Related Hospital Admissions in Older Patients: A Nested Case Control Study. *Drug Saf*, 39(1), 79-87. doi:10.1007/s40264-015-0361-1
- van Walraven, C., Bennett, C., Jennings, A., Austin, P. C., & Forster, A. J. (2011). Proportion of hospital readmissions deemed avoidable: a systematic review. *CMAJ*, 183(7), E391-402. doi:10.1503/cmaj.101860
- Wilke, T., Groth, A., Pfannkuche, M., Harks, O., Fuchs, A., Maywald, U., & Krabbe, B. (2015). Real life anticoagulation treatment of patients with atrial fibrillation in Germany: extent and causes of anticoagulant under-use. *J Thromb Thrombolysis*, 40(1), 97-107. doi:10.1007/s11239-014-1136-8
- Wu, J. Y., Leung, W. Y., Chang, S., Lee, B., Zee, B., Tong, P. C., & Chan, J. C. (2006). Effectiveness of telephone counselling by a pharmacist in reducing mortality in patients receiving polypharmacy: randomised controlled trial. *Bmj*, 333(7567), 522. doi:10.1136/bmj.38905.447118.2F
- Xu, Y., Yang, X., Huang, H., Peng, C., Ge, Y., Wu, H., . . . Yi, Y. (2019). Extreme Gradient Boosting Model Has a Better Performance in Predicting the Risk of 90-Day Readmissions in Patients with Ischaemic Stroke. *J Stroke Cerebrovasc Dis*, 28(12), 104441. doi:10.1016/j.jstrokecerebrovasdis.2019.104441
- Yam, C. H., Wong, E. L., Chan, F. W., Wong, F. Y., Leung, M. C., & Yeoh, E. K. (2010). Measuring and preventing potentially avoidable hospital readmissions: a review of the literature. *Hong Kong Med J*, 16(5), 383-389.
- Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1), 32-35. doi:10.1002/1097-0142(1950)3:1<32::aid-cnrcr2820030106>3.0.co;2-3
- Yu, S., Farooq, F., van Esbroeck, A., Fung, G., Anand, V., & Krishnapuram, B. (2015). Predicting readmission risk with institution-specific prediction models. *Artif Intell Med*, 65(2), 89-96. doi:10.1016/j.artmed.2015.08.005

11. Anhang

Anhang 1: Hintergrund zur Kodierung von (Re-) Hospitalisierungen

Anhang 2: Komorbiditäten der Patientenpopulationen nach den Elixhauser Gruppen (aus Nebendiagnosen).

Anhang 3: Soziodemographische Charakterisierung der Index-Fälle stratifiziert nach Fällen mit und ohne spezifische Rehospitalisierung.

Akronym: PREMISE

Förderkennzeichen: 01VSF18019

Anhang 4: Selektierte Prädiktoren im Rehospitalisierungsmodell für die einzelnen Erkrankungen unter Angabe der Odds Ratio (OR) basierend auf LASSO-Koeffizienten und unregularisierten Koeffizienten aus einem logistischen Regressionsmodell (GLM) inkl. p-Wert.

12. Anlagen

Anlage 1: Quellcode zur Publikation *Gerharz A et al. Methods Inf Med 2022*.

Anlage 2: Liste mit Definition und Operationalisierung der Prädiktorvariablen.

Ebenfalls auf Nachfrage verfügbar sind die Arbeitsdokumente zur Literaturrecherche (siehe **Abschnitt 5.6.1**).

Anhang 1: Hintergrund zur Kodierung von (Re-) Hospitalisierungen

Die Kodierung von stationären Diagnosen unterliegt hohen Ansprüchen. Maßgeblich hierfür sind „Allgemeine und spezifische Kodierrichtlinien“, welche jährlich vom Institut für das Entgeltsystem im Krankenhaus (InEK) herausgegeben werden. Darin sind auch Besonderheiten, sowie Fallbeispiele zu einzelnen Erkrankungen enthalten. Kodierte stationäre Diagnosen haben einen direkten Einfluss auf die Vergütung durch das pauschalierte Abrechnungssystem in deutschen Krankenhäusern. Daher sollen die Kodierrichtlinien hier unterstützend wirken, die notwendige Kodierqualität in Krankenhäusern zu erzielen, um eine einheitliche Kodierung identischer Krankenhausfälle zu gewährleisten (Institut für das Entgeltsystem im Krankenhaus GmbH (InEK), 2019). Geregelt sind die Vergabe der Hauptdiagnosen, von Nebendiagnose, das pauschalierende Vergütungssystem nach DRG (so genannte „Diagnosis related groups“ (DRG) – „DRG“-Fallpauschalen), das pauschalierende Entgeltsystem Psychiatrie, Psychotherapie und Psychosomatik („PEPP-System“), die Eingruppierung eines Krankenhausfalls in eine DRG-Fallpauschale anhand einer MDC (Major Diagnostic Category), die Einteilung von Nebendiagnosen in Schweregradstufen eingeteilt (Komplikations- und Komorbiditätslevel „complication and comorbidity Level“ – CCL) oder die Ermittlung der Entgelthöhe der jeweiligen Fallpauschale und die für die entsprechende DRG vorgesehene obere Grenzverweildauer (oGVD).

Die Grenzverweildauern sind auch für eine Fallzusammenführung nach der Fallpauschalenvereinbarung wichtig, welche sich dann wiederum in der Abrechnung niederschlägt. Demnach müssen unter bestimmten Umständen mehrere Behandlungsepisoden eines Patienten als ein Fall pauschaliert abgerechnet werden (GKV-Spitzenverband, Verband der Privaten Krankenversicherung, & Deutsche Krankenhausgesellschaft, 2018). In Abbildung S1 dargestellt sind die zu überprüfenden Angaben. Neben direkten Wiederaufnahme in das gleiche Krankenhaus müssen auch Krankenhausfälle zusammengeführt werden, bei denen eine Rückverlegung stattfindet (GKV-Spitzenverband et al., 2018). Eine solche Situation liegt vor, wenn ein Patient von einem Krankenhaus in ein weiteres verlegt wird und innerhalb von 30 Kalendertagen ab Entlassungsdatum aus dem ersten Krankenhaus in dieses erste Krankenhaus zurückverlegt wird. Dann werden die Fälle im ersten Krankenhaus werden zu einem Behandlungsfall zusammengefasst und es wird eine Neueinstufung der Fallpauschale vorgenommen.

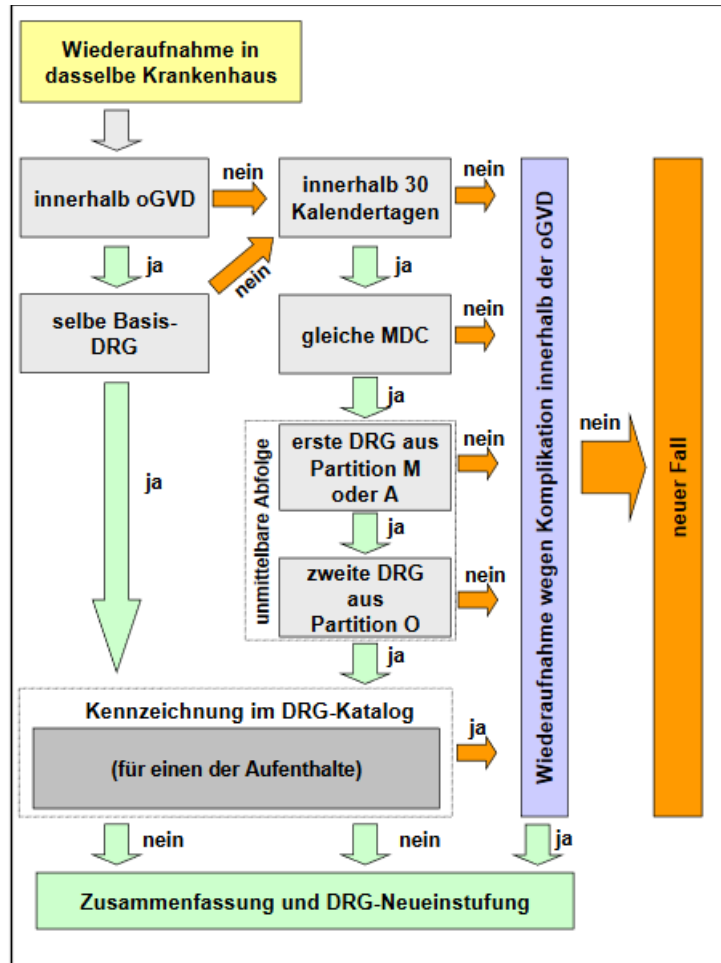


Abb. S1: Ablaufschema zur Prüfung einer möglichen Fallzusammenführung (adaptiert aus (Bundesministerium für Gesundheit und Soziale Sicherung (Referat 216), 2004)). Unter folgenden Bedingungen muss das Krankenhaus bei einer Wiederaufnahme des Patienten die Falldaten der beiden Fälle zu einem Fall zusammenfassen und eine Neueinstufung der Fallpauschale vornehmen, außer wenn für mindestens eine der DRG eine Ausnahme von der Wiederaufnahme im Fallpauschalen-Katalog vermerkt ist: Wenn der Patient innerhalb der oberen Grenzverweildauer ab Aufnahmedatum des ersten Falls in das gleiche Krankenhaus wieder aufgenommen wird und beide Fälle in die gleiche Basis-DRG eingestuft werden. Wenn der Patient innerhalb von 30 Kalendertagen ab Aufnahmedatum des ersten Falls in das gleiche Krankenhaus wieder aufgenommen wird und beide Fälle in die gleiche MDC eingestuft werden, sowie eine unmittelbare Diagnostik-Operation-Folge vorliegt, d. h. die DRG des ersten Falls in die medizinische oder andere Partition und die DRG des zweiten Falls in die operative Partition fällt. Wenn der Patient innerhalb der oberen Grenzverweildauer ab Aufnahmedatum des ersten Falls aufgrund einer Komplikation des ersten Falls in das gleiche Krankenhaus wieder aufgenommen wird und es sich bei der Komplikation nicht um unvermeidbare Nebenwirkungen einer onkologischen Behandlung handelt.

DRG: Diagnosis-related group. **MDC:** Major-Diagnostic-Category. **oGVD:** obere Grenzverweildauer. Partition A: Partition mit anderen Fallpauschalen. Partition M: Partition mit medizinischen Fallpauschalen. Partition O: Partition mit operativen Fallpauschalen.

Anhang 2: Komorbiditäten der Patientenpopulationen nach den Elixhauser Gruppen (aus Nebendiagnosen).

Komorbidität	Anteil Patienten mit Index-Erkrankung und Komorbidität (% und absolute Anzahl)					
	CPDP	Herzinsuffizienz	Akuter Myokardinfarkt	Osteoporose	Diabetes mellitus Typ 2	Schlaganfall, TIA oder Vorhofflimmern
Kongestive Herzinsuffizienz	33,47 % (5281)	61,19 % (32366)	46,45 % (7893)	13,21 % (703)	21,62 % (3790)	20,04 % (13178)
Kardiale Arrhythmien	30,88 % (4873)	65,26 % (34517)	39,77 % (6757)	23,02 % (1225)	29,05 % (5093)	31,97 % (21017)
Erkrankung der Herzklappen	8,92 % (1408)	32,29 % (17080)	19,99 % (3396)	5,47 % (291)	6,86 % (1202)	15,01 % (9872)
Pulmonale Herzkrankheit und Krankheiten des Lungenkreislaufes	11,12 % (1755)	18,81 % (9950)	6,93 % (1178)	2,42 % (129)	2,7 % (474)	4,09 % (2688)
Periphere Gefäßerkrankung	9,75 % (1539)	11,71 % (6191)	11,62 % (1975)	5,39 % (287)	33,54 % (5879)	8,08 % (5314)
Essentielle Hypertonie	56,63 % (8935)	52,8 % (27926)	66,09 % (11230)	62,81 % (3342)	65,21 % (11432)	68,49 % (45031)
Hypertone Herzkrankheit/ Sekundäre Hypertonie	10,54 % (1663)	21,15 % (11188)	15,4 % (2616)	4,17 % (222)	9,26 % (1624)	10,16 % (6677)
Lähmung	2,28 % (359)	2,37 % (1252)	2,18 % (370)	2,18 % (116)	4,23 % (742)	27,80 % (18278)
Andere neurologische Erkrankungen	4,18 % (659)	4,43 % (2342)	3,81 % (647)	5,39 % (287)	6,06 % (1062)	17,27 % (11353)
Chronische Lungenerkrankung	14,61 % (2305)	18,75 % (9917)	10,29 % (1748)	11,45 % (609)	8,60 % (1507)	6,69 % (4396)
Diabetes mellitus, ohne Komplikationen	22,06 % (3481)	23,75 % (12561)	25,53 % (4337)	13,19 % (702)	1,00 % (175)	20,99 % (13801)
Diabetes mellitus, mit Komplikationen	8,46 % (1335)	16,26 % (8599)	11,39 % (1935)	3,72 % (198)	36,57 % (6410)	6,02 % (3961)
Hypothyreose	12,94 % (2041)	14,2 % (7508)	12,47 % (2119)	16,71 % (889)	12,9 % (2262)	14,16 % (9312)
Niereninsuffizienz/-versagen	28,47 % (4492)	53,15 % (28111)	34,38 % (5841)	19,23 % (1023)	43,84 % (7686)	20,22 % (13293)
Lebererkrankung	2,03 % (321)	2,99 % (1579)	1,3 % (221)	1,84 % (98)	4,78 % (838)	1,15 % (754)
Peptische Ulkuskrankheit ohne Blutung	0,10 % (16)	0,22 % (119)	0,17 % (29)	0,08 % (4)	0,20 % (35)	0,13 % (87)
AIDS/HIV	0,01 % (1)	0,00 % (1)	0,01 % (1)	0,00 % (0)	0,00 % (0)	0,00 % (1)
Lymphom	0,24 % (38)	0,38 % (199)	0,20 % (34)	0,53 % (28)	0,19 % (34)	0,20 % (129)
Metastasierende Krebserkrankung	0,83 % (131)	0,82 % (432)	0,61 % (103)	0,81 % (43)	0,69 % (121)	0,56 % (369)

Solider Tumor, ohne Metastasen	3,33 % (526)	2,36 % (1250)	1,94 % (330)	2,37 % (126)	2,06 % (361)	1,64 % (1079)
Rheumatische Erkrankung	2,06 % (325)	1,86 % (982)	1,77 % (301)	6,05 % (322)	2,16 % (379)	1,63 % (1072)
Koagulopathie	2,74 % (433)	6,11 % (3229)	4,18 % (710)	5,34 % (284)	3,66 % (642)	2,49 % (1634)
Adipositas	8,05 % (1270)	8,45 % (4471)	6,19 % (1051)	4,00 % (213)	10,19 % (1786)	4,39 % (2888)
Gewichtsverlust	3,56 % (561)	1,44 % (764)	0,82 % (139)	1,58 % (84)	1,35 % (237)	0,68 % (449)
Störungen des Wasser- und Elektrolythaushaltes	32,67 % (5154)	39,28 % (20775)	28,79 % (4891)	23,27 % (1238)	32,56 % (5708)	19,08 % (12547)
Blutungsanämie	0,46 % (73)	0,87 % (462)	0,43 % (73)	0,32 % (17)	0,42 % (74)	0,18 % (121)
Mangelanämie	3,00 % (474)	5,21 % (2757)	2,77 % (471)	1,47 % (78)	3,21 % (562)	1,48 % (971)
Alkoholabusus	1,62 % (256)	0,95 % (501)	0,65 % (110)	0,60 % (32)	1,42 % (249)	0,96 % (629)
Drogenabusus	0,28 % (44)	0,14 % (74)	0,37 % (63)	0,60 % (32)	0,14 % (25)	0,14 % (90)
Psychotische Störung	0,61 % (97)	0,43 % (225)	0,42 % (71)	0,24 % (13)	0,80 % (141)	0,41 % (270)
Depression	8,05 % (1270)	5,75 % (3042)	3,28 % (558)	8,16 % (434)	5,93 % (1040)	5,05 % (3320)

Anmerkung: Der angegebene Anteil beschreibt jeweils den betroffenen Anteil der Patienten mit Erkrankung, die ebenfalls an der jeweiligen Komorbidität litten. Zum Beispiel litten 8,05 % der COPD-Patienten auch an Depression als Komorbidität.
AIDS: Acquired Immune Deficiency Syndrome. **COPD:** Chronisch-obstruktive Lungenerkrankung. **HIV:** Humanes Immundefizienz-Virus. **TIA:** Transitorische ischämische Attacke.

Anhang 3: Soziodemographische Charakterisierung der Index-Fälle stratifiziert nach Fällen mit und ohne spezifische Rehospitalisierung.

		Index-Hospitalisierung											
		COPD		Herzinsuffizienz		Akuter Myokardinfarkt		Osteoporose		Diabetes mellitus Typ 2		Schlaganfall, TIA oder Vorhofflimmern	
Anzahl Index-Fälle		28446		80870		18893		6111		23665		81632	
	Anzahl Patienten (gesamt)	15778		52890		16991		5321		17530		65750	
		Spezifisch?		Spezifisch?		Spezifisch?		Spezifisch?		Spezifisch?		Spezifisch?	
		ja	nein	ja	nein	ja	nein	ja	nein	ja	nein	ja	nein
Anzahl Index-Fälle (%)		7217 (25,37)	21229 (74,63)	16749 (20,71)	64121 (79,29)	4116 (21,79)	14777 (78,21)	664 (10,87)	5447 (89,13)	2621 (11,08)	21044 (88,92)	4397 (5,39)	77235 (94,61)
	Anzahl Patienten (%)*	4155 (26,33)	14698 (93,16)	12184 (23,04)	49362 (93,33)	3846 (22,64)	13919 (81,92)	595 (11,18)	5081 (95,49)	2049 (11,69)	17115 (97,63)	4153 (6,32)	63524 (96,61)
Anzahl Männer (%)#		4427 (61,34)	11862 (55,88)	7687 (45,90)	26758 (41,73)	2320 (56,37)	7711 (52,18)	88 (13,25)	777 (14,26)	1461 (55,74)	9876 (46,93)	2027 (46,10)	31054 (40,21)
	Alter (Mittel ± SD)	77,24 ± 6,05	78,28 ± 6,51	81,4 ± 6,60	81,84 ± 6,71	79,03 ± 6,46	79,72 ± 6,77	80,65 ± 6,20	81,3 ± 6,44	78,43 ± 6,27	78,88 ± 6,52	80,51 ± 6,55	79,48 ± 6,59
Anzahl Hosp. im Vorjahr (Mittel ± SD)		2,62 ± 2,50	1,49 ± 1,77	2,01 ± 2,01	1,45 ± 1,68	0,94 ± 1,48	0,80 ± 1,29	1,20 ± 1,63	1,10 ± 1,50	1,82 ± 1,82	1,24 ± 1,56	0,95 ± 1,30	0,83 ± 1,25

*Der prozentuale Anteil wurde anhand der Gesamtanzahl an Patienten, die Index-Fälle generierten, berechnet. Jeder Patient konnte mehr als einen Index-Fall beitragen, auf welchen eine Rehospitalisierung folgte oder nicht folgte. Die Summe des prozentualen Anteils an Patienten weicht daher von 100 % ab.

#Der prozentuale Anteil der durch Männer verursachten Fälle bezieht sich auf die Anzahl an Index-Fälle, jeweils unterschieden in Fälle mit Rehospitalisierung bzw. Fälle ohne Rehospitalisierung. Die Summe des prozentualen Anteils weicht daher von 100 % ab.

Anz.: Anzahl. **Bzw.:** beziehungsweise. **SD:** Standard Deviation = Standardabweichung. **TIA:** Transitorische ischämische Attacke.

Anhang 4: Selektierte Prädiktoren im Rehospitalisierungsmodell für die einzelnen Erkrankungen unter Angabe der Odds Ratio (OR) basierend auf LASSO-Koeffizienten und unregularisierten Koeffizienten aus einem logistischen Regressionsmodell (GLM) inkl. p-Wert. Regressionskoeffizienten sind durch Logarithmieren der gelisteten ORs zu erhalten.

Variable	Inhaltliche Bedeutung der Variable	OR (LASSO)	OR (GLM)	p-Wert (GLM)
COPD (chronisch obstruktive Lungenerkrankung)				
(Intercept)		0,65	0,82	0,305
STOPP B11	ACE-Inhibitoren/ ARB bei Hyperkaliämie	1,15	1,36	0,003*
STOPP B13	PDE-5-Hemmer bei HF/Nitrat-Therapie für Angina pectoris	1,23	1,89	0,015*
STOPP D8	Anticholinergika/ Antimuskarinika bei Delirium oder Demenz	1,80	> 100**	0,944
STOPP D13	Levodopa oder Dopaminagonisten bei benignem essenziellem Tremor	1,43	2,56	0,004*
STOPP F1	Prochlorperazin/ MCP bei Parkinsonismus	0,93	0,52	0,080
STOPP F3	Obstipierende AM bei chron. Obstipation	1,03	1,17	0,054
STOPP G1	Theophyllin als Monotherapie für COPD	0,94	0,68	0,047*
STOPP G3	Antimuskarine Bronchodilatoren bei Anamnese von Engwinkelglaukom/ obstruktive Uropathie	1,09	1,17	0,018*
STOPP H5	Glucocorticoide bei Arthrose	1,13	1,21	< 0,001*
STOPP J1	Sulfonylharnstoffe mit langer Wirkdauer bei DM	1,01	1,18	0,062
Herzinsuffizienz				
(Intercept)		0,20	0,21	< 0,001*
STOPP B9	Schleifendiuretika bei Hypertonie mit Harninkontinenz	0,94	0,90	0,004*
STOPP B11	ACE-Inhibitoren/ARB bei Hyperkaliämie	1,04	1,09	0,121
STOPP B12	Aldosteronantagonisten mit kaliumsparenden AM	1,03	1,06	0,024*
STOPP B13	PDE-5-Hemmer bei HF/Nitrat-Therapie für Angina pectoris	0,92	0,50	0,119
STOPP C3	TAH oder Antikoagulanzen bei Blutungsrisiko	1,01	1,03	0,33
STOPP C6	TAH und VKA/Antikoagulanzen bei stabiler koronarer Erkrankung, zerebrovaskulärer oder peripher arterieller Erkrankung	1,20	1,32	< 0,001*
STOPP D4	SSRI bei Hyponatriämie	1,19	1,40	0,008*
STOPP D13	Levodopa oder Dopaminagonisten bei benignem essenziellem Tremor	1,00	1,41	0,136
STOPP E2	330 ml/min/1,73m ²	0,96	0,84	0,048*
STOPP E3	Faktor Xa-Inhibitoren bei eGFR < 15 ml/min/1,73m ²	1,09	1,11	0,002*
STOPP H9	p.o. Bisphosphonate bei Anamnese von Erkrankungen des oberen GIT	1,04	1,36	0,135
STOPP I1	Antimuskarinika für überaktive Blase bei Demenz/kognitive Beeinträchtigung/ Engwinkelglaukom	0,95	0,76	0,15
STOPP I2	α-Blocker bei Hypotonie/Miktionssynkope	0,97	0,68	0,12

STOPP J3	Beta-Blocker bei DM mit häufigen hypoglykämischen Episoden	1,08	1,20	0,066
STOPP K2	Neuroleptika aufgrund Sturzrisiko	0,92	0,84	0,002*
STOPP K4	Z-Substanzen aufgrund Sturzrisiko	1,07	1,19	0,017*
STOPP M1	≥ zwei AM mit anticholinergen Eigenschaften	0,89	0,69	0,065
START A3	TAH bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung	1,01	1,02	0,374
START A7	Beta-Blocker bei ischämischer Herzkrankheit	1,01	1,08	0,020*
START A8	geeigneter Beta-Blocker bei systolischer HF	0,99	0,93	0,007*
START C3	TAH oder Antikoagulanzen bei Blutungsrisiko	0,92	0,90	< 0,001*
START E4	Antiresorptive oder knochenanabole Therapie bei OS	0,98	0,96	0,151
Akuter Myokardinfarkt				
(Intercept)		0,60	0,73	0,173
STOPP B3	Beta-Blocker mit Verapamil/Diltiazem	1,01	1,17	0,357
STOPP B7	Schleifendiuretika bei Knöchelödem	1,28	1,65	0,044*
STOPP B8	Thiaziddiuretika bei Hypokaliämie, Hyponatriämie, Hypercalcämie oder Gicht	1,04	1,11	0,155
STOPP B9	Schleifendiuretika bei Hypertonie mit Harn-inkontinenz	0,99	0,91	0,359
STOPP B12	Aldosteronantagonisten mit kaliumsparenden AM	1,16	1,19	0,002*
STOPP C5	ASS und VKA/Antikoagulanzen bei VHF	1,03	1,08	0,494
STOPP C6	TAH und VKA/Antikoagulanzen bei stabiler koronarer Erkrankung, zerebrovaskulärer oder peripher arterieller Erkrankung	1,11	1,10	0,331
STOPP D3	Neuroleptika mit antimuskarinen Effekten	3,36	8,11	0,096
STOPP D7	Anticholinergika für EPMS UAW	0,76	0,31	0,263
STOPP E3	Faktor Xa-Inhibitoren bei eGFR < 15 ml/min/1,73m ²	1,05	1,14	0,266
STOPP E6	Metformin bei eGFR < 30 ml/min/1,73m ²	1,03	1,07	0,503
STOPP G1	Theophyllin als Monotherapie für COPD	0,92	0,55	0,341
STOPP H5	Glucocorticoide bei Arthrose	0,96	0,87	0,117
STOPP H8	NSAR mit GC ohne PPI	0,80	0,64	0,095
STOPP I1	Antimuskarinika für überaktive Blase bei Demenz/kognitive Beeinträchtigung/ Engwinkelglaukom	0,93	0,64	0,264
STOPP J3	Beta-Blocker bei DM mit häufigen hypoglykämischen Episoden	1,40	1,60	0,036*
STOPP K3	Vasodilatoren aufgrund Sturzrisiko	0,80	0,41	0,238
STOPP K4	Z-Substanzen aufgrund Sturzrisiko	1,20	1,41	0,084
START A2	ASS bei chronischem AF wegen Kontraindikation	1,02	1,11	0,37
START A3	TAH bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung	0,97	0,93	0,174
START A5	Statine bei Vorgeschichte von koronarer, zerebraler/peripher vaskulärer Erkrankung	0,96	0,95	0,28
START A7	Beta-Blocker bei ischämischer Herzkrankheit	0,95	0,87	0,025*
START C2	Non-Trizyklika bei <i>major</i> Depression	1,02	1,07	0,137
START C3	TAH oder Antikoagulanzen bei Blutungsrisiko	0,91	0,87	0,022*

START E4	Antiresorptive oder knochenanabole Therapie bei OS	1,01	1,08	0,168
START E6	Xanthin-Oxidase-Hemmer bei Gicht	1,03	1,10	0,292
Osteoporose				
(Intercept)		0,25	0,24	0,015*
STOPP B3	Beta-Blocker mit Verapamil/Diltiazem	0,65	0,42	0,404
STOPP B4	Beta-Blocker bei Bradykardie/Herzblock	1,80	2,91	0,196
STOPP B7	Schleifendiuretika bei Knöchelödem	1,85	1,96	0,047*
STOPP B8	Thiaziddiuretika bei Hypokaliämie, Hyponatriämie, Hypercalcämie oder Gicht	0,98	0,94	0,76
STOPP B9	Schleifendiuretika bei Hypertonie mit Harn-inkontinenz	1,08	1,20	0,419
STOPP B12	Aldosteronantagonisten mit kaliumsparenden AM	0,74	0,68	0,254
STOPP C3	TAH oder Antikoagulanzen bei Blutungsrisiko	0,85	0,78	0,286
STOPP C5	ASS und VKA/Antikoagulanzen bei VHF	0,84	0,75	0,785
STOPP C6	TAH und VKA/Antikoagulanzen bei stabiler koronarer Erkrankung, zerebrovaskulärer oder peripher arterieller Erkrankung	0,94	0,82	0,85
STOPP C7	Ticlopidin unter jeglichem Umstand	0,54	0,00	0,987
STOPP C10	NSAR mit Antikoagulanzen	1,01	1,06	0,76
STOPP C11	NSAR mit TAH ohne PPI	0,98	0,91	0,801
STOPP D1	Trizyklika bei Demenz/Engwinkelglaukom/ Erregungsleitungsstörungen/ Prostatahypertrophie/Harnretention	0,77	0,66	0,394
STOPP D4	SSRI bei Hyponatriämie	0,74	0,54	0,413
STOPP D6	Antipsychotika bei Parkinsonismus	0,97	0,79	0,766
STOPP D13	Levodopa oder Dopaminagonisten bei benignem essen- tiellem Tremor	1,42	1,71	0,42
STOPP E2	Direkte Thrombin-Inhibitoren bei eGFR < 30 ml/min/1,73m ²	0,83	0,56	0,583
STOPP E4	NSAR bei eGFR < 50 ml/min/1,73m ²	1,29	1,43	0,035*
STOPP E5	Colchicin, bei eGFR < 15 ml/min/1,73m ²	0,62	0,00	0,986
STOPP E6	Metformin bei eGFR < 30 ml/min/1,73m ²	0,46	0,33	0,128
STOPP F1	Prochlorperazin/ MCP bei Parkinsonismus	1,02	1,25	0,659
STOPP F3	Obstipierende AM bei chron. Obstipation	1,09	1,17	0,244
STOPP G2	Systemische statt inhalative GC bei COPD	1,47	1,54	0,048*
STOPP G3	Antimuskarine Bronchodilatoren bei Anamnese von Engwinkelglaukom/ obstruktive Uropathie	0,97	0,67	0,703
STOPP H1	NSAR bei Ulcera/GIT-Blutungen ohne PPI	0,91	0,77	0,519
STOPP H5	Glucocorticoide bei Arthrose	1,07	1,11	0,424
STOPP H7	COX-2-selektive NSAR bei kardiovaskulärer Erkrankung	1,21	1,33	0,269
STOPP H8	NSAR mit GC ohne PPI	0,69	0,59	0,084
STOPP H9	p.o. Bisphosphonate bei Anamnese von Erkrankungen des oberen GIT	0,75	0,67	0,384
STOPP I1	Antimuskarinika für überaktive Blase bei Demenz/kogni- tive Beeinträchtigung/ Engwinkelglaukom	1,38	1,93	0,411

STOPP I2	α-Blocker bei Hypotonie/Miktionssynkope	0,24	0,00	0,984
STOPP J1	Sulfonylharnstoffe mit langer Wirkdauer bei DM	1,11	1,23	0,511
STOPP J3	Beta-Blocker bei DM mit häufigen hypoglykämischen Episoden	2,40	3,14	0,097
STOPP J4	Estrogene bei Anamnese von Brustkrebs/VTE	0,27	0,00	0,975
STOPP K1	Benzodiazepine aufgrund Sturzrisiko	0,63	0,55	0,056
STOPP K3	Vasodilatoren aufgrund Sturzrisiko	2,94	5,09	0,199
STOPP M1	≥ zwei AM mit anticholinergen Eigenschaften	0,99	0,83	0,8
START A2	ASS bei chronischem AF wegen Kontraindikation	1,00	2,07	0,183
START A3	TAH bei Vorgeschichte von koronarer, zerebraler oder peripher vaskulärer Erkrankung	1,14	1,19	0,197
START A7	Beta-Blocker bei ischämischer Herzkrankheit	0,98	0,93	0,621
START C2	Non-Trizyklika bei <i>major</i> Depression	1,13	1,17	0,111
START C3	TAH oder Antikoagulanzen bei Blutungsrisiko	0,84	0,81	0,12
START D1	PPI bei Reflux oder peptischen Stenosen	1,30	1,59	0,201
START E4	Antiresorptive oder knochenanabole Therapie bei OS	1,11	1,16	0,121
START E6	Xanthin-Oxidase-Hemmer bei Gicht	0,89	0,79	0,438
START F1	ACE-Hemmer oder ARB bei DM mit renaler Erkrankung	0,59	0,00	0,992
Diabetes mellitus Typ 2				
(Intercept)		0,07	0,15	< 0,001*
STOPP B12	Aldosteronantagonisten mit kaliumsparenden AM	1,04	1,21	0,038*
STOPP B13	PDE-5-Hemmer bei HF/Nitrat-Therapie für Angina pectoris	1,36	7,49	0,034*
STOPP C6	TAH und VKA/Antikoagulanzen bei stabiler koronarer Erkrankung, zerebrovaskulärer oder peripher arterieller Erkrankung	1,04	1,23	0,293
STOPP C7	Ticlopidin unter jeglichem Umstand	2,09	5,95	0,023*
START A5	Statine bei Vorgeschichte von koronarer, zerebraler/peripher vaskulärer Erkrankung	1,02	1,07	0,157
Schlaganfall, transitorische ischämische Attacke und Vorhofflimmern				
(Intercept)		0,02	0,01	< 0,001*
STOPP B3	Beta-Blocker mit Verapamil/Diltiazem	0,77	0,60	0,013*
STOPP B4	Beta-Blocker bei Bradykardie/Herzblock	0,79	0,61	0,010*
STOPP B9	Schleifendiuretika bei Hypertonie mit Harn-inkontinenz	1,15	1,22	0,016*
STOPP B12	Aldosteronantagonisten mit kaliumsparenden AM	0,93	0,84	0,052
STOPP C2	ASS ohne PPI bei Anamnese von Ulcera	1,22	1,31	0,08
STOPP C3	TAH oder Antikoagulanzen bei Blutungsrisiko	1,11	1,15	0,017*
STOPP C10	NSAR mit Antikoagulanzen	0,83	0,72	< 0,001*
STOPP C11	NSAR mit TAH ohne PPI	1,06	1,17	0,131
STOPP D4	SSRI bei Hyponatriämie	1,08	1,31	0,282
STOPP D6	Antipsychotika bei Parkinsonismus	1,01	1,15	0,455
STOPP D8	Anticholinergika/ Antimuskarinika bei Delirium oder Demenz	2,53	3,60	0,16

STOPP E4	NSAR bei eGFR < 50 ml/min/1,73m ²	1,18	1,33	0,001*
STOPP E5	Colchicin, bei eGFR < 15 ml/min/1,73m ²	0,73	0,21	0,12
STOPP G1	Theophyllin als Monotherapie für COPD	1,31	1,71	0,099
STOPP G3	Antimuskarine Bronchodilatoren bei Anamnese von Engwinkelglaukom/ obstruktive Uropathie	0,67	0,31	0,048*
STOPP H9	p.o. Bisphosphonate bei Anamnese von Erkrankungen des oberen GIT	1,25	1,68	0,155
STOPP I1	Antimuskarinika für überaktive Blase bei Demenz/kognitive Beeinträchtigung/ Engwinkelglaukom	1,16	1,42	0,153
STOPP J4	Estrogene bei Anamnese von Brustkrebs/ VTE	0,74	0,00	0,917
STOPP K2	Neuroleptika aufgrund Sturzrisiko	1,15	1,21	0,044*
START A2	ASS bei chronischem AF wegen Kontraindikation	0,99	0,91	0,326
START A5	Statine bei Vorgeschichte von koronarer, zerebraler/peripher vaskulärer Erkrankung	1,02	1,04	0,339
START C3	TAH oder Antikoagulanzen bei Blutungsrisiko	1,19	1,20	< 0,001*
START E6	Xanthin-Oxidase-Hemmer bei Gicht	1,02	1,11	0,222
START F1	ACE-Hemmer oder ARB bei DM mit renaler Erkrankung	1,20	3,24	0,303

*signifikanter p-Wert (< 0,05) **aufgrund der äußerst geringen Prävalenz des Kriteriums wurde die *Odds Ratio* auf 996495,60 geschätzt. **ACE:** Angiotensin-converting enzyme. **ARB:** Angiotensin-Rezeptor-Blocker. **AF:** Vorhofflimmern. **AM:** Arzneimittel. **AMI:** Akuter Myokardinfarkt. **ASS:** Acetylsalicylsäure. **chron.:** chronischer. **COPD:** Chronisch-obstruktive Lungenerkrankung. **COX-2:** Cyclooxygenase-2. **DM:** Diabetes mellitus Typ 2. **EPMS:** Extra-pyramidalmotorische Störungen. **GC:** Glucocorticoide. **GIT:** Gastrointestinaltrakt. **HF:** Herzinsuffizienz. **MCP:** Metoclopramid. **NSAR:** Nicht-steroidale Antirheumatika. **PDE-5:** Phosphodiesterase-5. **PPI:** Protonenpumpeninhibitor. **p.o.:** per orale. **SSRI:** Selektive Serotonin Reuptake-Inhibitoren. **TAH:** Thrombozytenaggregationshemmer. **VKA:** Vitamin K-Antagonisten. **S/AF:** Schlaganfall, transitorische ischämische Attacke und Vorhofflimmern. **UAW:** unerwünschte Arzneimittelwirkungen. **VTE:** Venöse Thromboembolie.

Anlage 1: Quellcode zur Publikation Gerharz A et al. *Methods Inf Med* 2022.

```
#####

library(ggplot2)
library(reshape2)
library(magrittr)
library(stringr)
library(data.table)
library(xlsx)
library(glmnet)
library(class)
library(pROC)

## Datasets
copd <- fread(file = "./data/part_of_all_features_modeling/copd90_all_rel_data.csv", sep2 = c(" ", ";", ""))
copd <- copd[, -45]
copd <- copd[, -69]
hf_ukhd <- fread(file = "./data/part_of_all_features_modeling/hf_ukhd90_all_rel_data.csv", sep2 = c(" ", ";", "")) %>%
  .[sex == "männlich", sex := "männlich"]
hf_ukhd <- hf_ukhd[, -45]
hf_ukhd <- hf_ukhd[, -69]
mi_ukhd <- fread(file = "./data/part_of_all_features_modeling/mi_ukhd90_all_rel_data.csv", sep2 = c(" ", ";", "")) %>%
  .[sex == "männlich", sex := "männlich"]
mi_ukhd <- mi_ukhd[, -45]
mi_ukhd <- mi_ukhd[, -69]
osteo <- fread(file = "./data/part_of_all_features_modeling/osteo90_all_rel_data.csv", sep2 = c(" ", ";", "")) %>%
  .[sex == "männlich", sex := "männlich"]
osteo <- osteo[, -45]
osteo <- osteo[, -69]
t2dm <- fread(file = "./data/part_of_all_features_modeling/t2dm90_all_rel_data.csv", sep2 = c(" ", ";", "")) %>%
  .[sex == "männlich", sex := "männlich"]
t2dm <- t2dm[, -45]
t2dm <- t2dm[, -69]
vhf <- fread(file = "./data/part_of_all_features_modeling/vhf90_all_rel_data.csv", sep2 = c(" ", ";", "")) %>%
  .[sex == "männlich", sex := "männlich"]
vhf <- vhf[, -45]
vhf <- vhf[, -69]

## Datasets original
# Read Function
{
  read.krakdata <- function(file, krankheit, thirty = TRUE){
    data <- fread(file) %>%
      .[, BEGINN := as.POSIXct(BEGINN %>% as.character, format = "%Y%m%d")] %>%
      .[, ENDE := as.POSIXct(ENDE %>% as.character, format = "%Y%m%d")] %>%
      .[, fBEGINN := as.POSIXct(fBEGINN %>% as.character, format = "%Y%m%d")] %>%
      .[, fENDE := as.POSIXct(fENDE %>% as.character, format = "%Y%m%d")] %>%
      .[, row := (1:N) %>%
        as.character] %>%
      .[, OPS_Datum := str_split(`OPS-Datum`, " ") %>%
        lapply(., FUN = function(x){as.POSIXct(x, format = "%Y%m%d")})
      ] %>%
      .[, `OPS-Datum` := NULL] %>%
      .[, OPS_Codes := str_split(`OPS-Codes`, " ") %>%
        .[, `OPS-Codes` := NULL] %>%
        .[, IK_KH := `IK-KH`] %>%
        .[, `IK-KH` := NULL] %>%
        .[, NEBENDIAG := str_split(NEBENDIAG, " ") %>%
          .[, TotIndex := as.logical(TotIndex)] %>%
          .[, INDEXFALLISTLETZTERFALL := as.logical(INDEXFALLISTLETZTERFALL)] %>%
          .[, eval(paste0("Rehosp_", ifelse(thirty, 30, 90), "_all_cause")))] %>%
          .[, eval(paste0("Rehosp_", ifelse(thirty, 30, 90), "_specific")))] %>%
          .[, eval(paste0("FAUFNDIAG ", krankheit))] := as.logical(eval(as.name(paste0("FAUFNDIAG ", krankheit)))) %>%
          .[, eval(paste0("FHD ", krankheit))] := as.logical(eval(as.name(paste0("FHD ", krankheit)))) %>%
          .[, eval(paste0("REHOSP ", krankheit))] := as.logical(eval(as.name(paste0("REHOSP ", krankheit))))
        ]
      ]
  }
}

# Datasets
{
  copd_90 <- read.krakdata("./incoming/LieferungMaerz/Erkrankungen/COPD_all_specific_90_korrigiert.csv", krankheit = "COPD", thirty = FALSE) %>%
    .[, row := NULL] %>%
    .[, Index_technisch := NULL] %>%
    .[, Rehosp_technisch := NULL] %>%
    .[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]

  hf_ukhd_90 <- read.krakdata("./incoming/LieferungMaerz/Erkrankungen/HF_UKHD_all_specific_90_korrigiert.csv", krankheit = "HF_UKHD", thirty =
  FALSE) %>%
    .[, row := NULL] %>%
    .[, Index_technisch := NULL] %>%
    .[, Rehosp_technisch := NULL] %>%
    .[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]

  mi_ukhd_90 <- read.krakdata("./incoming/LieferungMaerz/Erkrankungen/MI_UKHD_all_specific_90_korrigiert.csv", krankheit = "MI_UKHD", thirty =
  FALSE) %>%
    .[, row := NULL] %>%
    .[, Index_technisch := NULL] %>%
    .[, Rehosp_technisch := NULL] %>%
    .[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]

  osteo_90 <- read.krakdata("./incoming/LieferungMaerz/Erkrankungen/OSTEO_all_specific_90_korrigiert.csv", krankheit = "OSTEO", thirty = FALSE) %>%
    .[, row := NULL] %>%

```



```

.[, Index_technisch := NULL] %>%
.[, RehosP_technisch := NULL] %>%
.[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]

t2dm_90 <- read.krankdata("./incoming/LieferungMaerz/Erkrankungen/T2DM_all_specific_90_korrigiert.csv", krankheit = "T2DM", thirty = FALSE) %>%
.[, row := NULL] %>%
.[, Index_technisch := NULL] %>%
.[, RehosP_technisch := NULL] %>%
.[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]

vhf_90 <- read.krankdata("./incoming/LieferungMaerz/Erkrankungen/VHF_all_specific_90_korrigiert.csv", krankheit = "VHF", thirty = FALSE) %>%
.[, row := NULL] %>%
.[, Index_technisch := NULL] %>%
.[, RehosP_technisch := NULL] %>%
.[, VERSIDersterINDEXTOTRAUS_FINAL := NULL]
}

# Versicherter Data
{
versicherter_data <- fread("./incoming/LieferungMaerz/Weitere Daten/VERSICHERTER_korrigiert.csv") %>%
.[, GESCHLECHT := ifelse(GESCHLECHT == 1, "männlich", "weiblich")] %>%
.[, VERSSTATUS := ifelse(VERSSTATUS == 1,
"Mitglied",
ifelse(VERSSTATUS == 3,
"Familienversichert",
"Rentner"))] %>%
.[, BERUF := as.factor(BERUF)] %>%
.[, BETREUER := as.logical(BETREUER)] %>%
.[, TODES_DATUM := ifelse(TODES_DATUM == 999912, NA, TODES_DATUM)] %>%
.[, TODES_DATUM := as.POSIXct(TODES_DATUM %>% as.character %>% paste0("01"), format = "%Y%m%d")]
}

# Datasets nur mit ersten Hospitalisierungen pro Patienten (dauert)
{
# copd
temp <- copd_90[, row.before := 1:nrow(copd_90)] %>%
.[order(copd_90[, BEGINN])] %>%
.[, id := 1:nrow(copd_90)]

first_entries <- vapply(unique(temp[, VERSID]),
FUN = function(X){
temp[VERSID == X,
id][1]
},
1)
copd_90_first_entries <- temp[id %in% first_entries]

# hf_ukhd
temp <- hf_ukhd_90[, row.before := 1:nrow(hf_ukhd_90)] %>%
.[order(hf_ukhd_90[, BEGINN])] %>%
.[, id := 1:nrow(hf_ukhd_90)]

first_entries <- vapply(unique(temp[, VERSID]),
FUN = function(X){
temp[VERSID == X,
id][1]
},
1)
hf_ukhd_90_first_entries <- temp[id %in% first_entries]

# mi_ukhd
temp <- mi_ukhd_90[, row.before := 1:nrow(mi_ukhd_90)] %>%
.[order(mi_ukhd_90[, BEGINN])] %>%
.[, id := 1:nrow(mi_ukhd_90)]

first_entries <- vapply(unique(temp[, VERSID]),
FUN = function(X){
temp[VERSID == X,
id][1]
},
1)
mi_ukhd_90_first_entries <- temp[id %in% first_entries]

# osteo
temp <- osteo_90[, row.before := 1:nrow(osteo_90)] %>%
.[order(osteo_90[, BEGINN])] %>%
.[, id := 1:nrow(osteo_90)]

first_entries <- vapply(unique(temp[, VERSID]),
FUN = function(X){
temp[VERSID == X,
id][1]
},
1)
osteo_90_first_entries <- temp[id %in% first_entries]

# t2dm
temp <- t2dm_90[, row.before := 1:nrow(t2dm_90)] %>%
.[order(t2dm_90[, BEGINN])] %>%
.[, id := 1:nrow(t2dm_90)]

first_entries <- vapply(unique(temp[, VERSID]),
FUN = function(X){
temp[VERSID == X,
id][1]
},
1)
t2dm_90_first_entries <- temp[id %in% first_entries]

# vhf
temp <- vhf_90[, row.before := 1:nrow(vhf_90)] %>%
.[order(vhf_90[, BEGINN])] %>%
.[, id := 1:nrow(vhf_90)]

first_entries <- vapply(unique(temp[, VERSID]),

```

```

        FUN = function(X){
          temp[VERSID == X,
                id][1]
        },
        1)
vhf_90_first_entries <- temp[id %in% first_entries]
}

#### Deskriptive Tabelle (Patienten) ####
## Anzahl Index-Fälle
df <- data.frame(COPD = nrow(copd) %>% as.character,
                 HF_UKHD = nrow(hf_ukhd) %>% as.character,
                 MI_UKHD = nrow(mi_ukhd) %>% as.character,
                 OSTEO = nrow(osteo) %>% as.character,
                 T2DM = nrow(t2dm) %>% as.character,
                 VHF = nrow(vhf) %>% as.character,
                 stringsAsFactors = FALSE)
row.names(df) <- "Anzahl Index-Fälle"

## Anzahl Patienten
df[2,] <- c(copd_90$VERSID %>% unique %>% length %>% as.character,
           hf_ukhd_90$VERSID %>% unique %>% length %>% as.character,
           mi_ukhd_90$VERSID %>% unique %>% length %>% as.character,
           osteo_90$VERSID %>% unique %>% length %>% as.character,
           t2dm_90$VERSID %>% unique %>% length %>% as.character,
           vhf_90$VERSID %>% unique %>% length %>% as.character)
row.names(df)[2] <- "Anzahl Patienten (ges.)"

## Anzahl Männer
{
  copd_90 <- merge(x = copd_90,
                  y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                  by.x = c("VERSID", "DATENSATZ_JAHR"),
                  by.y = c("VERSID", "DATENSATZJAHR"),
                  all.y = FALSE)
  hf_ukhd_90 <- merge(x = hf_ukhd_90,
                    y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                    by.x = c("VERSID", "DATENSATZ_JAHR"),
                    by.y = c("VERSID", "DATENSATZJAHR"),
                    all.y = FALSE)
  mi_ukhd_90 <- merge(x = mi_ukhd_90,
                    y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                    by.x = c("VERSID", "DATENSATZ_JAHR"),
                    by.y = c("VERSID", "DATENSATZJAHR"),
                    all.y = FALSE)
  osteo_90 <- merge(x = osteo_90,
                  y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                  by.x = c("VERSID", "DATENSATZ_JAHR"),
                  by.y = c("VERSID", "DATENSATZJAHR"),
                  all.y = FALSE)
  t2dm_90 <- merge(x = t2dm_90,
                  y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                  by.x = c("VERSID", "DATENSATZ_JAHR"),
                  by.y = c("VERSID", "DATENSATZJAHR"),
                  all.y = FALSE)
  vhf_90 <- merge(x = vhf_90,
                 y = versicherter_data[,
                                     .(VERSID,
                                       GESCHLECHT,
                                       DATENSATZJAHR)],
                 by.x = c("VERSID", "DATENSATZ_JAHR"),
                 by.y = c("VERSID", "DATENSATZJAHR"),
                 all.y = FALSE)
}

df[3,] <- c(copd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           hf_ukhd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           mi_ukhd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           osteo_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           t2dm_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           vhf_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length) %>% as.character
row.names(df)[3] <- "Anzahl männlicher Patienten"

## Anzahl Männer (%)
df[4,] <- c(copd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           hf_ukhd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           mi_ukhd_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           osteo_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           t2dm_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length,
           vhf_90[GESCHLECHT == "männlich", VERSID] %>% unique %>% length)*100/df[2,] %>% as.numeric) %>%
  round(digits = 2) %>%
  as.character
row.names(df)[4] <- "Männlicher Anteil der Patienten (%)"

## Alter
# Berechne Alterlagemaße
df[5, ] <- c(paste0(copd[copd_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", copd[copd_90_first_entries[, row_before],
age] %>% sd %>% round(digits = 2), ")"),
           paste0(hf_ukhd[hf_ukhd_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", hf_ukhd[hf_ukhd_90_first_entries[,

```

```

row_before], age] %>% sd %>% round(digits = 2), ")"),
  paste0(mi_ukhd[mi_ukhd_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", mi_ukhd[mi_ukhd_90_first_entries[,
row_before], age] %>% sd %>% round(digits = 2), ")"),
  paste0(osteo[osteo_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", osteo[osteo_90_first_entries[,
row_before], age] %>% sd %>% round(digits = 2), ")"),
  paste0(t2dm[t2dm_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", t2dm[t2dm_90_first_entries[, row_before],
age] %>% sd %>% round(digits = 2), ")"),
  paste0(vhf[vhf_90_first_entries[, row_before], age] %>% mean %>% round(digits = 2), " (", vhf[vhf_90_first_entries[, row_before], age]
%>% sd %>% round(digits = 2), ")"))
row.names(df)[5] <- "Alter (mean (sd))"

## Anzahl vorheriger Hospitalisierungen im Jahr vor Indexfall
df[6, ] <- c(paste0(copd[copd_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (", copd[copd_90_first_entries[,
row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"),
  paste0(hf_ukhd[hf_ukhd_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (",
hf_ukhd[hf_ukhd_90_first_entries[, row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"),
  paste0(mi_ukhd[mi_ukhd_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (",
mi_ukhd[mi_ukhd_90_first_entries[, row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"),
  paste0(osteo[osteo_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (", osteo[osteo_90_first_entries[,
row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"),
  paste0(t2dm[t2dm_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (", t2dm[t2dm_90_first_entries[,
row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"),
  paste0(vhf[vhf_90_first_entries[, row_before], hosp_last_year] %>% mean %>% round(digits = 2), " (", vhf[vhf_90_first_entries[,
row_before], hosp_last_year] %>% sd %>% round(digits = 2), ")"))
row.names(df)[6] <- "Hospitalisierungen im Jahr vor Indexfall (mean (sd))"

#write.xlsx(df, file = "./descriptive_tabelle_patient.xlsx")

#### Deskriptive Tabelle (Fall) ####
## Anzahl Index-Fälle
df <- data.frame(COPD_Rehosp = sum(copd[, Rehosp_90_specific]) %>% as.character,
  COPD_keine_Rehosp = sum(1-copd[, Rehosp_90_specific]) %>% as.character,
  HF_UKHD_Rehosp = sum(hf_ukhd[, Rehosp_90_specific]) %>% as.character,
  HF_UKHD_keine_Rehosp = sum(1-hf_ukhd[, Rehosp_90_specific]) %>% as.character,
  MI_UKHD_Rehosp = sum(mi_ukhd[, Rehosp_90_specific]) %>% as.character,
  MI_UKHD_keine_Rehosp = sum(1-mi_ukhd[, Rehosp_90_specific]) %>% as.character,
  OSTEO_Rehosp = sum(osteo[, Rehosp_90_specific]) %>% as.character,
  OSTEO_keine_Rehosp = sum(1-osteo[, Rehosp_90_specific]) %>% as.character,
  T2DM_Rehosp = sum(t2dm[, Rehosp_90_specific]) %>% as.character,
  T2DM_keine_Rehosp = sum(1-t2dm[, Rehosp_90_specific]) %>% as.character,
  VHF_Rehosp = sum(vhf[, Rehosp_90_specific]) %>% as.character,
  VHF_keine_Rehosp = sum(1-vhf[, Rehosp_90_specific]) %>% as.character,
  stringsAsFactors = FALSE)
row.names(df) <- "Anzahl Index-Fälle"

## Anzahl Patienten
df[2,] <- c(copd_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  copd_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character,
  hf_ukhd_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  hf_ukhd_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character,
  mi_ukhd_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  mi_ukhd_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character,
  osteo_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  osteo_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character,
  t2dm_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  t2dm_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character,
  vhf_90[Rehosp_90_specific == TRUE, VERSID] %>% unique %>% length %>% as.character,
  vhf_90[Rehosp_90_specific == FALSE, VERSID] %>% unique %>% length %>% as.character)
row.names(df)[2] <- "Anzahl Patienten (ges.)"

## Anzahl Männer
{
  # copd_90 <- merge(x = copd_90,
  #                 y = versicherter_data[,
  #                   .(VERSID,
  #                     GESCHLECHT,
  #                     DATENSATZJAHR)],
  #                 by.x = c("VERSID", "DATENSATZ_JAHR"),
  #                 by.y = c("VERSID", "DATENSATZJAHR"),
  #                 all.y = FALSE)
  # hf_ukhd_90 <- merge(x = hf_ukhd_90,
  #                     y = versicherter_data[,
  #                       .(VERSID,
  #                         GESCHLECHT,
  #                         DATENSATZJAHR)],
  #                     by.x = c("VERSID", "DATENSATZ_JAHR"),
  #                     by.y = c("VERSID", "DATENSATZJAHR"),
  #                     all.y = FALSE)
  # mi_ukhd_90 <- merge(x = mi_ukhd_90,
  #                     y = versicherter_data[,
  #                       .(VERSID,
  #                         GESCHLECHT,
  #                         DATENSATZJAHR)],
  #                     by.x = c("VERSID", "DATENSATZ_JAHR"),
  #                     by.y = c("VERSID", "DATENSATZJAHR"),
  #                     all.y = FALSE)
  # osteo_90 <- merge(x = osteo_90,
  #                   y = versicherter_data[,
  #                     .(VERSID,
  #                       GESCHLECHT,
  #                       DATENSATZJAHR)],
  #                   by.x = c("VERSID", "DATENSATZ_JAHR"),
  #                   by.y = c("VERSID", "DATENSATZJAHR"),
  #                   all.y = FALSE)
  # t2dm_90 <- merge(x = t2dm_90,
  #                  y = versicherter_data[,
  #                    .(VERSID,
  #                      GESCHLECHT,
  #                      DATENSATZJAHR)],
  #                  by.x = c("VERSID", "DATENSATZ_JAHR"),
  #                  by.y = c("VERSID", "DATENSATZJAHR"),
  #                  all.y = FALSE)
  # vhf_90 <- merge(x = vhf_90,
  #                 y = versicherter_data[,
  #                   .(VERSID,
  #                     GESCHLECHT,

```

```

#                                     DATENSATZJAHR)],
#                                     by.x = c("VERSID", "DATENSATZ_JAHR"),
#                                     by.y = c("VERSID", "DATENSATZJAHR"),
#                                     all.y = FALSE)
)#auskommentiert. durchlaufen lassen, falls für Patienten hier drüber nicht gemacht

df[3,] <- c(copd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            copd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            hf_ukhd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            hf_ukhd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            mi_ukhd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            mi_ukhd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            osteo_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            osteo_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            t2dm_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            t2dm_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            vhf_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            vhf_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow)) %>% as.character
row.names(df)[3] <- "Anzahl männlicher Fälle"

## Anzahl Männer (%)
df[4,] <- c(copd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            copd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            hf_ukhd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            hf_ukhd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            mi_ukhd_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            mi_ukhd_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            osteo_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            osteo_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            t2dm_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            t2dm_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow,
            vhf_90[Rehosp_90_specific == TRUE & GESCHLECHT == "männlich"] %>% nrow,
            vhf_90[Rehosp_90_specific == FALSE & GESCHLECHT == "männlich"] %>% nrow)*100/df[1,] %>% as.numeric) %>%
round(digits = 2) %>%
as.character
row.names(df)[4] <- "Männlicher Anteil der Fälle (%)"

## Alter
# Berechne Alter
df[5, ] <- c(paste0(copd[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", copd[Rehosp_90_specific == TRUE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(copd[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", copd[Rehosp_90_specific == FALSE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(hf_ukhd[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", hf_ukhd[Rehosp_90_specific == TRUE, age] %>% sd
%>% round(digits = 2), ")"),
            paste0(hf_ukhd[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", hf_ukhd[Rehosp_90_specific == FALSE, age] %>% sd
%>% round(digits = 2), ")"),
            paste0(mi_ukhd[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", mi_ukhd[Rehosp_90_specific == TRUE, age] %>% sd
%>% round(digits = 2), ")"),
            paste0(mi_ukhd[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", mi_ukhd[Rehosp_90_specific == FALSE, age] %>% sd
%>% round(digits = 2), ")"),
            paste0(osteo[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", osteo[Rehosp_90_specific == TRUE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(osteo[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", osteo[Rehosp_90_specific == FALSE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(t2dm[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", t2dm[Rehosp_90_specific == TRUE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(t2dm[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", t2dm[Rehosp_90_specific == FALSE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(vhf[Rehosp_90_specific == TRUE, age] %>% mean %>% round(digits = 2), " (", vhf[Rehosp_90_specific == TRUE, age] %>% sd %>%
round(digits = 2), ")"),
            paste0(vhf[Rehosp_90_specific == FALSE, age] %>% mean %>% round(digits = 2), " (", vhf[Rehosp_90_specific == FALSE, age] %>% sd %>%
round(digits = 2), ")"))
row.names(df)[5] <- "Alter (mean (sd))"

## Anzahl vorheriger Hospitalisierungen im Jahr vor Indexfall
df[6, ] <- c(paste0(copd[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", copd[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(copd[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", copd[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(hf_ukhd[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", hf_ukhd[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(hf_ukhd[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", hf_ukhd[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(mi_ukhd[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", mi_ukhd[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(mi_ukhd[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", mi_ukhd[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(osteo[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", osteo[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(osteo[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", osteo[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(t2dm[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", t2dm[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(t2dm[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", t2dm[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(vhf[Rehosp_90_specific == TRUE, hosp_last_year] %>% mean %>% round(digits = 2), " (", vhf[Rehosp_90_specific == TRUE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"),
            paste0(vhf[Rehosp_90_specific == FALSE, hosp_last_year] %>% mean %>% round(digits = 2), " (", vhf[Rehosp_90_specific == FALSE,
hosp_last_year] %>% sd %>% round(digits = 2), ")"))
row.names(df)[6] <- "Hospitalisierungen im Jahr vor Indexfall (mean (sd))"

#write.xlsx(df, file = "../descriptive_tabelle_fall.xlsx")

#### Komorbiditäten (Patienten) ####
df <- data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Komorbidität)` =
paste0(copd[copd_90_first_entries[, row_before], 4] %>% sum, ", ", ((copd[copd_90_first_entries[, row_before], 4] %>%
sum)/nrow(copd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`HF UKHD (Anzahl Patienten (abs. + rel.) mit Komorbidität)` =
paste0(hf_ukhd[hf_ukhd_90_first_entries[, row_before], 4] %>% sum, ", ", ((hf_ukhd[hf_ukhd_90_first_entries[, row_before], 4] %>%
sum)/nrow(hf_ukhd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`MI UKHD (Anzahl Patienten (abs. + rel.) mit Komorbidität)` =
paste0(mi_ukhd[mi_ukhd_90_first_entries[, row_before], 4] %>% sum, ", ", ((mi_ukhd[mi_ukhd_90_first_entries[, row_before], 4] %>%
sum)/nrow(mi_ukhd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`OSTEO (Anzahl Patienten (abs. + rel.) mit Komorbidität)` =

```



```

= "< 0.01"))),
  `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Komorbidität ohne Rehosp.)` =
    paste0(mi_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((mi_ukhd[Rehosp_90_specific ==
FALSE, eval(as.name(names(copd)[i]))] %>% sum)/nrow(mi_ukhd[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2),
no = "< 0.01"))),
  `OSTEO (Anzahl Patienten (abs. + rel.) mit Komorbidität und Rehosp.)` =
    paste0(osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((osteo[Rehosp_90_specific == TRUE,
eval(as.name(names(copd)[i]))] %>% sum)/nrow(osteo[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "<
0.01"))),
  `OSTEO (Anzahl Patienten (abs. + rel.) mit Komorbidität ohne Rehosp.)` =
    paste0(osteo[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((osteo[Rehosp_90_specific ==
FALSE, eval(as.name(names(copd)[i]))] %>% sum)/nrow(osteo[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no =
"< 0.01"))),
  `T2DM (Anzahl Patienten (abs. + rel.) mit Komorbidität und Rehosp.)` =
    paste0(t2dm[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((t2dm[Rehosp_90_specific == TRUE,
eval(as.name(names(copd)[i]))] %>% sum)/nrow(t2dm[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "<
0.01"))),
  `T2DM (Anzahl Patienten (abs. + rel.) mit Komorbidität ohne Rehosp.)` =
    paste0(t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((t2dm[Rehosp_90_specific == FALSE,
eval(as.name(names(copd)[i]))] %>% sum)/nrow(t2dm[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "<
0.01"))),
  `VHF (Anzahl Patienten (abs. + rel.) mit Komorbidität und Rehosp.)` =
    paste0(vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((vhf[Rehosp_90_specific == TRUE,
eval(as.name(names(copd)[i]))] %>% sum)/nrow(vhf[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "<
0.01"))),
  `VHF (Anzahl Patienten (abs. + rel.) mit Komorbidität ohne Rehosp.)` =
    paste0(vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] %>% sum, ", ", ((vhf[Rehosp_90_specific == FALSE,
eval(as.name(names(copd)[i]))] %>% sum)/nrow(vhf[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "<
0.01"))),
  stringsAsFactors = FALSE)
}

row.names(df) <- names(copd)[4:34]

#write.xlsx(df, file = "./comorbidities_fall.xlsx")

#### START-Kriterien (Patienten) ####
df <- data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((copd[copd_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((copd[copd_90_first_entries[, row_before], 84] ==
TRUE) %>% sum)/nrow(copd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((hf_ukhd[hf_ukhd_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((hf_ukhd[hf_ukhd_90_first_entries[,
row_before], 84] == TRUE) %>% sum)/nrow(hf_ukhd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((mi_ukhd[mi_ukhd_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((mi_ukhd[mi_ukhd_90_first_entries[,
row_before], 84] == TRUE) %>% sum)/nrow(mi_ukhd_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((osteo[osteo_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((osteo[osteo_90_first_entries[, row_before], 84]
== TRUE) %>% sum)/nrow(osteo_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((t2dm[t2dm_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((t2dm[t2dm_90_first_entries[, row_before], 84] ==
TRUE) %>% sum)/nrow(t2dm_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((vhf[vhf_90_first_entries[, row_before], 84] == TRUE) %>% sum, ", ", (((vhf[vhf_90_first_entries[, row_before], 84] ==
TRUE) %>% sum)/nrow(vhf_90_first_entries)) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  stringsAsFactors = FALSE)

for(i in 85:98){
  df <- rbind(df,
    data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((copd[copd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((copd[copd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(copd_90_first_entries)) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01"))),
      `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((hf_ukhd[hf_ukhd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((hf_ukhd[hf_ukhd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(hf_ukhd_90_first_entries)) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
      `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((mi_ukhd[mi_ukhd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((mi_ukhd[mi_ukhd_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(mi_ukhd_90_first_entries)) %>% {ifelse(test
= . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
      `OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((osteo[osteo_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((osteo[osteo_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(osteo_90_first_entries)) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01"))),
      `T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((t2dm[t2dm_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((t2dm[t2dm_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(t2dm_90_first_entries)) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01"))),
      `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
      paste0((vhf[vhf_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      (((vhf[vhf_90_first_entries[, row_before], eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(vhf_90_first_entries)) %>% {ifelse(test = . > 0.01,
yes = round(., digits = 2), no = "< 0.01"))),
      stringsAsFactors = FALSE)
    )
}

row.names(df) <- names(copd)[84:98]

#write.xlsx(df, file = "./START_patienten.xlsx")

#### START-Kriterien (Fälle) ####
df <- data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
  paste0((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `COPD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
  paste0((copd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((copd[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
  paste0((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
  paste0((hf_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((hf_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01"))),
  stringsAsFactors = FALSE)
}

```

```

sum)/nrow(hf_ukhd[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((mi_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((mi_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((osteo[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((osteo[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((osteo[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((osteo[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((t2dm[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((t2dm[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`VHF (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((vhf[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((vhf[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
`VHF (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((vhf[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((vhf[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == FALSE])) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}},
stringsAsFactors = FALSE)

## Ermittle genaue Werte für Tabelle in Paper
# df2 <- data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `COPD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((copd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((copd[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((hf_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((hf_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE)
%>% sum)/nrow(hf_ukhd[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((mi_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((mi_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE)
%>% sum)/nrow(mi_ukhd[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# `OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((osteo[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((osteo[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((osteo[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((osteo[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# `T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((t2dm[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((t2dm[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
# paste0((vhf[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ", (((vhf[Rehosp_90_specific == TRUE, 84] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE]))*100 %>% round(., digits = 2)),
# `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
# paste0((vhf[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum, ", ", (((vhf[Rehosp_90_specific == FALSE, 84] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
stringsAsFactors = FALSE)

##
## Struktur:
## Zeile 1: ermittle absolute Anzahl an Patienten mit Rehospitalisierung mit Kriterium
## Zeile 2: ermittle relative Anzahl an Patienten mit Kriterium von allen Patienten mit Rehospitalisierung
## Zeile 3: siehe Zeile 2
## Zeile 4: ermittle relative Anzahl an Patienten mit Kriterium von allen Patienten ohne Rehospitalisierung und subtrahiere von Zeile 3
## Zeile 5: nimm die Differenz aus Zeile 3/4 und schreibe + oder - davor und runde auf 2 signifikante Ziffern
## Zeile 6: Schreibe finales Prozentzeichen
##
df2 <- data.frame(AMI = paste0((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ",
{{{(mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum}/nrow(mi_ukhd[Rehosp_90_specific == TRUE])}*100} %>%
signif(., digits = 3), "% ("),
{{{(((mi_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE]))*100} -
{{{(mi_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum}/nrow(mi_ukhd[Rehosp_90_specific == FALSE])}*100}} %>%
{ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
"%)"",
COPD = paste0((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ",
{{{(copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum}/nrow(copd[Rehosp_90_specific == TRUE])}*100} %>% signif(.,
digits = 3), "% ("),
{{{(((copd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum)/nrow(copd[Rehosp_90_specific == TRUE]))*100} -
{{{(copd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum}/nrow(copd[Rehosp_90_specific == FALSE])}*100}} %>%
{ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
"%)"",
DM = paste0((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ",
{{{(t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum}/nrow(t2dm[Rehosp_90_specific == TRUE])}*100} %>% signif(., digits
= 3), "% ("),
{{{(((t2dm[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum)/nrow(t2dm[Rehosp_90_specific == TRUE]))*100} -
{{{(t2dm[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum}/nrow(t2dm[Rehosp_90_specific == FALSE])}*100}} %>%
{ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
"%)"",
HF = paste0((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ",
{{{(hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum}/nrow(hf_ukhd[Rehosp_90_specific == TRUE])}*100} %>%
signif(., digits = 3), "% ("),
{{{(((hf_ukhd[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE]))*100} -
{{{(hf_ukhd[Rehosp_90_specific == FALSE, 84] == TRUE) %>% sum}/nrow(hf_ukhd[Rehosp_90_specific == FALSE])}*100}} %>%
{ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
"%)"",
OS = paste0((osteo[Rehosp_90_specific == TRUE, 84] == TRUE) %>% sum, ", ",

```



```

# paste0((t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
((t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(t2dm[Rehosp_90_specific == FALSE])*100 %>% round(.,
digits = 2)),
# `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ", ((vhf[Rehosp_90_specific
== TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(vhf[Rehosp_90_specific == TRUE])*100 %>% round(., digits = 2)),
# `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(vhf[Rehosp_90_specific == FALSE])*100 %>% round(., digits
= 2)),
# stringsAsFactors = FALSE)

df2 <- rbind(df2,
  data.frame(AMI = paste0((mi_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((mi_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((mi_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE])*100)} -
    {(((mi_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(mi_ukhd[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
    COPD = paste0((copd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((copd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((copd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == TRUE])*100)} -
    {(((copd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(copd[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
    DM = paste0((t2dm[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((t2dm[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((t2dm[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == TRUE])*100)} -
    {(((t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(t2dm[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
    HF = paste0((hf_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((hf_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((hf_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE])*100)} -
    {(((hf_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(hf_ukhd[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
    OS = paste0((osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == TRUE])*100)} -
    {(((osteo[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
    `S/AF` = paste0((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
    {((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE])*100} %>% signif(., digits = 3), "% ("",
    {(((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE])*100)} -
    {(((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == FALSE])*100)} %>%
    {ifelse(test = . > 0, yes = paste0("+", signif(., digits = 2)), no = signif(., digits = 2))},
    "%)"),
  )
)
}

row.names(df) <- names(copd)[84:98]
row.names(df2) <- names(copd)[84:98]

#write.xlsx(df, file = "./START_fall.xlsx")
#write.xlsx(df2, file = "./START_fall_S5.xlsx")

## Cancel never selected Criteria
df2 <- df2[-15,]

#### STOPP-Kriterien (Patienten) ####
df <- data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((copd[copd_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((copd[copd_90_first_entries[, row_before], 35] ==
TRUE) %>% sum)/nrow(copd_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  `HF_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((hf_ukhd[hf_ukhd_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((hf_ukhd[hf_ukhd_90_first_entries[,
row_before], 35] == TRUE) %>% sum)/nrow(hf_ukhd_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  `MI_UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((mi_ukhd[mi_ukhd_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((mi_ukhd[mi_ukhd_90_first_entries[,
row_before], 35] == TRUE) %>% sum)/nrow(mi_ukhd_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  `OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((osteo[osteo_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((osteo[osteo_90_first_entries[, row_before], 35]
== TRUE) %>% sum)/nrow(osteo_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  `T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((t2dm[t2dm_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((t2dm[t2dm_90_first_entries[, row_before], 35] ==
TRUE) %>% sum)/nrow(t2dm_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  `VHF (Anzahl Patienten (abs. + rel.) mit Kriterium)` =
  paste0((vhf[vhf_90_first_entries[, row_before], 35] == TRUE) %>% sum, ", ", ((vhf[vhf_90_first_entries[, row_before], 35] ==
TRUE) %>% sum)/nrow(vhf_90_first_entries) %>% {ifelse(test = . > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
  stringsAsFactors = FALSE)

for(i in 36:83){
  df <- rbind(df,

```



```

sum)/nrow(vhf[Rehosp_90_specific == FALSE]))*100 %>% round(., digits = 2)),
# stringsAsFactors = FALSE)
##
## Struktur:
## Zeile 1: ermittle absolute Anzahl an Patienten mit Rehospitalisierung mit Kriterium
## Zeile 2: ermittle relative Anzahl an Patienten mit Kriterium von allen Patienten mit Rehospitalisierung
## Zeile 3: siehe Zeile 2
## Zeile 4: ermittle relative Anzahl an Patienten mit Kriterium von allen Patienten ohne Rehospitalisierung und subtrahiere von Zeile 3
## Zeile 5: nimm die Differenz aus Zeile 3/4 und schreibe + oder - davor und runde auf 2 signifikante Ziffern
## Zeile 6: Schreibe finales Prozentzeichen
##
df2 <- data.frame(AMI = paste0((mi_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(mi_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(mi_ukhd[Rehosp_90_specific == TRUE])}*100) %>%
{round(x = signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((mi_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE])}*100)} -
{{{(mi_ukhd[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(mi_ukhd[Rehosp_90_specific == FALSE])}*100}}) %>%
%>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " ),
COPD = paste0((copd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(copd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(copd[Rehosp_90_specific == TRUE])}*100) %>% {round(x =
= signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((copd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(copd[Rehosp_90_specific == TRUE])}*100)} -
{{{(copd[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(copd[Rehosp_90_specific == FALSE])}*100}}) %>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " ),
DM = paste0((t2dm[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(t2dm[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(t2dm[Rehosp_90_specific == TRUE])}*100) %>% {round(x =
signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((t2dm[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(t2dm[Rehosp_90_specific == TRUE])}*100)} -
{{{(t2dm[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(t2dm[Rehosp_90_specific == FALSE])}*100}}) %>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " ),
HF = paste0((hf_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(hf_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(hf_ukhd[Rehosp_90_specific == TRUE])}*100) %>%
{round(x = signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((hf_ukhd[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE])}*100)} -
{{{(hf_ukhd[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(hf_ukhd[Rehosp_90_specific == FALSE])}*100}}) %>%
%>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " ),
OS = paste0((osteo[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(osteo[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(osteo[Rehosp_90_specific == TRUE])}*100) %>% {round(x =
= signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((osteo[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(osteo[Rehosp_90_specific == TRUE])}*100)} -
{{{(osteo[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(osteo[Rehosp_90_specific == FALSE])}*100}}) %>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " ),
`S/AF` = paste0((vhf[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum, ", ",
{{{(vhf[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum}/nrow(vhf[Rehosp_90_specific == TRUE])}*100) %>% {round(x =
= signif(., digits = 3), digits = 2)}, "% (" ,
{{{(((vhf[Rehosp_90_specific == TRUE, 35] == TRUE) %>% sum)/nrow(vhf[Rehosp_90_specific == TRUE])}*100)} -
{{{(vhf[Rehosp_90_specific == FALSE, 35] == TRUE) %>% sum}/nrow(vhf[Rehosp_90_specific == FALSE])}*100}}) %>%
{ifelse(test = . > 0, yes = paste0("+", round(x = signif(., digits = 2), digits = 2)), no = round(x = signif(.,
digits = 2), digits = 2))},
"% " )
)
for(i in 36:83){
df <- rbind(df,
data.frame(`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((copd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ", ((copd[Rehosp_90_specific
== TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(copd[Rehosp_90_specific == TRUE]) %>% {ifelse(test = . > 0.01, yes = round(., digits
= 2), no = "< 0.01")}),
`COPD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((copd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((copd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(copd[Rehosp_90_specific == FALSE]) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01")}),
`HF UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((hf_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((hf_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(hf_ukhd[Rehosp_90_specific == TRUE]) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`HF UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((hf_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((hf_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(hf_ukhd[Rehosp_90_specific == FALSE]) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`MI UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((mi_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((mi_ukhd[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(mi_ukhd[Rehosp_90_specific == TRUE]) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`MI UKHD (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((mi_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((mi_ukhd[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(mi_ukhd[Rehosp_90_specific == FALSE]) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((osteo[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(osteo[Rehosp_90_specific == TRUE]) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01")}),
`OSTEO (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((osteo[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((osteo[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(osteo[Rehosp_90_specific == FALSE]) %>% {ifelse(test =
. > 0.01, yes = round(., digits = 2), no = "< 0.01")}),
`T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium und Rehosp.)` =
paste0((t2dm[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ", ((t2dm[Rehosp_90_specific
== TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(t2dm[Rehosp_90_specific == TRUE]) %>% {ifelse(test = . > 0.01, yes = round(., digits
= 2), no = "< 0.01")}),
`T2DM (Anzahl Patienten (abs. + rel.) mit Kriterium ohne Rehosp.)` =
paste0((t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((t2dm[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(t2dm[Rehosp_90_specific == FALSE]) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01")}),
`S/AF` = paste0((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(vhf[Rehosp_90_specific == TRUE]) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01")}),
`S/AF` = paste0((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
(((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum)/nrow(vhf[Rehosp_90_specific == FALSE]) %>% {ifelse(test = . >
0.01, yes = round(., digits = 2), no = "< 0.01")})
)
}

```



```

sum)/nrow(osteo[Rehosp_90_specific == TRUE])*100}} -
      {(((osteo[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(osteo[Rehosp_90_specific == FALSE])*100)}} %>%
      {ifelse(test = . > 0, yes = paste0("+", round(signif(., digits = 2), digits = 2)), no = round(signif(.,
digits = 2), digits = 2))),
      "%}"),
      `S/AF` = paste0((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>% sum, ", ",
      {(((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE])*100)} %>% {round(signif(., digits = 3), digits = 2)}, "% (",
      {(((vhf[Rehosp_90_specific == TRUE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == TRUE])*100}} -
      {(((vhf[Rehosp_90_specific == FALSE, eval(as.name(names(copd)[i]))] == TRUE) %>%
sum)/nrow(vhf[Rehosp_90_specific == FALSE])*100)}} %>%
      {ifelse(test = . > 0, yes = paste0("+", round(signif(., digits = 2), digits = 2)), no = round(signif(.,
digits = 2), digits = 2))),
      "%} ")
    )
  )
}

row.names(df) <- names(copd)[35:83]
row.names(df2) <- names(copd)[35:83]

#write.xlsx(df, file = "./STOPP_fall.xlsx")
#write.xlsx(df2, file = "./STOPP_fall_S5.xlsx")

#####
#####
##### Models #####
#####

### Prepare final matrix ###
finalmatrix <- data.frame(matrix(ncol = 3*6, nrow = length(names(copd))))

rownames(finalmatrix) <- c("Intercept", "sex = female", "Age*", "Hospitalisations last year", names(copd)[4:98])
colnames(finalmatrix) <- c("GLM COPD", "pValue COPD", "LASSO COPD",
  "GLM HI", "pValue HI", "LASSO HI",
  "GLM AMI", "pValue AMI", "LASSO AMI",
  "GLM OS", "pValue OS", "LASSO OS",
  "GLM DM", "pValue DM", "LASSO DM",
  "GLM S/AF", "pValue S/AF", "LASSO S/AF")

increasematrix <- data.frame(matrix(ncol = 2, nrow = 6))
colnames(increasematrix) <- c("all_zero", "all_one")
rownames(increasematrix) <- c("COPD", "HI", "AMI", "OS", "DM", "S/AF")

### GLM, LASSO Koeffizienten Tabelle ###
# copd
# filter for all criteria with at least 1% TRUE
filternumbers <- vector(mode = "numeric")
for(i in 4:98){

  if(!sum(copd[,eval(as.name(names(copd)[i]))] == TRUE) > floor(nrow(copd)*0.01)){

    filternumbers <- c(filternumbers, i)

  }

}

copd_glm <- glm(Rehosp_90_specific ~ ., data = as.data.frame(copd)[,-filternumbers], family = "binomial")

df <- summary(copd_glm)$coefficients
df <- as.data.frame(df)
rownames(df)[rownames(df) == "age"] <- "Age*"
rownames(df)[rownames(df) == "hosp_last_year"] <- "Hospitalisations last year*"
rownames(df) <- str_remove(rownames(df), pattern = "TRUE")

df_final <- df[, c(1,4)]
df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

# LASSO
X <- model.matrix(Rehosp_90_specific ~ .,
  data = copd[, -1][, -filternumbers])

set.seed(14102020)
glm_pen_cv <- cv.glmnet(y = copd[, Rehosp_90_specific], x = X,
  family = "binomial",
  nfolds = 5,
  alpha = 1,
  type.measure = "auc")

## Fit auf alle Daten mit berechneten Lambdas
glm_pen <- glmnet(y = copd[, Rehosp_90_specific], x = X,
  family = "binomial",
  alpha = 1,
  lambda = glm_pen_cv$lambda)

coef_glmnet <- coef(glm_pen)
opt.glmnet <- match(glm_pen_cv$lambda.min, glm_pen_cv$lambda)

## Coefficients mit optimalem Lambda
coef.glmnet.opt <- coef_glmnet[,opt.glmnet]

names(coef.glmnet.opt) <- str_remove(names(coef.glmnet.opt), pattern = "TRUE")
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "age"] <- "Age*"
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "hosp_last_year"] <- "Hospitalisations last year*"

df_final$GLM <- df_final$GLM %>% round(digits = 3)
df_final$LASSO <- coef.glmnet.opt[names(coef.glmnet.opt) %in% rownames(df_final)] %>% round(digits = 3)

rownames(df_final)[1] <- "Intercept"

```

```

rownames(df_final)[2] <- "sex = female"

# Fill finalmatrix with coefficients
finalmatrix[rownames(finalmatrix) %in% rownames(df_final), 1:3] <- df_final

# Get increase for all START/STOPP = 0 vs. = 1
X_inc <- rbind(X[1,], X[1,])
X_inc[1,28:63] <- 0
X_inc[2,28:63] <- 1
# first row all 0, second row all 1

preds_inc <- predict(
  glm_pen,
  X_inc,
  glm_pen_cv$lambda[opt.glmnet],
  type = "response"
)

increasematrix[1,] <- preds_inc

#####

## plot coefficients path vs. lambda
par(cex=1.7,mar=c(5,5,3.5,3))
plot(glm_pen$lambda,coef_glmnet[2,],ylim=c(-0.5,0.5),type="l",
     lwd=2,ylab=expression(hat(beta[j])),xlab=expression(lambda))
abline(h=0,lty=3,lwd=2,col="grey")
for(i in 3:nrow(coef_glmnet)){
  lines(glm_pen$lambda,coef_glmnet[i,],lwd=2)
}
axis(3,at=glm_pen$lambda[opt.glmnet],label=substitute(paste(lambda["opt"]," = ",nn), list(nn=glm_pen$lambda[opt.glmnet])))
abline(v=glm_pen$lambda[opt.glmnet],lty=2,lwd=2,col=2)

## Alternative ggplot
ggdf <- data.frame(lambda = glm_pen$lambda,
                  coefficient = coef_glmnet[2,],
                  variable = rownames(coef_glmnet)[2])
for(i in 3:nrow(coef_glmnet)){
  ggdf <- rbind(ggdf,
              data.frame(lambda = glm_pen$lambda,
                        coefficient = coef_glmnet[i,],
                        variable = rownames(coef_glmnet)[i]))
}

# saveRDS(ggdf, file = "./modelling_copd_90/lasso_coeff_plot.rds")
# saveRDS(list(glm_pen, opt.glmnet), file = "./modelling_copd_90/lasso_coeff_plot_misc.rds")

ggplot(ggdf, aes(x = lambda, y = coefficient, color = variable)) +
  geom_vline(xintercept = glm_pen$lambda[opt.glmnet], linetype = "dashed") +
  geom_line() +
  theme(legend.position = "none") +
  ylab(expression(hat(beta[j]))) +
  xlab(expression(lambda)) +
  annotate("text", y = 0.26, x = glm_pen$lambda[opt.glmnet] + 0.0075,
         label = substitute(
           paste(lambda[opt], " = ", nn),
           list(nn = round(glm_pen$lambda[opt.glmnet], digits = 4))
         ))
) +
  ggtitle("LASSO coefficients (COPD)")

#####

# hf_ukhd
# filter for all criteria with at least 1% TRUE
filternumbers <- vector(mode = "numeric")
for(i in 4:98){
  if(!sum(hf_ukhd[,eval(as.name(names(hf_ukhd)[i]))]) == TRUE) > floor(nrow(hf_ukhd)*0.01)){
    filternumbers <- c(filternumbers, i)
  }
}

hf_ukhd_glm <- glm(Rehosp_90_specific ~ ., data = as.data.frame(hf_ukhd)[,-filternumbers], family = "binomial")

df <- summary(hf_ukhd_glm)$coefficients
df <- as.data.frame(df)
rownames(df)[rownames(df) == "age"] <- "Age*"
rownames(df)[rownames(df) == "hosp_last_year"] <- "Hospitalisations last year*"
rownames(df) <- str_remove(rownames(df), pattern = "TRUE")

df_final <- df[, c(1,4)]
df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

# LASSO
X <- model.matrix(Rehosp_90_specific ~ .,
                data = hf_ukhd[, -1][, -filternumbers])

set.seed(14102020)
glm_pen_cv <- cv.glmnet(y = hf_ukhd[, Rehosp_90_specific], x = X,
                      family = "binomial",

```

```

        nfolds = 5,
        alpha = 1,
        type.measure = "auc")

## Fit auf alle Daten mit berechneten Lambdas
glm_pen <- glmnet(y = hf_ukhd[, Rehosp_90_specific], x = X,
                family = "binomial",
                alpha = 1,
                lambda = glm_pen_cv$lambda)

coef_glmnet <- coef(glm_pen)
opt.glmnet <- match(glm_pen_cv$lambda.min, glm_pen_cv$lambda)

## Coefficients mit optimalem Lambda
coef.glmnet.opt <- coef_glmnet[,opt.glmnet]

names(coef.glmnet.opt) <- str_remove(names(coef.glmnet.opt), pattern = "TRUE")
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "age"] <- "Age*"
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "hosp_last_year"] <- "Hospitalisations last year*"

df_final$GLM <- df_final$GLM %>% round(digits = 3)
df_final$LASSO <- coef.glmnet.opt[names(coef.glmnet.opt) %in% rownames(df_final)] %>% round(digits = 3)

rownames(df_final)[1] <- "Intercept"
rownames(df_final)[2] <- "sex = female"

# Fill finalmatrix with coefficients
finalmatrix[rownames(finalmatrix) %in% rownames(df_final), 4:6] <- df_final

# Get increase for all START/STOPP = 0 vs. = 1
X_inc <- rbind(X[1,], X[1,])
X_inc[1,27:63] <- 0
X_inc[2,27:63] <- 1
# first row all 0, second row all 1

preds_inc <- predict(
  glm_pen,
  X_inc,
  glm_pen_cv$lambda[opt.glmnet],
  type = "response"
)

increasematrix[2,] <- preds_inc

## Alternative ggplot
ggdf <- data.frame(lambda = glm_pen$lambda,
                  coefficient = coef_glmnet[2,],
                  variable = rownames(coef_glmnet)[2])
for(i in 3:nrow(coef_glmnet)){
  ggdf <- rbind(ggdf,
              data.frame(lambda = glm_pen$lambda,
                        coefficient = coef_glmnet[i,],
                        variable = rownames(coef_glmnet)[i]))
}

# saveRDS(ggdf, file = "./modelling_hf_ukhd_90/lasso_coeff_plot.rds")
# saveRDS(list(glm_pen, opt.glmnet), file = "./modelling_hf_ukhd_90/lasso_coeff_plot_misc.rds")

ggplot(ggdf, aes(x = lambda, y = coefficient, color = variable)) +
  geom_vline(xintercept = glm_pen$lambda[opt.glmnet], linetype = "dashed") +
  geom_line() +
  theme(legend.position = "none") +
  ylab(expression(hat(beta[j]))) +
  xlab(expression(lambda)) +
  annotate("text", y = 0.26, x = glm_pen$lambda[opt.glmnet] + 0.0035,
         label = substitute(
           paste(lambda[opt], " = ", nn),
           list(nn = round(glm_pen$lambda[opt.glmnet], digits = 4))
         )
  ) +
  ggtitle("LASSO coefficients (HF UKHD)")

# mi_ukhd
# filter for all criteria with at least 1% TRUE
filternumbers <- vector(mode = "numeric")
for(i in 4:98){
  if(!sum(mi_ukhd[,eval(as.name(names(mi_ukhd)[i]))] == TRUE) > floor(nrow(mi_ukhd)*0.01)){
    filternumbers <- c(filternumbers, i)
  }
}

mi_ukhd_glm <- glm(Rehosp_90_specific ~ ., data = as.data.frame(mi_ukhd)[,-filternumbers], family = "binomial")

df <- summary(mi_ukhd_glm)$coefficients
df <- as.data.frame(df)
rownames(df)[rownames(df) == "age"] <- "Age*"
rownames(df)[rownames(df) == "hosp_last_year"] <- "Hospitalisations last year*"
rownames(df) <- str_remove(rownames(df), pattern = "TRUE")

df_final <- df[, c(1,4)]
df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

# LASSO
X <- model.matrix(Rehosp_90_specific ~ .,
                 data = mi_ukhd)[, -1][, -filternumbers]

set.seed(14102020)
glm_pen_cv <- cv.glmnet(y = mi_ukhd[, Rehosp_90_specific], x = X,

```

```

        family = "binomial",
        nfolds = 5,
        alpha = 1,
        type.measure = "auc")

## Fit auf alle Daten mit berechneten Lambdas
glm_pen <- glmnet(y = mi_ukhd[, Rehosp_90_specific], x = X,
                 family = "binomial",
                 alpha = 1,
                 lambda = glm_pen_cv$lambda)

coef_glmnet <- coef(glm_pen)
opt.glmnet <- match(glm_pen_cv$lambda.min, glm_pen_cv$lambda)

## Coefficients mit optimalem Lambda
coef.glmnet.opt <- coef_glmnet[,opt.glmnet]

names(coef.glmnet.opt) <- str_remove(names(coef.glmnet.opt), pattern = "TRUE")
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "age"] <- "Age*"
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "hosp_last_year"] <- "Hospitalisations last year*"

df_final$GLM <- df_final$GLM %>% round(digits = 3)
df_final$LASSO <- coef.glmnet.opt[names(coef.glmnet.opt) %in% rownames(df_final)] %>% round(digits = 3)

rownames(df_final)[1] <- "Intercept"
rownames(df_final)[2] <- "sex = female"

# Fill finalmatrix with coefficients
finalmatrix[rownames(finalmatrix) %in% rownames(df_final), 7:9] <- df_final

# Get increase for all START/STOPP = 0 vs. = 1
X_inc <- rbind(X[1,], X[1,])
X_inc[1,26:60] <- 0
X_inc[2,26:60] <- 1
# first row all 0, second row all 1

preds_inc <- predict(
  glm_pen,
  X_inc,
  glm_pen_cv$lambda[opt.glmnet],
  type = "response"
)

increasematrix[3,] <- preds_inc

## Alternative ggplot
ggdf <- data.frame(lambda = glm_pen$lambda,
                  coefficient = coef_glmnet[2,],
                  variable = rownames(coef_glmnet)[2])
for(i in 3:nrow(coef_glmnet)){
  ggdf <- rbind(ggdf,
              data.frame(lambda = glm_pen$lambda,
                        coefficient = coef_glmnet[i,],
                        variable = rownames(coef_glmnet)[i]))
}

# saveRDS(ggdf, file = "./modelling_mi_ukhd_90/lasso_coeff_plot.rds")
# saveRDS(list(glm_pen, opt.glmnet), file = "./modelling_mi_ukhd_90/lasso_coeff_plot_misc.rds")

ggplot(ggdf, aes(x = lambda, y = coefficient, color = variable)) +
  geom_vline(xintercept = glm_pen$lambda[opt.glmnet], linetype = "dashed") +
  geom_line() +
  theme(legend.position = "none") +
  ylab(expression(hat(beta)[j])) +
  xlab(expression(lambda)) +
  annotate("text", y = 0.26, x = glm_pen$lambda[opt.glmnet] + 0.0035,
         label = substitute(
           paste(lambda[opt], " = ", nn),
           list(nn = round(glm_pen$lambda[opt.glmnet], digits = 4))
         )
  ) +
  ggtitle("LASSO coefficients (MI UKHD)")

# osteo
# filter for all criteria with at least 1% TRUE
filternumbers <- vector(mode = "numeric")
for(i in 4:98){
  if(!sum(osteo[,eval(as.name(names(osteo)[i]))] == TRUE) > floor(nrow(osteo)*0.01)){
    filternumbers <- c(filternumbers, i)
  }
}

osteo_glm <- glm(Rehosp_90_specific ~ ., data = as.data.frame(osteo)[,-filternumbers], family = "binomial")

df <- summary(osteo_glm)$coefficients
df <- as.data.frame(df)
rownames(df)[rownames(df) == "age"] <- "Age*"
rownames(df)[rownames(df) == "hosp_last_year"] <- "Hospitalisations last year*"
rownames(df) <- str_remove(rownames(df), pattern = "TRUE")

df_final <- df[, c(1,4)]
df_final$Pr(>|z|) <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

# LASSO
X <- model.matrix(Rehosp_90_specific ~ .,
                 data = osteo)[, -1][, -filternumbers]

set.seed(14102020)

```



```

set.seed(14102020)
glm_pen_cv <- cv.glmnet(y = t2dm[, Rehosp_90_specific], x = X,
                      family = "binomial",
                      nfolds = 5,
                      alpha = 1,
                      type.measure = "auc")

## Fit auf alle Daten mit berechneten Lambdas
glm_pen <- glmnet(y = t2dm[, Rehosp_90_specific], x = X,
                family = "binomial",
                alpha = 1,
                lambda = glm_pen_cv$lambda)

coef_glmnet <- coef(glm_pen)
opt.glmnet <- match(glm_pen_cv$lambda.min, glm_pen_cv$lambda)

## Coefficients mit optimalem Lambda
coef.glmnet.opt <- coef_glmnet[,opt.glmnet]

names(coef.glmnet.opt) <- str_remove(names(coef.glmnet.opt), pattern = "TRUE")
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "age"] <- "Age*"
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "hosp_last_year"] <- "Hospitalisations last year*"

df_final$GLM <- df_final$GLM %>% round(digits = 3)
df_final$LASSO <- coef.glmnet.opt[names(coef.glmnet.opt) %in% rownames(df_final)] %>% round(digits = 3)

rownames(df_final)[1] <- "Intercept"
rownames(df_final)[2] <- "sex = female"

# Fill finalmatrix with coefficients
finalmatrix[rownames(finalmatrix) %in% rownames(df_final), 13:15] <- df_final

# Get increase for all START/STOPP = 0 vs. = 1
X_inc <- rbind(X[1,], X[1,1])
X_inc[1,28:61] <- 0
X_inc[2,28:61] <- 1
# first row all 0, second row all 1

preds_inc <- predict(
  glm_pen,
  X_inc,
  glm_pen_cv$lambda[opt.glmnet],
  type = "response"
)

increasematrix[5,] <- preds_inc

## Alternative ggplot
ggdf <- data.frame(lambda = glm_pen$lambda,
                  coefficient = coef_glmnet[2,],
                  variable = rownames(coef_glmnet)[2])
for(i in 3:nrow(coef_glmnet)){
  ggdf <- rbind(ggdf,
              data.frame(lambda = glm_pen$lambda,
                        coefficient = coef_glmnet[i,],
                        variable = rownames(coef_glmnet)[i]))
}

# saveRDS(ggdf, file = "./modelling_t2dm_90/lasso_coeff_plot.rds")
# saveRDS(list(glm_pen, opt.glmnet), file = "./modelling_t2dm_90/lasso_coeff_plot_misc.rds")

ggplot(ggdf, aes(x = lambda, y = coefficient, color = variable)) +
  geom_vline(xintercept = glm_pen$lambda[opt.glmnet], linetype = "dashed") +
  geom_line() +
  theme(legend.position = "none") +
  ylab(expression(hat(beta[j]))) +
  xlab(expression(lambda)) +
  annotate("text", y = 0.65, x = glm_pen$lambda[opt.glmnet] + 0.0075,
         label = substitute(
           paste(lambda[opt], " = ", nn),
           list(nn = round(glm_pen$lambda[opt.glmnet], digits = 4))
         )
  ) +
  ggtitle("LASSO coefficients (T2DM)")

# vhf
# filter for all criteria with at least 1% TRUE
filternumbers <- vector(mode = "numeric")
for(i in 4:98){
  if(!sum(vhf[,eval(as.name(names(vhf)[i]))]) == TRUE) > floor(nrow(vhf)*0.01)){
    filternumbers <- c(filternumbers, i)
  }
}

vhf_glm <- glm(Rehosp_90_specific ~ ., data = as.data.frame(vhf)[,-filternumbers], family = "binomial")

df <- summary(vhf_glm)$coefficients
df <- as.data.frame(df)
rownames(df)[rownames(df) == "age"] <- "Age*"
rownames(df)[rownames(df) == "hosp_last_year"] <- "Hospitalisations last year*"
rownames(df) <- str_remove(rownames(df), pattern = "TRUE")

df_final <- df[, c(1,4)]
df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

# LASSO
X <- model.matrix(Rehosp_90_specific ~ .,
                 data = vhf)[, -1][, -filternumbers]

```

```

set.seed(14102020)
glm_pen_cv <- cv.glmnet(y = vhf[, Rehosp_90_specific], x = X,
  family = "binomial",
  nfolds = 5,
  alpha = 1,
  type.measure = "auc")

## Fit auf alle Daten mit berechneten Lambdas
glm_pen <- glmnet(y = vhf[, Rehosp_90_specific], x = X,
  family = "binomial",
  alpha = 1,
  lambda = glm_pen_cv$lambda)

coef_glmnet <- coef(glm_pen)
opt.glmnet <- match(glm_pen_cv$lambda.min, glm_pen_cv$lambda)

## Coefficients mit optimalem Lambda
coef.glmnet.opt <- coef_glmnet[,opt.glmnet]

names(coef.glmnet.opt) <- str_remove(names(coef.glmnet.opt), pattern = "TRUE")
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "age"] <- "Age*"
names(coef.glmnet.opt)[names(coef.glmnet.opt) == "hosp_last_year"] <- "Hospitalisations last year*"

df_final$GLM <- df_final$GLM %>% round(digits = 3)
df_final$LASSO <- coef.glmnet.opt[names(coef.glmnet.opt) %in% rownames(df_final)] %>% round(digits = 3)

rownames(df_final)[1] <- "Intercept"
rownames(df_final)[2] <- "sex = female"

# Fill finalmatrix with coefficients
finalmatrix[rownames(finalmatrix) %in% rownames(df_final), 16:18] <- df_final

# Get increase for all START/STOPP = 0 vs. = 1
X_inc <- rbind(X[1,], X[1,1])
X_inc[1,26:57] <- 0
X_inc[2,26:57] <- 1
# first row all 0, second row all 1

preds_inc <- predict(
  glm_pen,
  X_inc,
  glm_pen_cv$lambda[opt.glmnet],
  type = "response"
)

increasematrix[6,] <- preds_inc

temp <- reshape2::melt(increasematrix)
temp$disease <- rep(c("COPD", "HF", "AMI", "OS", "DM", "S/AF"), times = 2)
temp$variable <- as.character(temp$variable)
temp$variable[temp$variable == "all_zero"] <- "START/STOPP = 0"
temp$variable[temp$variable == "all_one"] <- "START/STOPP = 1"
names(temp)[1] <- "Setting"

ggplot(temp, aes(x = disease, fill = Setting, y = value)) +
  geom_bar(stat = "identity", position = "dodge") +
  ylab("Predicted rehospitalisation probability") +
  xlab("Disease") +
  scale_fill_manual(values = c("START/STOPP = 0" = "black", "START/STOPP = 1" = "gray"))

#saveRDS(increasematrix, file = "./increasematrix.rds")

## Alternative ggplot
ggdf <- data.frame(lambda = glm_pen$lambda,
  coefficient = coef_glmnet[2,],
  variable = rownames(coef_glmnet)[2])
for(i in 3:nrow(coef_glmnet)){
  ggdf <- rbind(ggdf,
    data.frame(lambda = glm_pen$lambda,
      coefficient = coef_glmnet[i,],
      variable = rownames(coef_glmnet)[i]))
}

# saveRDS(ggdf, file = "./modelling_vhf_90/lasso_coeff_plot.rds")
# saveRDS(list(glm_pen, opt.glmnet), file = "./modelling_vhf_90/lasso_coeff_plot_misc.rds")

ggplot(ggdf, aes(x = lambda, y = coefficient, color = variable)) +
  geom_vline(xintercept = glm_pen$lambda[opt.glmnet], linetype = "dashed") +
  geom_line() +
  theme(legend.position = "none") +
  ylab(expression(hat(beta[j]))) +
  xlab(expression(lambda)) +
  annotate("text", y = 0.46, x = glm_pen$lambda[opt.glmnet] + 0.0015,
    label = substitute(
      paste(lambda[opt], " = ", nn),
      list(nn = round(glm_pen$lambda[opt.glmnet], digits = 5))
    )
  ) +
  ggtitle("LASSO coefficients (VHF)")

#write.xlsx(finalmatrix, file = "./Tabelle_coeffs.xlsx")
# read.xlsx(file = "./Tabelle_coeffs.xlsx", sheetIndex = 1)

#### increasematrix ####
# Computed in part above
#increasematrix <- readRDS(file = "./increasematrix.rds")

temp <- reshape2::melt(increasematrix)
temp$disease <- rep(c("COPD", "HF", "AMI", "OS", "DM", "S/AF"), times = 2)
temp$variable <- as.character(temp$variable)
temp$variable[temp$variable == "all_zero"] <- "START/STOPP = 0"
temp$variable[temp$variable == "all_one"] <- "START/STOPP = 1"
names(temp)[1] <- "Setting"

```

```

ggplot(temp, aes(x = disease, fill = Setting, y = value)) +
  geom_bar(stat = "identity", position = "dodge") +
  ylab("Predicted rehospitalisation probability") +
  xlab("Disease") +
  scale_fill_manual(values = c("START/STOPP = 0" = "black", "START/STOPP = 1" = "gray"))

#####
#####
##### Figures #####
#####
#####

#### Performance Evaluation ####
# Anzahl Folds
k <- 10

##
## Speichere Ergebnisse Parametertuning LASSO
##
copd_params_lasso <- vector(mode = "numeric")
copd_cv_results_lasso <- list()
hf_ukhd_params_lasso <- vector(mode = "numeric")
hf_ukhd_cv_results_lasso <- list()
mi_ukhd_params_lasso <- vector(mode = "numeric")
mi_ukhd_cv_results_lasso <- list()
osteo_params_lasso <- vector(mode = "numeric")
osteo_cv_results_lasso <- list()
t2dm_params_lasso <- vector(mode = "numeric")
t2dm_cv_results_lasso <- list()
vhf_params_lasso <- vector(mode = "numeric")
vhf_cv_results_lasso <- list()

##
## Speichere Modellgüte der Folds
##
copd_model_performance <- data.frame()
hf_ukhd_model_performance <- data.frame()
mi_ukhd_model_performance <- data.frame()
osteo_model_performance <- data.frame()
t2dm_model_performance <- data.frame()
vhf_model_performance <- data.frame()

set.seed(28092020)
##
## Bestimmte Folds
##
copd_folds_global <- sample(1:k, replace = TRUE, size = nrow(copd))
hf_ukhd_folds_global <- sample(1:k, replace = TRUE, size = nrow(hf_ukhd))
mi_ukhd_folds_global <- sample(1:k, replace = TRUE, size = nrow(mi_ukhd))
osteo_folds_global <- sample(1:k, replace = TRUE, size = nrow(osteo))
t2dm_folds_global <- sample(1:k, replace = TRUE, size = nrow(t2dm))
vhf_folds_global <- sample(1:k, replace = TRUE, size = nrow(vhf))

##
## Bestimme zu filternde Variablen
##
filternumbers_copd <- vector(mode = "numeric")
filternumbers_hf_ukhd <- vector(mode = "numeric")
filternumbers_mi_ukhd <- vector(mode = "numeric")
filternumbers_osteo <- vector(mode = "numeric")
filternumbers_t2dm <- vector(mode = "numeric")
filternumbers_vhf <- vector(mode = "numeric")

for(i in 4:98){

  if(!sum(copd[,eval(as.name(names(copd)[i]))] == TRUE) > floor(nrow(copd)*0.01)){

    filternumbers_copd <- c(filternumbers_copd, i)

  }

  if(!sum(hf_ukhd[,eval(as.name(names(hf_ukhd)[i]))] == TRUE) > floor(nrow(hf_ukhd)*0.01)){

    filternumbers_hf_ukhd <- c(filternumbers_hf_ukhd, i)

  }

  if(!sum(mi_ukhd[,eval(as.name(names(mi_ukhd)[i]))] == TRUE) > floor(nrow(mi_ukhd)*0.01)){

    filternumbers_mi_ukhd <- c(filternumbers_mi_ukhd, i)

  }

  if(!sum(osteo[,eval(as.name(names(osteo)[i]))] == TRUE) > floor(nrow(osteo)*0.01)){

    filternumbers_osteo <- c(filternumbers_osteo, i)

  }

  if(!sum(t2dm[,eval(as.name(names(t2dm)[i]))] == TRUE) > floor(nrow(t2dm)*0.01)){

    filternumbers_t2dm <- c(filternumbers_t2dm, i)

  }

  if(!sum(vhf[,eval(as.name(names(vhf)[i]))] == TRUE) > floor(nrow(vhf)*0.01)){

    filternumbers_vhf <- c(filternumbers_vhf, i)

  }

}

##

```

```

## Bestimme zu filternde Variablen für Basis-Modelle (ohne START/STOPP)
##

filternumbers_copd2 <- c(filternumbers_copd, 35:98)
filternumbers_hf_ukhd2 <- c(filternumbers_hf_ukhd, 35:98)
filternumbers_mi_ukhd2 <- c(filternumbers_mi_ukhd, 35:98)
filternumbers_osteo2 <- c(filternumbers_osteo, 35:98)
filternumbers_t2dm2 <- c(filternumbers_t2dm, 35:98)
filternumbers_vhf2 <- c(filternumbers_vhf, 35:98)

for(i in 1:k){

  ##
  ## Bestimme Folds
  ##
  copd_train_ids <- (1:nrow(copd))[copd_folds_global != i]
  copd_test_ids <- (1:nrow(copd))[copd_folds_global == i]
  hf_ukhd_train_ids <- (1:nrow(hf_ukhd))[hf_ukhd_folds_global != i]
  hf_ukhd_test_ids <- (1:nrow(hf_ukhd))[hf_ukhd_folds_global == i]
  mi_ukhd_train_ids <- (1:nrow(mi_ukhd))[mi_ukhd_folds_global != i]
  mi_ukhd_test_ids <- (1:nrow(mi_ukhd))[mi_ukhd_folds_global == i]
  osteo_train_ids <- (1:nrow(osteo))[osteo_folds_global != i]
  osteo_test_ids <- (1:nrow(osteo))[osteo_folds_global == i]
  t2dm_train_ids <- (1:nrow(t2dm))[t2dm_folds_global != i]
  t2dm_test_ids <- (1:nrow(t2dm))[t2dm_folds_global == i]
  vhf_train_ids <- (1:nrow(vhf))[vhf_folds_global != i]
  vhf_test_ids <- (1:nrow(vhf))[vhf_folds_global == i]

  ##
  ##
  ## Fit glm ##
  ##
  ##
  copd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(copd[copd_train_ids]), family = "binomial")
  hf_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(hf_ukhd[hf_ukhd_train_ids]), family = "binomial")
  mi_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(mi_ukhd[mi_ukhd_train_ids]), family = "binomial")
  osteo_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(osteo[osteo_train_ids]), family = "binomial")
  t2dm_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(t2dm[t2dm_train_ids]), family = "binomial")
  vhf_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(vhf[vhf_train_ids]), family = "binomial")

  ##
  ##
  ## Fit basis glm ##
  ##
  ##
  copd_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(copd[copd_train_ids]), family = "binomial")
  hf_ukhd_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(hf_ukhd[hf_ukhd_train_ids]), family = "binomial")
  mi_ukhd_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(mi_ukhd[mi_ukhd_train_ids]), family = "binomial")
  osteo_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(osteo[osteo_train_ids]), family = "binomial")
  t2dm_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(t2dm[t2dm_train_ids]), family = "binomial")
  vhf_glm_ml2 <- glm(Rehosp_90_specific ~ ., data = as.data.frame(vhf[vhf_train_ids]), family = "binomial")

  ##
  ##
  ## Fit LASSO ##
  ##
  ##
  copd_X <- model.matrix(Rehosp_90_specific ~ .,
                        data = copd[, -1], -filternumbers_copd)
  hf_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
                           data = hf_ukhd[, -1], -filternumbers_hf_ukhd)
  mi_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
                           data = mi_ukhd[, -1], -filternumbers_mi_ukhd)
  osteo_X <- model.matrix(Rehosp_90_specific ~ .,
                         data = osteo[, -1], -filternumbers_osteo)
  t2dm_X <- model.matrix(Rehosp_90_specific ~ .,
                        data = t2dm[, -1], -filternumbers_t2dm)
  vhf_X <- model.matrix(Rehosp_90_specific ~ .,
                      data = vhf[, -1], -filternumbers_vhf)

  copd_glm_pen_cv <- cv.glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X[copd_train_ids,],
                              family = "binomial",
                              alpha = 1,
                              nfolds = 5,
                              type.measure = "auc")
  hf_ukhd_glm_pen_cv <- cv.glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X[hf_ukhd_train_ids,],
                                  family = "binomial",
                                  alpha = 1,
                                  nfolds = 5,
                                  type.measure = "auc")
  mi_ukhd_glm_pen_cv <- cv.glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X[mi_ukhd_train_ids,],
                                  family = "binomial",
                                  alpha = 1,
                                  nfolds = 5,
                                  type.measure = "auc")
  osteo_glm_pen_cv <- cv.glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X[osteo_train_ids,],
                                family = "binomial",
                                alpha = 1,
                                nfolds = 5,
                                type.measure = "auc")
  t2dm_glm_pen_cv <- cv.glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X[t2dm_train_ids,],
                              family = "binomial",
                              alpha = 1,
                              nfolds = 5,
                              type.measure = "auc")
  vhf_glm_pen_cv <- cv.glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X[vhf_train_ids,],
                              family = "binomial",
                              alpha = 1,
                              nfolds = 5,
                              type.measure = "auc")

  copd_opt.lambda <- match(copd_glm_pen_cv$lambda.min, copd_glm_pen_cv$lambda)
  hf_ukhd_opt.lambda <- match(hf_ukhd_glm_pen_cv$lambda.min, hf_ukhd_glm_pen_cv$lambda)

```

```

mi_ukhd_opt.lambda <- match(mi_ukhd_glm_pen_cv$lambda.min,mi_ukhd_glm_pen_cv$lambda)
osteo_opt.lambda <- match(osteo_glm_pen_cv$lambda.min,osteo_glm_pen_cv$lambda)
t2dm_opt.lambda <- match(t2dm_glm_pen_cv$lambda.min,t2dm_glm_pen_cv$lambda)
vhf_opt.lambda <- match(vhf_glm_pen_cv$lambda.min,vhf_glm_pen_cv$lambda)

##
## Speichere optimales Lambda
##
copd_params_lasso <- c(copd_params_lasso, copd_glm_pen_cv$lambda[copd_opt.lambda])
hf_ukhd_params_lasso <- c(hf_ukhd_params_lasso, hf_ukhd_glm_pen_cv$lambda[hf_ukhd_opt.lambda])
mi_ukhd_params_lasso <- c(mi_ukhd_params_lasso, mi_ukhd_glm_pen_cv$lambda[mi_ukhd_opt.lambda])
osteo_params_lasso <- c(osteo_params_lasso, osteo_glm_pen_cv$lambda[osteo_opt.lambda])
t2dm_params_lasso <- c(t2dm_params_lasso, t2dm_glm_pen_cv$lambda[t2dm_opt.lambda])
vhf_params_lasso <- c(vhf_params_lasso, vhf_glm_pen_cv$lambda[vhf_opt.lambda])

##
## Speichere Parametertuning Ergebnisse
##
copd_cv_results_lasso <- c(copd_cv_results_lasso, list(
  data.frame(lambda = copd_glm_pen_cv$lambda,
    auc = copd_glm_pen_cv$cvm)
))
hf_ukhd_cv_results_lasso <- c(hf_ukhd_cv_results_lasso, list(
  data.frame(lambda = hf_ukhd_glm_pen_cv$lambda,
    auc = hf_ukhd_glm_pen_cv$cvm)
))
mi_ukhd_cv_results_lasso <- c(mi_ukhd_cv_results_lasso, list(
  data.frame(lambda = mi_ukhd_glm_pen_cv$lambda,
    auc = mi_ukhd_glm_pen_cv$cvm)
))
osteo_cv_results_lasso <- c(osteo_cv_results_lasso, list(
  data.frame(lambda = osteo_glm_pen_cv$lambda,
    auc = osteo_glm_pen_cv$cvm)
))
t2dm_cv_results_lasso <- c(t2dm_cv_results_lasso, list(
  data.frame(lambda = t2dm_glm_pen_cv$lambda,
    auc = t2dm_glm_pen_cv$cvm)
))
vhf_cv_results_lasso <- c(vhf_cv_results_lasso, list(
  data.frame(lambda = vhf_glm_pen_cv$lambda,
    auc = vhf_glm_pen_cv$cvm)
))

## Fit auf alle Daten mit berechnetem Lambda
copd_glm_pen <- glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X[copd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = copd_glm_pen_cv$lambda)
hf_ukhd_glm_pen <- glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X[hf_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = hf_ukhd_glm_pen_cv$lambda)
mi_ukhd_glm_pen <- glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X[mi_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = mi_ukhd_glm_pen_cv$lambda)
osteo_glm_pen <- glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X[osteo_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = osteo_glm_pen_cv$lambda)
t2dm_glm_pen <- glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X[t2dm_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = t2dm_glm_pen_cv$lambda)
vhf_glm_pen <- glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X[vhf_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = vhf_glm_pen_cv$lambda)

print(paste("LASSO fit done. Fold:", i))

##
##
## Fit basis LASSO ##
##
copd_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = copd)[, -1][, -filternumbers_copd2]
hf_ukhd_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = hf_ukhd)[, -1][, -filternumbers_hf_ukhd2]
mi_ukhd_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = mi_ukhd)[, -1][, -filternumbers_mi_ukhd2]
osteo_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = osteo)[, -1][, -filternumbers_osteo2]
t2dm_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = t2dm)[, -1][, -filternumbers_t2dm2]
vhf_X2 <- model.matrix(Rehosp_90_specific ~ .,
  data = vhf)[, -1][, -filternumbers_vhf2]

copd_glm_pen_cv2 <- cv.glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X2[copd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
hf_ukhd_glm_pen_cv2 <- cv.glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X2[hf_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
mi_ukhd_glm_pen_cv2 <- cv.glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X2[mi_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
osteo_glm_pen_cv2 <- cv.glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X2[osteo_train_ids,],
  family = "binomial",

```

```

alpha = 1,
nfolds = 5,
type.measure = "auc")
t2dm_glm_pen_cv2 <- cv.glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X2[t2dm_train_ids,],
family = "binomial",
alpha = 1,
nfolds = 5,
type.measure = "auc")
vhf_glm_pen_cv2 <- cv.glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X2[vhf_train_ids,],
family = "binomial",
alpha = 1,
nfolds = 5,
type.measure = "auc")

copd_opt.lambda2 <- match(copd_glm_pen_cv2$lambda.min, copd_glm_pen_cv2$lambda)
hf_ukhd_opt.lambda2 <- match(hf_ukhd_glm_pen_cv2$lambda.min, hf_ukhd_glm_pen_cv2$lambda)
mi_ukhd_opt.lambda2 <- match(mi_ukhd_glm_pen_cv2$lambda.min, mi_ukhd_glm_pen_cv2$lambda)
osteo_opt.lambda2 <- match(osteo_glm_pen_cv2$lambda.min, osteo_glm_pen_cv2$lambda)
t2dm_opt.lambda2 <- match(t2dm_glm_pen_cv2$lambda.min, t2dm_glm_pen_cv2$lambda)
vhf_opt.lambda2 <- match(vhf_glm_pen_cv2$lambda.min, vhf_glm_pen_cv2$lambda)

## Fit auf alle Daten mit berechnetem Lambda
copd_glm_pen2 <- glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X2[copd_train_ids,],
family = "binomial",
alpha = 1,
lambda = copd_glm_pen_cv2$lambda)
hf_ukhd_glm_pen2 <- glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X2[hf_ukhd_train_ids,],
family = "binomial",
alpha = 1,
lambda = hf_ukhd_glm_pen_cv2$lambda)
mi_ukhd_glm_pen2 <- glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X2[mi_ukhd_train_ids,],
family = "binomial",
alpha = 1,
lambda = mi_ukhd_glm_pen_cv2$lambda)
osteo_glm_pen2 <- glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X2[osteo_train_ids,],
family = "binomial",
alpha = 1,
lambda = osteo_glm_pen_cv2$lambda)
t2dm_glm_pen2 <- glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X2[t2dm_train_ids,],
family = "binomial",
alpha = 1,
lambda = t2dm_glm_pen_cv2$lambda)
vhf_glm_pen2 <- glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X2[vhf_train_ids,],
family = "binomial",
alpha = 1,
lambda = vhf_glm_pen_cv2$lambda)

##
##
## Predictions der Modelle für Testdaten ##
##
##
## GLM
##
# Predictions
copd_test_preds_glm <- predict(copd_glm_ml, as.data.frame(copd[copd_test_ids]), -filternumbers_copd, type = "response")
hf_ukhd_test_preds_glm <- predict(hf_ukhd_glm_ml, as.data.frame(hf_ukhd[hf_ukhd_test_ids]), -filternumbers_hf_ukhd, type = "response")
mi_ukhd_test_preds_glm <- predict(mi_ukhd_glm_ml, as.data.frame(mi_ukhd[mi_ukhd_test_ids]), -filternumbers_mi_ukhd, type = "response")
osteo_test_preds_glm <- predict(osteo_glm_ml, as.data.frame(osteo[osteo_test_ids]), -filternumbers_osteo, type = "response")
t2dm_test_preds_glm <- predict(t2dm_glm_ml, as.data.frame(t2dm[t2dm_test_ids]), -filternumbers_t2dm, type = "response")
vhf_test_preds_glm <- predict(vhf_glm_ml, as.data.frame(vhf[vhf_test_ids]), -filternumbers_vhf, type = "response")

# Precision table for roc function
copd_predicted_data_glm <- data.table(test_response = copd[copd_test_ids, Rehosp_90_specific], pred = copd_test_preds_glm)
hf_ukhd_predicted_data_glm <- data.table(test_response = hf_ukhd[hf_ukhd_test_ids, Rehosp_90_specific], pred = hf_ukhd_test_preds_glm)
mi_ukhd_predicted_data_glm <- data.table(test_response = mi_ukhd[mi_ukhd_test_ids, Rehosp_90_specific], pred = mi_ukhd_test_preds_glm)
osteo_predicted_data_glm <- data.table(test_response = osteo[osteo_test_ids, Rehosp_90_specific], pred = osteo_test_preds_glm)
t2dm_predicted_data_glm <- data.table(test_response = t2dm[t2dm_test_ids, Rehosp_90_specific], pred = t2dm_test_preds_glm)
vhf_predicted_data_glm <- data.table(test_response = vhf[vhf_test_ids, Rehosp_90_specific], pred = vhf_test_preds_glm)

# Roc function
copd_roc_object_glm <- roc(copd_predicted_data_glm, "test_response", "pred")
hf_ukhd_roc_object_glm <- roc(hf_ukhd_predicted_data_glm, "test_response", "pred")
mi_ukhd_roc_object_glm <- roc(mi_ukhd_predicted_data_glm, "test_response", "pred")
osteo_roc_object_glm <- roc(osteo_predicted_data_glm, "test_response", "pred")
t2dm_roc_object_glm <- roc(t2dm_predicted_data_glm, "test_response", "pred")
vhf_roc_object_glm <- roc(vhf_predicted_data_glm, "test_response", "pred")

##
## Basis GLM
##
copd_test_preds_glm2 <- predict(copd_glm_ml2, as.data.frame(copd[copd_test_ids]), -filternumbers_copd2, type = "response")
hf_ukhd_test_preds_glm2 <- predict(hf_ukhd_glm_ml2, as.data.frame(hf_ukhd[hf_ukhd_test_ids]), -filternumbers_hf_ukhd2, type = "response")
mi_ukhd_test_preds_glm2 <- predict(mi_ukhd_glm_ml2, as.data.frame(mi_ukhd[mi_ukhd_test_ids]), -filternumbers_mi_ukhd2, type = "response")
osteo_test_preds_glm2 <- predict(osteo_glm_ml2, as.data.frame(osteo[osteo_test_ids]), -filternumbers_osteo2, type = "response")
t2dm_test_preds_glm2 <- predict(t2dm_glm_ml2, as.data.frame(t2dm[t2dm_test_ids]), -filternumbers_t2dm2, type = "response")
vhf_test_preds_glm2 <- predict(vhf_glm_ml2, as.data.frame(vhf[vhf_test_ids]), -filternumbers_vhf2, type = "response")

##
## LASSO
##
copd_test_preds_lasso <- predict(
copd_glm_pen,
copd_X[copd_test_ids,],
type = "response",
copd_glm_pen_cv$lambda[copd_opt.lambda]
)
hf_ukhd_test_preds_lasso <- predict(
hf_ukhd_glm_pen,
hf_ukhd_X[hf_ukhd_test_ids,],
type = "response",
hf_ukhd_glm_pen_cv$lambda[hf_ukhd_opt.lambda]
)

```

```

mi_ukhd_test_preds_lasso <- predict(
  mi_ukhd_glm_pen,
  mi_ukhd_X[mi_ukhd_test_ids,],
  type = "response",
  mi_ukhd_glm_pen_cv$lambda[mi_ukhd_opt.lambda]
)
osteo_test_preds_lasso <- predict(
  osteo_glm_pen,
  osteo_X[osteo_test_ids,],
  type = "response",
  osteo_glm_pen_cv$lambda[osteo_opt.lambda]
)
t2dm_test_preds_lasso <- predict(
  t2dm_glm_pen,
  t2dm_X[t2dm_test_ids,],
  type = "response",
  t2dm_glm_pen_cv$lambda[t2dm_opt.lambda]
)
vhf_test_preds_lasso <- predict(
  vhf_glm_pen,
  vhf_X[vhf_test_ids,],
  type = "response",
  vhf_glm_pen_cv$lambda[vhf_opt.lambda]
)

# Prediction table for roc function
copd_predicted_data_lasso <- data.table(test_response = copd[copd_test_ids, Rehos90_specific], pred = copd_test_preds_lasso[, 1])
hf_ukhd_predicted_data_lasso <- data.table(test_response = hf_ukhd[hf_ukhd_test_ids, Rehos90_specific], pred = hf_ukhd_test_preds_lasso[, 1])
mi_ukhd_predicted_data_lasso <- data.table(test_response = mi_ukhd[mi_ukhd_test_ids, Rehos90_specific], pred = mi_ukhd_test_preds_lasso[, 1])
osteo_predicted_data_lasso <- data.table(test_response = osteo[osteo_test_ids, Rehos90_specific], pred = osteo_test_preds_lasso[, 1])
t2dm_predicted_data_lasso <- data.table(test_response = t2dm[t2dm_test_ids, Rehos90_specific], pred = t2dm_test_preds_lasso[, 1])
vhf_predicted_data_lasso <- data.table(test_response = vhf[vhf_test_ids, Rehos90_specific], pred = vhf_test_preds_lasso[, 1])

# roc function
copd_roc_object_lasso <- roc(copd_predicted_data_lasso, "test_response", "pred")
hf_ukhd_roc_object_lasso <- roc(hf_ukhd_predicted_data_lasso, "test_response", "pred")
mi_ukhd_roc_object_lasso <- roc(mi_ukhd_predicted_data_lasso, "test_response", "pred")
osteo_roc_object_lasso <- roc(osteo_predicted_data_lasso, "test_response", "pred")
t2dm_roc_object_lasso <- roc(t2dm_predicted_data_lasso, "test_response", "pred")
vhf_roc_object_lasso <- roc(vhf_predicted_data_lasso, "test_response", "pred")

##
## Basis LASSO
##
copd_test_preds_lasso2 <- predict(
  copd_glm_pen2,
  copd_X2[copd_test_ids,],
  type = "response",
  copd_glm_pen_cv2$lambda[copd_opt.lambda2]
)
hf_ukhd_test_preds_lasso2 <- predict(
  hf_ukhd_glm_pen2,
  hf_ukhd_X2[hf_ukhd_test_ids,],
  type = "response",
  hf_ukhd_glm_pen_cv2$lambda[hf_ukhd_opt.lambda2]
)
mi_ukhd_test_preds_lasso2 <- predict(
  mi_ukhd_glm_pen2,
  mi_ukhd_X2[mi_ukhd_test_ids,],
  type = "response",
  mi_ukhd_glm_pen_cv2$lambda[mi_ukhd_opt.lambda2]
)
osteo_test_preds_lasso2 <- predict(
  osteo_glm_pen2,
  osteo_X2[osteo_test_ids,],
  type = "response",
  osteo_glm_pen_cv2$lambda[osteo_opt.lambda2]
)
t2dm_test_preds_lasso2 <- predict(
  t2dm_glm_pen2,
  t2dm_X2[t2dm_test_ids,],
  type = "response",
  t2dm_glm_pen_cv2$lambda[t2dm_opt.lambda2]
)
vhf_test_preds_lasso2 <- predict(
  vhf_glm_pen2,
  vhf_X2[vhf_test_ids,],
  type = "response",
  vhf_glm_pen_cv2$lambda[vhf_opt.lambda2]
)

## IDIs - GLM
idi_copd_glm <- mean(copd_test_preds_glm2[copd[copd_test_ids, Rehos90_specific] == TRUE]) - mean(copd_test_preds_glm2[copd[copd_test_ids,
Rehos90_specific] == FALSE]) -
(mean(copd_test_preds_glm[copd[copd_test_ids, Rehos90_specific] == TRUE]) - mean(copd_test_preds_glm[copd[copd_test_ids, Rehos90_specific] ==
FALSE]))
idi_hf_ukhd_glm <- mean(hf_ukhd_test_preds_glm2[hf_ukhd[hf_ukhd_test_ids, Rehos90_specific] == TRUE]) -
mean(hf_ukhd_test_preds_glm2[hf_ukhd[hf_ukhd_test_ids, Rehos90_specific] == FALSE]) -
(mean(hf_ukhd_test_preds_glm[hf_ukhd[hf_ukhd_test_ids, Rehos90_specific] == TRUE]) - mean(hf_ukhd_test_preds_glm[hf_ukhd[hf_ukhd_test_ids,
Rehos90_specific] == FALSE]))
idi_mi_ukhd_glm <- mean(mi_ukhd_test_preds_glm2[mi_ukhd[mi_ukhd_test_ids, Rehos90_specific] == TRUE]) -
mean(mi_ukhd_test_preds_glm2[mi_ukhd[mi_ukhd_test_ids, Rehos90_specific] == FALSE]) -
(mean(mi_ukhd_test_preds_glm[mi_ukhd[mi_ukhd_test_ids, Rehos90_specific] == TRUE]) - mean(mi_ukhd_test_preds_glm[mi_ukhd[mi_ukhd_test_ids,
Rehos90_specific] == FALSE]))
idi_osteo_glm <- mean(osteo_test_preds_glm2[osteo[osteo_test_ids, Rehos90_specific] == TRUE]) - mean(osteo_test_preds_glm2[osteo[osteo_test_ids,
Rehos90_specific] == FALSE]) -
(mean(osteo_test_preds_glm[osteo[osteo_test_ids, Rehos90_specific] == TRUE]) - mean(osteo_test_preds_glm[osteo[osteo_test_ids,
Rehos90_specific] == FALSE]))
idi_t2dm_glm <- mean(t2dm_test_preds_glm2[t2dm[t2dm_test_ids, Rehos90_specific] == TRUE]) - mean(t2dm_test_preds_glm2[t2dm[t2dm_test_ids,
Rehos90_specific] == FALSE]) -
(mean(t2dm_test_preds_glm[t2dm[t2dm_test_ids, Rehos90_specific] == TRUE]) - mean(t2dm_test_preds_glm[t2dm[t2dm_test_ids, Rehos90_specific] ==
FALSE]))
idi_vhf_glm <- mean(vhf_test_preds_glm2[vhf[vhf_test_ids, Rehos90_specific] == TRUE]) - mean(vhf_test_preds_glm2[vhf[vhf_test_ids,
Rehos90_specific] == FALSE]) -
(mean(vhf_test_preds_glm[vhf[vhf_test_ids, Rehos90_specific] == TRUE]) - mean(vhf_test_preds_glm[vhf[vhf_test_ids, Rehos90_specific] ==
FALSE]))

```



```

## IDIs - LASSO
idi_copd_lasso <- mean(copd_test_preds_lasso2[copd[copd_test_ids, Rehosp_90_specific] == TRUE]) - mean(copd_test_preds_lasso2[copd[copd_test_ids,
Rehosp_90_specific] == FALSE]) -
  (mean(copd_test_preds_lasso[copd[copd_test_ids, Rehosp_90_specific] == TRUE]) - mean(copd_test_preds_lasso[copd[copd_test_ids,
Rehosp_90_specific] == FALSE]))
idi_hf_ukhd_lasso <- mean(hf_ukhd_test_preds_lasso2[hf_ukhd[hf_ukhd_test_ids, Rehosp_90_specific] == TRUE]) -
mean(hf_ukhd_test_preds_lasso2[hf_ukhd[hf_ukhd_test_ids, Rehosp_90_specific] == FALSE]) -
  (mean(hf_ukhd_test_preds_lasso[hf_ukhd[hf_ukhd_test_ids, Rehosp_90_specific] == TRUE]) - mean(hf_ukhd_test_preds_lasso[hf_ukhd[hf_ukhd_test_ids,
Rehosp_90_specific] == FALSE]))
idi_mi_ukhd_lasso <- mean(mi_ukhd_test_preds_lasso2[mi_ukhd[mi_ukhd_test_ids, Rehosp_90_specific] == TRUE]) -
mean(mi_ukhd_test_preds_lasso2[mi_ukhd[mi_ukhd_test_ids, Rehosp_90_specific] == FALSE]) -
  (mean(mi_ukhd_test_preds_lasso[mi_ukhd[mi_ukhd_test_ids, Rehosp_90_specific] == TRUE]) - mean(mi_ukhd_test_preds_lasso[mi_ukhd[mi_ukhd_test_ids,
Rehosp_90_specific] == FALSE]))
idi_osteo_lasso <- mean(osteo_test_preds_lasso2[osteo[osteo_test_ids, Rehosp_90_specific] == TRUE]) -
mean(osteo_test_preds_lasso2[osteo[osteo_test_ids, Rehosp_90_specific] == FALSE]) -
  (mean(osteo_test_preds_lasso[osteo[osteo_test_ids, Rehosp_90_specific] == TRUE]) - mean(osteo_test_preds_lasso[osteo[osteo_test_ids,
Rehosp_90_specific] == FALSE]))
idi_t2dm_lasso <- mean(t2dm_test_preds_lasso2[t2dm[t2dm_test_ids, Rehosp_90_specific] == TRUE]) - mean(t2dm_test_preds_lasso2[t2dm[t2dm_test_ids,
Rehosp_90_specific] == FALSE]) -
  (mean(t2dm_test_preds_lasso[t2dm[t2dm_test_ids, Rehosp_90_specific] == TRUE]) - mean(t2dm_test_preds_lasso[t2dm[t2dm_test_ids,
Rehosp_90_specific] == FALSE]))
idi_vhf_lasso <- mean(vhf_test_preds_lasso2[vhf[vhf_test_ids, Rehosp_90_specific] == TRUE]) - mean(vhf_test_preds_lasso2[vhf[vhf_test_ids,
Rehosp_90_specific] == FALSE]) -
  (mean(vhf_test_preds_lasso[vhf[vhf_test_ids, Rehosp_90_specific] == TRUE]) - mean(vhf_test_preds_lasso[vhf[vhf_test_ids, Rehosp_90_specific] ==
FALSE]))

# Save AUCs
copd_model_performance <- rbind(copd_model_performance,
  data.frame(glm_auc = copd_roc_object_glm$auc,
    lasso_auc = copd_roc_object_lasso$auc,
    glm_idi = idi_copd_glm,
    lasso_idi = idi_copd_lasso,
    fold = i))
hf_ukhd_model_performance <- rbind(hf_ukhd_model_performance,
  data.frame(glm_auc = hf_ukhd_roc_object_glm$auc,
    lasso_auc = hf_ukhd_roc_object_lasso$auc,
    glm_idi = idi_hf_ukhd_glm,
    lasso_idi = idi_hf_ukhd_lasso,
    fold = i))
mi_ukhd_model_performance <- rbind(mi_ukhd_model_performance,
  data.frame(glm_auc = mi_ukhd_roc_object_glm$auc,
    lasso_auc = mi_ukhd_roc_object_lasso$auc,
    glm_idi = idi_mi_ukhd_glm,
    lasso_idi = idi_mi_ukhd_lasso,
    fold = i))
osteo_model_performance <- rbind(osteo_model_performance,
  data.frame(glm_auc = osteo_roc_object_glm$auc,
    lasso_auc = osteo_roc_object_lasso$auc,
    glm_idi = idi_osteo_glm,
    lasso_idi = idi_osteo_lasso,
    fold = i))
t2dm_model_performance <- rbind(t2dm_model_performance,
  data.frame(glm_auc = t2dm_roc_object_glm$auc,
    lasso_auc = t2dm_roc_object_lasso$auc,
    glm_idi = idi_t2dm_glm,
    lasso_idi = idi_t2dm_lasso,
    fold = i))
vhf_model_performance <- rbind(vhf_model_performance,
  data.frame(glm_auc = vhf_roc_object_glm$auc,
    lasso_auc = vhf_roc_object_lasso$auc,
    glm_idi = idi_vhf_glm,
    lasso_idi = idi_vhf_lasso,
    fold = i))

print(paste("Predictions done. Fold:", i))

}

copd_model_performance <- readRDS(file = "./modelling_copd_90/model_performance2.rds")
copd_params_lasso <- readRDS(file = "./modelling_copd_90/params_lasso2.rds")
copd_cv_results_lasso <- readRDS(file = "./modelling_copd_90/cv_results_lasso2.rds")

hf_ukhd_model_performance <- readRDS(file = "./modelling_hf_ukhd_90/model_performance2.rds")
hf_ukhd_params_lasso <- readRDS(file = "./modelling_hf_ukhd_90/params_lasso2.rds")
hf_ukhd_cv_results_lasso <- readRDS(file = "./modelling_hf_ukhd_90/cv_results_lasso2.rds")

mi_ukhd_model_performance <- readRDS(file = "./modelling_mi_ukhd_90/model_performance2.rds")
mi_ukhd_params_lasso <- readRDS(file = "./modelling_mi_ukhd_90/params_lasso2.rds")
mi_ukhd_cv_results_lasso <- readRDS(file = "./modelling_mi_ukhd_90/cv_results_lasso2.rds")

osteo_model_performance <- readRDS(file = "./modelling_osteo_90/model_performance2.rds")
osteo_params_lasso <- readRDS(file = "./modelling_osteo_90/params_lasso2.rds")
osteo_cv_results_lasso <- readRDS(file = "./modelling_osteo_90/cv_results_lasso2.rds")

t2dm_model_performance <- readRDS(file = "./modelling_t2dm_90/model_performance2.rds")
t2dm_params_lasso <- readRDS(file = "./modelling_t2dm_90/params_lasso2.rds")
t2dm_cv_results_lasso <- readRDS(file = "./modelling_t2dm_90/cv_results_lasso2.rds")

vhf_model_performance <- readRDS(file = "./modelling_vhf_90/model_performance2.rds")
vhf_params_lasso <- readRDS(file = "./modelling_vhf_90/params_lasso2.rds")
vhf_cv_results_lasso <- readRDS(file = "./modelling_vhf_90/cv_results_lasso2.rds")

##
## Gesamt Performance
##
model_performance <- rbind(copd_model_performance,
  hf_ukhd_model_performance,
  mi_ukhd_model_performance,
  osteo_model_performance,
  t2dm_model_performance,
  vhf_model_performance)
model_performance$Krankheit <- rep(c("COPD", "HF", "AMI", "OS", "DM", "S/AF"), each = 10)

```

```

df <- reshape2::melt(model_performance[, c(1, 2, 6)])
df$variable <- rep(c("GLM", "LASSO"), each = 60)
df2 <- colMeans(model_performance[, 1:2]) %>%
  reshape2::melt(.)
df2$variable <- unique(df$variable)
names(df)[2] <- "Model"

# Falls filternumbers vorher nicht erzeugt wurden
{
# filternumbers_copd <- vector(mode = "numeric")
# filternumbers_hf_ukhd <- vector(mode = "numeric")
# filternumbers_mi_ukhd <- vector(mode = "numeric")
# filternumbers_osteo <- vector(mode = "numeric")
# filternumbers_t2dm <- vector(mode = "numeric")
# filternumbers_vhf <- vector(mode = "numeric")
#
# for(i in 4:98){
#
#   if(!sum(copd[,eval(as.name(names(copd)[i]))] == TRUE) > floor(nrow(copd)*0.01)){
#
#     filternumbers_copd <- c(filternumbers_copd, i)
#
#   }
#
#   if(!sum(hf_ukhd[,eval(as.name(names(hf_ukhd)[i]))] == TRUE) > floor(nrow(hf_ukhd)*0.01)){
#
#     filternumbers_hf_ukhd <- c(filternumbers_hf_ukhd, i)
#
#   }
#
#   if(!sum(mi_ukhd[,eval(as.name(names(mi_ukhd)[i]))] == TRUE) > floor(nrow(mi_ukhd)*0.01)){
#
#     filternumbers_mi_ukhd <- c(filternumbers_mi_ukhd, i)
#
#   }
#
#   if(!sum(osteo[,eval(as.name(names(osteo)[i]))] == TRUE) > floor(nrow(osteo)*0.01)){
#
#     filternumbers_osteo <- c(filternumbers_osteo, i)
#
#   }
#
#   if(!sum(t2dm[,eval(as.name(names(t2dm)[i]))] == TRUE) > floor(nrow(t2dm)*0.01)){
#
#     filternumbers_t2dm <- c(filternumbers_t2dm, i)
#
#   }
#
#   if(!sum(vhf[,eval(as.name(names(vhf)[i]))] == TRUE) > floor(nrow(vhf)*0.01)){
#
#     filternumbers_vhf <- c(filternumbers_vhf, i)
#
#   }
# }
}

#tiff("./boxplot_glm_lasso.tiff", units="in", width=10, height=5, res=300)
pl <- ggplot(df, aes(x = Krankheit, y = value, fill = Model)) +
  geom_boxplot() +
  xlab("Model") +
  ylab("c-Statistic") +
  scale_y_continuous(expand= expand_scale(mult = c(0, 0)), limits = c(0.45, 0.7)) +
  theme_classic(base_size = 16) +
  theme(panel.grid.major.y = element_line(size = 1,
    colour = "lightgrey"),
    panel.grid.minor.y = element_line(size = 0.25,
    colour = "lightgrey"),
    legend.position = "none") +
  scale_fill_manual(values = c("black", "darkgrey")) +
  annotate(geom = "text", x = 1, y = 0.53, label = paste("p = ", 99-length(filternumbers_mi_ukhd))) +
  annotate(geom = "text", x = 2, y = 0.62, label = paste("p = ", 99-length(filternumbers_copd))) +
  annotate(geom = "text", x = 3, y = 0.61, label = paste("p = ", 99-length(filternumbers_t2dm))) +
  annotate(geom = "text", x = 4, y = 0.59, label = paste("p = ", 99-length(filternumbers_hf_ukhd))) +
  annotate(geom = "text", x = 5, y = 0.46, label = paste("p = ", 99-length(filternumbers_osteo))) +
  annotate(geom = "text", x = 6, y = 0.61, label = paste("p = ", 99-length(filternumbers_vhf))) +
  annotate(geom = "text", x = 0.7, y = 0.615, label = paste("GLM")) +
  annotate(geom = "text", x = 1.3, y = 0.615, label = paste("LASSO")) +
  ggtitle("(A)")
#dev.off()

## IDI
df <- reshape2::melt(model_performance[, c(3, 4, 6)])
df$variable <- rep(c("GLM", "LASSO"), each = 60)
# df2 <- colMeans(model_performance[, 1:2]) %>%
#   reshape2::melt(.)
# df2$variable <- unique(df$variable)
names(df)[2] <- "Model"

## IDI negativ aufnehmen, weil beim Berechnen des IDIs die Modelle vertauscht wurden!!
ggplot(df, aes(x = Krankheit, y = -value, fill = Model)) +
  geom_boxplot() +
  xlab("Model") +
  ylab("IDI") +
  scale_y_continuous(expand= expand_scale(mult = c(0, 0)), limits = c(-0.01, 0.01)) +
  theme_classic(base_size = 16) +
  theme(panel.grid.major.y = element_line(size = 1,
    colour = "lightgrey"),
    panel.grid.minor.y = element_line(size = 0.25,
    colour = "lightgrey"),
    legend.position = "none") +
  scale_fill_manual(values = c("black", "darkgrey")) +
  # annotate(geom = "text", x = 1, y = 0.53, label = paste("p = ", 99-length(filternumbers_mi_ukhd))) +
  # annotate(geom = "text", x = 2, y = 0.62, label = paste("p = ", 99-length(filternumbers_copd))) +
  # annotate(geom = "text", x = 3, y = 0.61, label = paste("p = ", 99-length(filternumbers_t2dm))) +
  # annotate(geom = "text", x = 4, y = 0.59, label = paste("p = ", 99-length(filternumbers_hf_ukhd))) +

```

```

# annotate(geom = "text", x = 5, y = 0.46, label = paste("p = ", 99-length(filternumbers_osteo))) +
# annotate(geom = "text", x = 6, y = 0.61, label = paste("p = ", 99-length(filternumbers_vhf))) +
annotate(geom = "text", x = 0.7, y = 0.004, label = paste("GLM")) +
annotate(geom = "text", x = 1.3, y = 0.004, label = paste("LASSO")) #+
# ggtitle("A)")

##
## ROC
##
set.seed(14102020)
copd_train_ids <- sample(c(TRUE, FALSE), nrow(copd), prob = c(0.7, 0.3), replace = TRUE)
hf_ukhd_train_ids <- sample(c(TRUE, FALSE), nrow(hf_ukhd), prob = c(0.7, 0.3), replace = TRUE)
mi_ukhd_train_ids <- sample(c(TRUE, FALSE), nrow(mi_ukhd), prob = c(0.7, 0.3), replace = TRUE)
osteo_train_ids <- sample(c(TRUE, FALSE), nrow(osteo), prob = c(0.7, 0.3), replace = TRUE)
t2dm_train_ids <- sample(c(TRUE, FALSE), nrow(t2dm), prob = c(0.7, 0.3), replace = TRUE)
vhf_train_ids <- sample(c(TRUE, FALSE), nrow(vhf), prob = c(0.7, 0.3), replace = TRUE)

## Trainiere Modelle
# copd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(copd[copd_train_ids]), -filternumbers_copd, family = "binomial")
# hf_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(hf_ukhd[hf_ukhd_train_ids]), -filternumbers_hf_ukhd, family = "binomial")
# mi_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(mi_ukhd[mi_ukhd_train_ids]), -filternumbers_mi_ukhd, family = "binomial")
# osteo_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(osteo[osteo_train_ids]), -filternumbers_osteo, family = "binomial")
# t2dm_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(t2dm[t2dm_train_ids]), -filternumbers_t2dm, family = "binomial")
# vhf_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(vhf[vhf_train_ids]), -filternumbers_vhf, family = "binomial")

##
##
## Fit LASSO ##
##
##
copd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = copd[, -1][, -filternumbers_copd])
hf_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = hf_ukhd[, -1][, -filternumbers_hf_ukhd])
mi_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = mi_ukhd[, -1][, -filternumbers_mi_ukhd])
osteo_X <- model.matrix(Rehosp_90_specific ~ .,
  data = osteo[, -1][, -filternumbers_osteo])
t2dm_X <- model.matrix(Rehosp_90_specific ~ .,
  data = t2dm[, -1][, -filternumbers_t2dm])
vhf_X <- model.matrix(Rehosp_90_specific ~ .,
  data = vhf[, -1][, -filternumbers_vhf])

copd_glm_pen_cv <- cv.glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X[copd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
hf_ukhd_glm_pen_cv <- cv.glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X[hf_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
mi_ukhd_glm_pen_cv <- cv.glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X[mi_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
osteo_glm_pen_cv <- cv.glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X[osteo_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
t2dm_glm_pen_cv <- cv.glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X[t2dm_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
vhf_glm_pen_cv <- cv.glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X[vhf_train_ids,],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")

copd_opt.lambda <- match(copd_glm_pen_cv$lambda.min, copd_glm_pen_cv$lambda)
hf_ukhd_opt.lambda <- match(hf_ukhd_glm_pen_cv$lambda.min, hf_ukhd_glm_pen_cv$lambda)
mi_ukhd_opt.lambda <- match(mi_ukhd_glm_pen_cv$lambda.min, mi_ukhd_glm_pen_cv$lambda)
osteo_opt.lambda <- match(osteo_glm_pen_cv$lambda.min, osteo_glm_pen_cv$lambda)
t2dm_opt.lambda <- match(t2dm_glm_pen_cv$lambda.min, t2dm_glm_pen_cv$lambda)
vhf_opt.lambda <- match(vhf_glm_pen_cv$lambda.min, vhf_glm_pen_cv$lambda)

## Fit auf alle Daten mit berechnetem Lambda
copd_glm_pen <- glmnet(y = copd[copd_train_ids, Rehosp_90_specific], x = copd_X[copd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = copd_glm_pen_cv$lambda)
hf_ukhd_glm_pen <- glmnet(y = hf_ukhd[hf_ukhd_train_ids, Rehosp_90_specific], x = hf_ukhd_X[hf_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = hf_ukhd_glm_pen_cv$lambda)
mi_ukhd_glm_pen <- glmnet(y = mi_ukhd[mi_ukhd_train_ids, Rehosp_90_specific], x = mi_ukhd_X[mi_ukhd_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = mi_ukhd_glm_pen_cv$lambda)
osteo_glm_pen <- glmnet(y = osteo[osteo_train_ids, Rehosp_90_specific], x = osteo_X[osteo_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = osteo_glm_pen_cv$lambda)
t2dm_glm_pen <- glmnet(y = t2dm[t2dm_train_ids, Rehosp_90_specific], x = t2dm_X[t2dm_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = t2dm_glm_pen_cv$lambda)
vhf_glm_pen <- glmnet(y = vhf[vhf_train_ids, Rehosp_90_specific], x = vhf_X[vhf_train_ids,],
  family = "binomial",
  alpha = 1,
  lambda = vhf_glm_pen_cv$lambda)

```

```

## Berechne Test-Predictions
# GLM
# copd_test_preds_glm <- predict(copd_glm_ml, as.data.frame(copd[!copd_train_ids]), -filternumbers_copd, type = "response")
# hf_ukhd_test_preds_glm <- predict(hf_ukhd_glm_ml, as.data.frame(hf_ukhd[!hf_ukhd_train_ids]), -filternumbers_hf_ukhd, type = "response")
# mi_ukhd_test_preds_glm <- predict(mi_ukhd_glm_ml, as.data.frame(mi_ukhd[!mi_ukhd_train_ids]), -filternumbers_mi_ukhd, type = "response")
# osteo_test_preds_glm <- predict(osteo_glm_ml, as.data.frame(osteo[!osteo_train_ids]), -filternumbers_osteo, type = "response")
# t2dm_test_preds_glm <- predict(t2dm_glm_ml, as.data.frame(t2dm[!t2dm_train_ids]), -filternumbers_t2dm, type = "response")
# vhf_test_preds_glm <- predict(vhf_glm_ml, as.data.frame(vhf[!vhf_train_ids]), -filternumbers_vhf, type = "response")
#
#
# copd_predicted_data_glm <- data.table(test_response = copd[!copd_train_ids, Rehosp_90_specific], pred = copd_test_preds_glm)
# hf_ukhd_predicted_data_glm <- data.table(test_response = hf_ukhd[!hf_ukhd_train_ids, Rehosp_90_specific], pred = hf_ukhd_test_preds_glm)
# mi_ukhd_predicted_data_glm <- data.table(test_response = mi_ukhd[!mi_ukhd_train_ids, Rehosp_90_specific], pred = mi_ukhd_test_preds_glm)
# osteo_predicted_data_glm <- data.table(test_response = osteo[!osteo_train_ids, Rehosp_90_specific], pred = osteo_test_preds_glm)
# t2dm_predicted_data_glm <- data.table(test_response = t2dm[!t2dm_train_ids, Rehosp_90_specific], pred = t2dm_test_preds_glm)
# vhf_predicted_data_glm <- data.table(test_response = vhf[!vhf_train_ids, Rehosp_90_specific], pred = vhf_test_preds_glm)
#
# copd_roc_object_glm <- roc(copd_predicted_data_glm, "test_response", "pred")
# hf_ukhd_roc_object_glm <- roc(hf_ukhd_predicted_data_glm, "test_response", "pred")
# mi_ukhd_roc_object_glm <- roc(mi_ukhd_predicted_data_glm, "test_response", "pred")
# osteo_roc_object_glm <- roc(osteo_predicted_data_glm, "test_response", "pred")
# t2dm_roc_object_glm <- roc(t2dm_predicted_data_glm, "test_response", "pred")
# vhf_roc_object_glm <- roc(vhf_predicted_data_glm, "test_response", "pred")
#
## Vergleichende ROC-Kurve GLM
# df <- data.frame(Sensitivity = copd_roc_object_glm$sensitivities,
#                 Specificity = copd_roc_object_glm$specificities,
#                 best.x = coords(copd_roc_object_glm, "best", transpose = TRUE)[2],
#                 best.y = coords(copd_roc_object_glm, "best", transpose = TRUE)[3],
#                 Krankheit = "COPD")
# df <- rbind(df,
#            data.frame(Sensitivity = hf_ukhd_roc_object_glm$sensitivities,
#                      Specificity = hf_ukhd_roc_object_glm$specificities,
#                      best.x = coords(hf_ukhd_roc_object_glm, "best", transpose = TRUE)[2],
#                      best.y = coords(hf_ukhd_roc_object_glm, "best", transpose = TRUE)[3],
#                      Krankheit = "HI"))
# df <- rbind(df,
#            data.frame(Sensitivity = mi_ukhd_roc_object_glm$sensitivities,
#                      Specificity = mi_ukhd_roc_object_glm$specificities,
#                      best.x = coords(mi_ukhd_roc_object_glm, "best", transpose = TRUE)[2],
#                      best.y = coords(mi_ukhd_roc_object_glm, "best", transpose = TRUE)[3],
#                      Krankheit = "AMI"))
# df <- rbind(df,
#            data.frame(Sensitivity = osteo_roc_object_glm$sensitivities,
#                      Specificity = osteo_roc_object_glm$specificities,
#                      best.x = coords(osteo_roc_object_glm, "best", transpose = TRUE)[2],
#                      best.y = coords(osteo_roc_object_glm, "best", transpose = TRUE)[3],
#                      Krankheit = "OS"))
# df <- rbind(df,
#            data.frame(Sensitivity = t2dm_roc_object_glm$sensitivities,
#                      Specificity = t2dm_roc_object_glm$specificities,
#                      best.x = coords(t2dm_roc_object_glm, "best", transpose = TRUE)[2],
#                      best.y = coords(t2dm_roc_object_glm, "best", transpose = TRUE)[3],
#                      Krankheit = "DM"))
# df <- rbind(df,
#            data.frame(Sensitivity = vhf_roc_object_glm$sensitivities,
#                      Specificity = vhf_roc_object_glm$specificities,
#                      best.x = coords(vhf_roc_object_glm, "best", transpose = TRUE)[2],
#                      best.y = coords(vhf_roc_object_glm, "best", transpose = TRUE)[3],
#                      Krankheit = "S/AF"))
#
# ggplot(df, aes(x = 1-Specificity, y = Sensitivity)) +
#   geom_segment(aes(x = 0, y = 0, xend = 1, yend = 1), size = 1, color = "black") +
#   geom_line(size = 1.2) +
#   scale_x_continuous(name = "Specificity",
#                     breaks = c(0, 0.25, 0.5, 0.75, 1),
#                     labels = c("1.00", "0.75", "0.50", "0.25", "0.00"),
#                     expand = expand_scale(mult = c(0, 0)), limits = c(0, 1)
#   ) +
#   ylab("Sensitivity") +
#   theme_classic() +
#   scale_y_continuous(expand = expand_scale(mult = c(0, 0)), limits = c(0, 1)) +
#   geom_hline(data = df, aes(yintercept = best.y)) +
#   geom_vline(data = df, aes(xintercept = 1-best.x)) +
#   facet_wrap(vars(Krankheit), scales = "free")
#
## Berechne Test-Predictions
# LASSO
copd_test_preds_lasso <- predict(
  copd_glm_pen,
  copd_X[!copd_train_ids,],
  type = "response",
  copd_glm_pen_cv$lambda[copd_opt.lambda]
)
hf_ukhd_test_preds_lasso <- predict(
  hf_ukhd_glm_pen,
  hf_ukhd_X[!hf_ukhd_train_ids,],
  type = "response",
  hf_ukhd_glm_pen_cv$lambda[hf_ukhd_opt.lambda]
)
mi_ukhd_test_preds_lasso <- predict(
  mi_ukhd_glm_pen,
  mi_ukhd_X[!mi_ukhd_train_ids,],
  type = "response",
  mi_ukhd_glm_pen_cv$lambda[mi_ukhd_opt.lambda]
)
osteo_test_preds_lasso <- predict(
  osteo_glm_pen,
  osteo_X[!osteo_train_ids,],
  type = "response",
  osteo_glm_pen_cv$lambda[osteo_opt.lambda]
)
t2dm_test_preds_lasso <- predict(
  t2dm_glm_pen,
  t2dm_X[!t2dm_train_ids,],

```

```

type = "response",
t2dm_glm_pen_cv$lambda[t2dm_opt.lambda]
)
vhf_test_preds_lasso <- predict(
  vhf_glm_pen,
  vhf_X[!vhf_train_ids,],
  type = "response",
  vhf_glm_pen_cv$lambda[vhf_opt.lambda]
)

copd_predicted_data_lasso <- data.table(test_response = copd[!copd_train_ids, Rehosp_90_specific], pred = copd_test_preds_lasso[, 1])
hf_ukhd_predicted_data_lasso <- data.table(test_response = hf_ukhd[!hf_ukhd_train_ids, Rehosp_90_specific], pred = hf_ukhd_test_preds_lasso[, 1])
mi_ukhd_predicted_data_lasso <- data.table(test_response = mi_ukhd[!mi_ukhd_train_ids, Rehosp_90_specific], pred = mi_ukhd_test_preds_lasso[, 1])
osteo_predicted_data_lasso <- data.table(test_response = osteo[!osteo_train_ids, Rehosp_90_specific], pred = osteo_test_preds_lasso[, 1])
t2dm_predicted_data_lasso <- data.table(test_response = t2dm[!t2dm_train_ids, Rehosp_90_specific], pred = t2dm_test_preds_lasso[, 1])
vhf_predicted_data_lasso <- data.table(test_response = vhf[!vhf_train_ids, Rehosp_90_specific], pred = vhf_test_preds_lasso[, 1])

copd_roc_object_lasso <- roc(copd_predicted_data_lasso, "test_response", "pred", ci = TRUE)
hf_ukhd_roc_object_lasso <- roc(hf_ukhd_predicted_data_lasso, "test_response", "pred", ci = TRUE)
mi_ukhd_roc_object_lasso <- roc(mi_ukhd_predicted_data_lasso, "test_response", "pred", ci = TRUE)
osteo_roc_object_lasso <- roc(osteo_predicted_data_lasso, "test_response", "pred", ci = TRUE)
t2dm_roc_object_lasso <- roc(t2dm_predicted_data_lasso, "test_response", "pred", ci = TRUE)
vhf_roc_object_lasso <- roc(vhf_predicted_data_lasso, "test_response", "pred", ci = TRUE)

## Vergleichende ROC-Kurve LASSO
df <- data.frame(Sensitivity = mi_ukhd_roc_object_lasso$sensitivities,
  Specificity = mi_ukhd_roc_object_lasso$specificities,
  thresh = coords(mi_ukhd_roc_object_lasso, "best", transpose = TRUE
  , best.weights = c(1, sum(copd[!copd_train_ids, Rehosp_90_specific])/length(copd_train_ids))
  ) [1],
  best.x = coords(mi_ukhd_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(mi_ukhd_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("AMI, c-Statistic = ", mi_ukhd_roc_object_lasso$auc %>% round(digits = 2)))
df <- rbind(df,
  data.frame(Sensitivity = copd_roc_object_lasso$sensitivities,
  Specificity = copd_roc_object_lasso$specificities,
  thresh = coords(copd_roc_object_lasso, "best", transpose = TRUE) [1],
  best.x = coords(copd_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(copd_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("COPD, c-Statistic = ", copd_roc_object_lasso$auc %>% round(digits = 2))))
df <- rbind(df,
  data.frame(Sensitivity = t2dm_roc_object_lasso$sensitivities,
  Specificity = t2dm_roc_object_lasso$specificities,
  thresh = coords(t2dm_roc_object_lasso, "best", transpose = TRUE) [1],
  best.x = coords(t2dm_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(t2dm_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("DM, c-Statistic = ", t2dm_roc_object_lasso$auc %>% round(digits = 2))))
df <- rbind(df,
  data.frame(Sensitivity = hf_ukhd_roc_object_lasso$sensitivities,
  Specificity = hf_ukhd_roc_object_lasso$specificities,
  thresh = coords(hf_ukhd_roc_object_lasso, "best", transpose = TRUE) [1],
  best.x = coords(hf_ukhd_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(hf_ukhd_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("HF, c-Statistic = ", hf_ukhd_roc_object_lasso$auc %>% round(digits = 2))))
df <- rbind(df,
  data.frame(Sensitivity = osteo_roc_object_lasso$sensitivities,
  Specificity = osteo_roc_object_lasso$specificities,
  thresh = coords(osteo_roc_object_lasso, "best", transpose = TRUE) [1],
  best.x = coords(osteo_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(osteo_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("OS, c-Statistic = ", osteo_roc_object_lasso$auc %>% round(digits = 2))))
df <- rbind(df,
  data.frame(Sensitivity = vhf_roc_object_lasso$sensitivities,
  Specificity = vhf_roc_object_lasso$specificities,
  thresh = coords(vhf_roc_object_lasso, "best", transpose = TRUE) [1],
  best.x = coords(vhf_roc_object_lasso, "best", transpose = TRUE) [2],
  best.y = coords(vhf_roc_object_lasso, "best", transpose = TRUE) [3],
  Krankheit = paste("S/AF, c-Statistic = ", vhf_roc_object_lasso$auc %>% round(digits = 2))))

#tiff("./roc_lasso.tiff", units="in", width=10, height=5, res=300)
p2 <- ggplot(df, aes(x = 1-Specificity, y = Sensitivity)) +
  geom_segment(aes(x = 0, y = 0, xend = 1, yend = 1), size = 1, color = "black") +
  geom_line(size = 1.2) +
  scale_x_continuous(name = "Specificity",
  breaks = c(0, 0.25, 0.5, 0.75, 1),
  labels = c("1.00", "0.75", "0.50", "0.25", "0.00"),
  expand = expand_scale(mult = c(0, 0)), limits = c(0, 1)
  ) +
  ylab("Sensitivity") +
  theme_classic(base_size = 16) +
  scale_y_continuous(expand = expand_scale(mult = c(0, 0)), limits = c(0, 1)) +
  geom_segment(data = df, aes(y = best.y, yend = best.y, x = 0, xend = 1-best.x)) +
  geom_segment(data = df, aes(x = 1-best.x, xend = 1-best.x, y = 0, yend = best.y)) +
  #annotate(geom = "text", x = best.x, y = 0.2, label = paste(best.x, " specificity")) +
  geom_text(aes(x = 1.1-best.x, y = 0.2, label = paste(round(best.x, digits = 2)))) +
  geom_text(aes(y = best.y + 0.1, x = 0.2, label = paste(round(best.y, digits = 2)))) +
  geom_text(aes(y = 0.4, x = 0.75, label = paste("t = ", round(thresh, digits = 2)))) +
  facet_wrap(vars(Krankheit), scales = "free") +
  ggtitle("(B)")
#dev.off()

library(gridExtra)
tiff("./performance_A_B_600.tiff", units="in", width=10, height=10, res=600)
grid.arrange(p1, p2, nrow = 2)
dev.off()

#####
##### Gestacktes Modell #####
#####

## Trainiere Modelle

```

```

# copd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(copd)[, -filternumbers_copd], family = "binomial")
# hf_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(hf_ukhd)[, -filternumbers_hf_ukhd], family = "binomial")
# mi_ukhd_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(mi_ukhd)[, -filternumbers_mi_ukhd], family = "binomial")
# osteo_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(osteo)[, -filternumbers_osteo], family = "binomial")
# t2dm_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(t2dm)[, -filternumbers_t2dm], family = "binomial")
# vhf_glm_ml <- glm(Rehosp_90_specific ~ ., data = as.data.frame(vhf)[, -filternumbers_vhf], family = "binomial")

##
##
## Fit LASSO ##
##
##
set.seed(14102020)
copd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = copd)[, -1]
hf_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = hf_ukhd)[, -1]
mi_ukhd_X <- model.matrix(Rehosp_90_specific ~ .,
  data = mi_ukhd)[, -1]
osteo_X <- model.matrix(Rehosp_90_specific ~ .,
  data = osteo)[, -1]
t2dm_X <- model.matrix(Rehosp_90_specific ~ .,
  data = t2dm)[, -1]
vhf_X <- model.matrix(Rehosp_90_specific ~ .,
  data = vhf)[, -1]

copd_glm_pen_cv <- cv.glmnet(y = copd[, Rehosp_90_specific], x = copd_X[, -filternumbers_copd],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
hf_ukhd_glm_pen_cv <- cv.glmnet(y = hf_ukhd[, Rehosp_90_specific], x = hf_ukhd_X[, -filternumbers_hf_ukhd],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
mi_ukhd_glm_pen_cv <- cv.glmnet(y = mi_ukhd[, Rehosp_90_specific], x = mi_ukhd_X[, -filternumbers_mi_ukhd],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
osteo_glm_pen_cv <- cv.glmnet(y = osteo[, Rehosp_90_specific], x = osteo_X[, -filternumbers_osteo],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
t2dm_glm_pen_cv <- cv.glmnet(y = t2dm[, Rehosp_90_specific], x = t2dm_X[, -filternumbers_t2dm],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
vhf_glm_pen_cv <- cv.glmnet(y = vhf[, Rehosp_90_specific], x = vhf_X[, -filternumbers_vhf],
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")

copd_opt.lambda <- match(copd_glm_pen_cv$lambda.min, copd_glm_pen_cv$lambda)
hf_ukhd_opt.lambda <- match(hf_ukhd_glm_pen_cv$lambda.min, hf_ukhd_glm_pen_cv$lambda)
mi_ukhd_opt.lambda <- match(mi_ukhd_glm_pen_cv$lambda.min, mi_ukhd_glm_pen_cv$lambda)
osteo_opt.lambda <- match(osteo_glm_pen_cv$lambda.min, osteo_glm_pen_cv$lambda)
t2dm_opt.lambda <- match(t2dm_glm_pen_cv$lambda.min, t2dm_glm_pen_cv$lambda)
vhf_opt.lambda <- match(vhf_glm_pen_cv$lambda.min, vhf_glm_pen_cv$lambda)

## Fit auf alle Daten mit berechnetem Lambda

copd_glm_pen <- glmnet(y = copd[, Rehosp_90_specific], x = copd_X[, -filternumbers_copd],
  family = "binomial",
  alpha = 1,
  lambda = copd_glm_pen_cv$lambda)
hf_ukhd_glm_pen <- glmnet(y = hf_ukhd[, Rehosp_90_specific], x = hf_ukhd_X[, -filternumbers_hf_ukhd],
  family = "binomial",
  alpha = 1,
  lambda = hf_ukhd_glm_pen_cv$lambda)
mi_ukhd_glm_pen <- glmnet(y = mi_ukhd[, Rehosp_90_specific], x = mi_ukhd_X[, -filternumbers_mi_ukhd],
  family = "binomial",
  alpha = 1,
  lambda = mi_ukhd_glm_pen_cv$lambda)
osteo_glm_pen <- glmnet(y = osteo[, Rehosp_90_specific], x = osteo_X[, -filternumbers_osteo],
  family = "binomial",
  alpha = 1,
  lambda = osteo_glm_pen_cv$lambda)
t2dm_glm_pen <- glmnet(y = t2dm[, Rehosp_90_specific], x = t2dm_X[, -filternumbers_t2dm],
  family = "binomial",
  alpha = 1,
  lambda = t2dm_glm_pen_cv$lambda)
vhf_glm_pen <- glmnet(y = vhf[, Rehosp_90_specific], x = vhf_X[, -filternumbers_vhf],
  family = "binomial",
  alpha = 1,
  lambda = vhf_glm_pen_cv$lambda)

# Gesamtdatensatz erzeugen
# df <- rbind(copd, hf_ukhd, mi_ukhd, osteo, t2dm, vhf)
df_X <- rbind(copd_X, hf_ukhd_X, mi_ukhd_X, osteo_X, t2dm_X, vhf_X)

## Berechne Predictions für gestacktes Modell
# GLM
# copd_preds_glm <- predict(copd_glm_ml, as.data.frame(df)[, -filternumbers_copd])
# hf_ukhd_preds_glm <- predict(hf_ukhd_glm_ml, as.data.frame(df)[, -filternumbers_hf_ukhd])
# mi_ukhd_preds_glm <- predict(mi_ukhd_glm_ml, as.data.frame(df)[, -filternumbers_mi_ukhd])
# osteo_preds_glm <- predict(osteo_glm_ml, as.data.frame(df)[, -filternumbers_osteo])
# t2dm_preds_glm <- predict(t2dm_glm_ml, as.data.frame(df)[, -filternumbers_t2dm])
# vhf_preds_glm <- predict(vhf_glm_ml, as.data.frame(df)[, -filternumbers_vhf])

# LASSO

```

```

copd_preds_lasso <- predict(
  copd_glm_pen,
  df_X[, -filternumbers_copd],
  copd_glm_pen_cv$lambda[copd_opt.lambda]
)
hf_ukhd_preds_lasso <- predict(
  hf_ukhd_glm_pen,
  df_X[, -filternumbers_hf_ukhd],
  hf_ukhd_glm_pen_cv$lambda[hf_ukhd_opt.lambda]
)
mi_ukhd_preds_lasso <- predict(
  mi_ukhd_glm_pen,
  df_X[, -filternumbers_mi_ukhd],
  mi_ukhd_glm_pen_cv$lambda[mi_ukhd_opt.lambda]
)
osteo_preds_lasso <- predict(
  osteo_glm_pen,
  df_X[, -filternumbers_osteo],
  osteo_glm_pen_cv$lambda[osteo_opt.lambda]
)
t2dm_preds_lasso <- predict(
  t2dm_glm_pen,
  df_X[, -filternumbers_t2dm],
  t2dm_glm_pen_cv$lambda[t2dm_opt.lambda]
)
vhf_preds_lasso <- predict(
  vhf_glm_pen,
  df_X[, -filternumbers_vhf],
  vhf_glm_pen_cv$lambda[vhf_opt.lambda]
)

## Predictions zu neuem Datensatz mergen
df_glm <- data.frame(
  # copd = copd_preds_glm,
  # hf_ukhd = hf_ukhd_preds_glm,
  # mi_ukhd = mi_ukhd_preds_glm,
  # osteo = osteo_preds_glm,
  # t2dm = t2dm_preds_glm,
  # vhf = vhf_preds_glm,
  # Rehosp_90_all_cause =
  c(copd_90[order(FALL_ID), Rehosp_90_all_cause],
    hf_ukhd_90[order(FALL_ID), Rehosp_90_all_cause],
    mi_ukhd_90[order(FALL_ID), Rehosp_90_all_cause],
    osteo_90[order(FALL_ID), Rehosp_90_all_cause],
    t2dm_90[order(FALL_ID), Rehosp_90_all_cause],
    vhf_90[order(FALL_ID), Rehosp_90_all_cause])
)

df_lasso <- data.table(
  Copd = copd_preds_lasso %>% as.vector,
  hf_ukhd = hf_ukhd_preds_lasso %>% as.vector,
  mi_ukhd = mi_ukhd_preds_lasso %>% as.vector,
  osteo = osteo_preds_lasso %>% as.vector,
  t2dm = t2dm_preds_lasso %>% as.vector,
  vhf = vhf_preds_lasso %>% as.vector,
  Rehosp_90_all_cause =
  c(copd_90[order(FALL_ID), Rehosp_90_all_cause],
    hf_ukhd_90[order(FALL_ID), Rehosp_90_all_cause],
    mi_ukhd_90[order(FALL_ID), Rehosp_90_all_cause],
    osteo_90[order(FALL_ID), Rehosp_90_all_cause],
    t2dm_90[order(FALL_ID), Rehosp_90_all_cause],
    vhf_90[order(FALL_ID), Rehosp_90_all_cause])
)

#### Gestackte Modelle ####
# GLM
glm_glm_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_glm, family = "binomial")
glm_lasso_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_lasso, family = "binomial")

# Interpretation
summary(glm_glm_stacked)
summary(glm_lasso_stacked)

# LASSO
set.seed(14102020)
# # Auf GLM Predictions gebaut
df_glm <- df_glm %>% as.data.table()
df_lasso <- df_lasso %>% as.data.table()
#
df_X <- model.matrix(Rehosp_90_all_cause ~ .,
  data = df_glm)[, -1]
lasso_stacked_cv <- cv.glmnet(y = df_glm[, Rehosp_90_all_cause], x = df_X,
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
lasso_stacked_opt.lambda <- match(lasso_stacked_cv$lambda.min, lasso_stacked_cv$lambda)
lasso_stacked <- glmnet(y = df_glm[, Rehosp_90_all_cause], x = df_X,
  family = "binomial",
  alpha = 1,
  lambda = lasso_stacked_cv$lambda)
#
coef_glmnet <- coef(lasso_stacked)
coef_glmnet.opt.glm <- coef_glmnet[, lasso_stacked_opt.lambda]
#
# # Auf Lasso Predictions gebaut
df_X <- model.matrix(Rehosp_90_all_cause ~ .,
  data = df_lasso)[, -1]
lasso_stacked_cv <- cv.glmnet(y = df_lasso[, Rehosp_90_all_cause], x = df_X,
  family = "binomial",
  alpha = 1,
  nfolds = 5,
  type.measure = "auc")
lasso_stacked_opt.lambda <- match(lasso_stacked_cv$lambda.min, lasso_stacked_cv$lambda)
lasso_stacked <- glmnet(y = df_lasso[, Rehosp_90_all_cause], x = df_X,

```

```

#           family = "binomial",
#           alpha = 1,
#           lambda = lasso_stacked_cv$lambda)
#
# coef_glmnet <- coef(lasso_stacked)
# coef.glmnet.opt.lasso <- coef_glmnet[,lasso_stacked_opt.lambda)
#
## Interpretation
# coef.glmnet.opt.glm
# coef.glmnet.opt.lasso

# glm Tabelle
# df_final <- summary(glm_glm_stacked)$coefficients[, c(1,4)] %>% as.data.frame()
# df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)
#
# colnames(df_final) <- c("GLM", "p-Wert")
#
# df_final$GLM <- df_final$GLM %>% round(digits = 3)
# df_final$LASSO <- coef.glmnet.opt.glm %>% round(digits = 3)

# write.csv2(df_final, file = "./Tabelle_stacked_on_glm_coeffs.csv")

# lasso Tabelle
df_final <- summary(glm_lasso_stacked)$coefficients[, c(1,4)] %>% as.data.frame()
df_final$`Pr(>|z|)` <- df_final$`Pr(>|z|)` %>% round(digits = 3)

colnames(df_final) <- c("GLM", "p-Wert")

df_final$GLM <- df_final$GLM %>% round(digits = 3)
# df_final$LASSO <- coef.glmnet.opt.lasso %>% round(digits = 3)

#write.csv2(df_final, file = "./Tabelle_stacked_modell_coeffs.csv")

##
## Interpretation und Erklärungen für Ergebnisse
##

summary(glm_glm_stacked)
summary(glm_lasso_stacked)

coef.glmnet.opt.glm
coef.glmnet.opt.lasso

cor(df_glm)
cor(df_lasso)

#### Gestackte Modelle Performance Evaluation ####
## Train-/Testsplit für ROC
# df_glm <- df_glm %>% as.data.table()
df_lasso <- df_lasso %>% as.data.table()

set.seed(04112020)
train_ids <- sample(1:nrow(df_lasso), size = 0.7*nrow(df_lasso), replace = FALSE)
test_ids <- (1:nrow(df_lasso))[-train_ids]

# GLM
glm_glm_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_glm[train_ids,], family = "binomial")
glm_lasso_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_lasso[train_ids,], family = "binomial")

# test_preds_glm_glm <- predict(glm_glm_stacked, df_glm[test_ids,], type = "response")
# predicted_data_glm_glm <- data.table(test_response = df_glm[test_ids, Rehosp_90_all_cause], pred = test_preds_glm_glm)
# roc_object_glm_glm <- roc(predicted_data_glm_glm, "test_response", "pred")

test_preds_glm_lasso <- predict(glm_lasso_stacked, df_lasso[test_ids,], type = "response")
predicted_data_glm_lasso <- data.table(test_response = df_lasso[test_ids, Rehosp_90_all_cause], pred = test_preds_glm_lasso)
roc_object_glm_lasso <- roc(predicted_data_glm_lasso, "test_response", "pred", ci = TRUE)

# Lasso
# set.seed(14102020)
# # Auf GLM Predictions gebaut
# df_glm <- df_glm %>% as.data.table()
# df_lasso <- df_lasso %>% as.data.table()
#
# df_X_glm <- model.matrix(Rehosp_90_all_cause ~ .,
# data = df_glm)[, -1]
# lasso_stacked_cv_glm <- cv.glmnet(y = df_glm[train_ids, Rehosp_90_all_cause], x = df_X_glm[train_ids,],
# family = "binomial",
# alpha = 1,
# nfolds = 5,
# type.measure = "auc")
# lasso_stacked_opt.lambda_glm <- match(lasso_stacked_cv_glm$lambda.min,lasso_stacked_cv_glm$lambda)
# lasso_stacked_glm <- glmnet(y = df_glm[train_ids, Rehosp_90_all_cause], x = df_X_glm[train_ids,],
# family = "binomial",
# alpha = 1,
# lambda = lasso_stacked_cv_glm$lambda)
#
# # coef_glmnet <- coef(lasso_stacked)
# # coef.glmnet.opt.glm <- coef_glmnet[,lasso_stacked_opt.lambda)
#
# # Auf Lasso Predictions gebaut
# df_X_lasso <- model.matrix(Rehosp_90_all_cause ~ .,
# data = df_lasso)[, -1]
# lasso_stacked_cv_lasso <- cv.glmnet(y = df_lasso[train_ids, Rehosp_90_all_cause], x = df_X_lasso[train_ids,],
# family = "binomial",
# alpha = 1,
# nfolds = 5,
# type.measure = "auc")
# lasso_stacked_opt.lambda_lasso <- match(lasso_stacked_cv_lasso$lambda.min,lasso_stacked_cv_lasso$lambda)
# lasso_stacked_lasso <- glmnet(y = df_lasso[train_ids, Rehosp_90_all_cause], x = df_X_lasso[train_ids,],
# family = "binomial",
# alpha = 1,
# lambda = lasso_stacked_cv_lasso$lambda)
#

```



```

## # coef_glmnet <- coef(lasso_stacked)
## # coef_glmnet.opt.lasso <- coef_glmnet[,lasso_stacked_opt.lambda]
##
## # lasso_glm
## test_preds_lasso_glm <- predict(
##   lasso_stacked_glm,
##   df_X_glm[test_ids,],
##   type = "response",
##   lasso_stacked_cv_glm$lambda[lasso_stacked_opt.lambda_glm]
## )
##
## predicted_data_lasso_glm <- data.table(test_response = df_glm[test_ids, Rehosp_90_all_cause], pred = test_preds_lasso_glm[, 1])
##
## roc_object_lasso_glm <- roc(predicted_data_lasso_glm, "test_response", "pred")
##
## # lasso_lasso
## test_preds_lasso_lasso <- predict(
##   lasso_stacked_lasso,
##   df_X_glm[test_ids,],
##   type = "response",
##   lasso_stacked_cv_lasso$lambda[lasso_stacked_opt.lambda_lasso]
## )
##
## predicted_data_lasso_lasso <- data.table(test_response = df_lasso[test_ids, Rehosp_90_all_cause], pred = test_preds_lasso_lasso[, 1])
##
## roc_object_lasso_lasso <- roc(predicted_data_lasso_lasso, "test_response", "pred")

## ROC-Kurven

## df <- data.frame(specificity = c(roc_object_glm_glm$specificities,
##                                 roc_object_glm_lasso$specificities,
##                                 roc_object_lasso_glm$specificities,
##                                 roc_object_lasso_lasso$specificities),
##                 sensitivity = c(roc_object_glm_glm$sensitivities,
##                                 roc_object_glm_lasso$sensitivities,
##                                 roc_object_lasso_glm$sensitivities,
##                                 roc_object_lasso_lasso$sensitivities),
##                 Modell = c(rep("GLM on GLM", times = length(roc_object_glm_glm$sensitivities)),
##                            rep("GLM on LASSO", times = length(roc_object_glm_lasso$sensitivities)),
##                            rep("LASSO on GLM", times = length(roc_object_lasso_glm$sensitivities)),
##                            rep("LASSO on LASSO", times = length(roc_object_lasso_lasso$sensitivities))
##                 )
## )

df <- data.frame(specificity =
  roc_object_glm_lasso$specificities,

  sensitivity =
  roc_object_glm_lasso$sensitivities,

  Modell =
  rep("GLM on LASSO", times = length(roc_object_glm_lasso$sensitivities)),
  thresh = coords(roc_object_glm_lasso, "best", transpose = TRUE)[1],
  best.x = coords(roc_object_glm_lasso, "best", transpose = TRUE)[2],
  best.y = coords(roc_object_glm_lasso, "best", transpose = TRUE)[3]
)

# saveRDS(df, file = "./roc_data.rds")

tiff("./roc_stacked_model.tiff", units="in", width=10, height=5, res=300)
ggplot(df,
  aes(x = 1-specificity, y = sensitivity)) +
  geom_segment(aes(x = 0, y = 0, xend = 1, yend = 1), size = 1, color = "black") +
  geom_line(size = 1.2) +
  scale_x_continuous(name = "Specificity",
    breaks = c(0,0.25,0.5,0.75,1),
    labels = c("1.00", "0.75", "0.50", "0.25", "0.00"),
    expand= expand_scale(mult = c(0, 0)), limits = c(0, 1)
  ) +
  ylab("Sensitivity") +
  theme_classic() +
  geom_text(aes(x = 1.1-best.x, y = 0.2, label = paste(round(best.x, digits = 2)))) +
  geom_text(aes(y = best.y + 0.1, x = 0.2, label = paste(round(best.y, digits = 2)))) +
  geom_text(aes(y = 0.4, x = 0.75, label = paste("t = ", round(thresh, digits = 2)))) +
  geom_segment(data = df, aes(y = best.y, yend = best.y, x = 0, xend = 1-best.x)) +
  geom_segment(data = df, aes(x = 1-best.x, xend = 1-best.x, y = 0, yend = best.y)) +
  scale_y_continuous(expand= expand_scale(mult = c(0, 0)), limits = c(0, 1)
)
dev.off()

##
## Kreuzvalidierung
##
## Anzahl Folds
k <- 10

##
## Speichere Modellgüte der Folds
##
model_performance <- data.frame()

set.seed(05112020)
##
## Bestimmte Folds
##
performance_folds_global <- sample(1:k, replace = TRUE, size = nrow(df_lasso))

for(i in 1:k){
  ##
  ## Bestimme Fold Trainings- und Testdaten
  ##
  train_ids <- (1:nrow(df_lasso))[performance_folds_global != i]
  test_ids <- (1:nrow(df_lasso))[performance_folds_global == i]

```

```

##
##
## Fit glm ##
##
# glm_glm_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_glm[train_ids,], family = "binomial")
glm_lasso_stacked <- glm(Rehosp_90_all_cause ~ ., data = df_lasso[train_ids,], family = "binomial")

# test_preds_glm_glm <- predict(glm_glm_stacked, df_glm[test_ids,], type = "response")
# predicted_data_glm_glm <- data.table(test_response = df_glm[test_ids, Rehosp_90_all_cause], pred = test_preds_glm_glm)
# roc_object_glm_glm <- roc(predicted_data_glm_glm, "test_response", "pred")

test_preds_glm_lasso <- predict(glm_lasso_stacked, df_lasso[test_ids,], type = "response")
predicted_data_glm_lasso <- data.table(test_response = df_lasso[test_ids, Rehosp_90_all_cause], pred = test_preds_glm_lasso)
roc_object_glm_lasso <- roc(predicted_data_glm_lasso, "test_response", "pred")

print(paste0("GLM done. Fold:", i))

##
##
## Fit Lasso ##
##
##
# df_glm <- df_glm %>% as.data.table()
# df_lasso <- df_lasso %>% as.data.table()
#
# df_X_glm <- model.matrix(Rehosp_90_all_cause ~ .,
#                         data = df_glm)[, -1]
# lasso_stacked_cv_glm <- cv.glmnet(y = df_glm[train_ids, Rehosp_90_all_cause], x = df_X_glm[train_ids,],
#                                 family = "binomial",
#                                 alpha = 1,
#                                 nfolds = 5,
#                                 type.measure = "auc")
# lasso_stacked_opt.lambda_glm <- match(lasso_stacked_cv_glm$lambda.min, lasso_stacked_cv_glm$lambda)
# lasso_stacked_glm <- glmnet(y = df_glm[train_ids, Rehosp_90_all_cause], x = df_X_glm[train_ids,],
#                             family = "binomial",
#                             alpha = 1,
#                             lambda = lasso_stacked_cv_glm$lambda)
#
# # coef_glmnet <- coef(lasso_stacked)
# # coef.glmnet.opt.glm <- coef_glmnet[, lasso_stacked_opt.lambda]
#
# # Auf Lasso Predictions gebaut
# df_X_lasso <- model.matrix(Rehosp_90_all_cause ~ .,
#                           data = df_lasso)[, -1]
# lasso_stacked_cv_lasso <- cv.glmnet(y = df_lasso[train_ids, Rehosp_90_all_cause], x = df_X_lasso[train_ids,],
#                                   family = "binomial",
#                                   alpha = 1,
#                                   nfolds = 5,
#                                   type.measure = "auc")
# lasso_stacked_opt.lambda_lasso <- match(lasso_stacked_cv_lasso$lambda.min, lasso_stacked_cv_lasso$lambda)
# lasso_stacked_lasso <- glmnet(y = df_lasso[train_ids, Rehosp_90_all_cause], x = df_X_lasso[train_ids,],
#                              family = "binomial",
#                              alpha = 1,
#                              lambda = lasso_stacked_cv_lasso$lambda)
#
# # coef_glmnet <- coef(lasso_stacked)
# # coef.glmnet.opt.lasso <- coef_glmnet[, lasso_stacked_opt.lambda]
#
# # lasso_glm
# test_preds_lasso_glm <- predict(
#   lasso_stacked_glm,
#   df_X_glm[test_ids,],
#   type = "response",
#   lasso_stacked_cv_glm$lambda[lasso_stacked_opt.lambda_glm]
# )
#
# predicted_data_lasso_glm <- data.table(test_response = df_glm[test_ids, Rehosp_90_all_cause], pred = test_preds_lasso_glm[, 1])
#
# roc_object_lasso_glm <- roc(predicted_data_lasso_glm, "test_response", "pred")
#
# # lasso_lasso
# test_preds_lasso_lasso <- predict(
#   lasso_stacked_lasso,
#   df_X_lasso[test_ids,],
#   type = "response",
#   lasso_stacked_cv_lasso$lambda[lasso_stacked_opt.lambda_lasso]
# )
#
# predicted_data_lasso_lasso <- data.table(test_response = df_lasso[test_ids, Rehosp_90_all_cause], pred = test_preds_lasso_lasso[, 1])
#
# roc_object_lasso_lasso <- roc(predicted_data_lasso_lasso, "test_response", "pred")
#
##
##
## Results ##
##
##
model_performance <- rbind(
  model_performance,
  data.frame(
    # glm_glm = roc_object_glm_glm$auc,
    # glm_lasso = roc_object_glm_lasso$auc,
    # lasso_glm = roc_object_lasso_glm$auc,
    # lasso_lasso = roc_object_lasso_lasso$auc,
    fold = i
  )
)

print(paste0("LASSO done. Fold:", i))
}

```


Anlage 2: Liste mit Definition und Operationalisierung der Prädiktorvariablen.

Demografie:

Name	Ausprägungstyp	Erklärung	Operationalisierung
sex	binär	Geschlecht: männlich/weiblich	Information geliefert
age	numerisch	Alter in Jahren	Information geliefert
hosp_last_year	numerisch	Anzahl an Krankenhausaufenthalten in den letzten 365 Tagen vor Indexhospitalisierung	Gezählt aus gelieferten Datensätzen

Elixhauser Komorbiditäten: 0: nicht vorliegend, 1: vorliegend

Name	Ausprägungstyp	Erklärung	Operationalisierung
chf	binär	Congestive Heart Failure	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
carit	binär	Cardiac Ahythmias	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
valv	binär	Valvular Disease	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
pcd	binär	Pulmonary Circulation Disorders	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
pvd	binär	Peripheral Vascular Disorders	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
hypunc	binär	Hypertension, uncomplicated	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
hycp	binär	Hypertension, complicated	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
para	binär	Paralysis	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
ond	binär	Other Neurological Disorders	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
cpd	binär	Chronic Pulmonary Disease	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
diabunc	binär	Diabetes, uncomplicated	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
diabc	binär	Diabetes, complicated	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
hypothy	binär	Hypothyroidism	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
rf	binär	Renal Failure	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
ld	binär	Liver Disease	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
pud	binär	Peptic Ulcer Disease, Excluding Bleeding	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
aids	binär	AIDS/HIV	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
lymph	binär	Lymphoma	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
metacanc	binär	Metastatic Cancer	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
solidtum	binär	Solid Tumour, Without Metastasis	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
rheumd	binär	Rheumatoid Arthritis/Collaged Vascular Disease	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
coag	binär	Coagulopathy	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
obes	binär	Obesity	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
wloss	binär	Weight Loss	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores

fed	binär	Fluid and Electrolyte Disorders	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
blane	binär	Blood Loss Anaemia	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
dane	binär	Deficiency Anaemia	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
alcohol	binär	Alcohol Abuse	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
drug	binär	Drug Abuse	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
psycho	binär	Psychoses	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores
depre	binär	Depression	Operationalisiert mit "comorbidity" R-Paket für Elixhauser Scores

STOPP-Kriterien:

0: nicht vorliegend, 1: vorliegend

Name	Ausprägungstyp	Erklärung	Operationalisierung
B3	binär	Betablocker + Veradilta	ATC: C07 + ATC: C08DA01, C08DA51, C08DA81, C08GA02, C08DB01
B4	binär	Betablocker + (Brady oder Heartblock)	ATC: C07 + ICD-10: R001, I441, I442
B7	binär	Sulfons + Venousins + kein Heartfail + kein Liverdis + keinNephrotic + kein Chronic Renal	ATC: C03CA, C03CB + ICD-10: I872 + nicht ICD-10: I50 + nicht ICD-10: K72, K70, K74, K767 + nicht ICD-10: N04 + nicht ICD-10: N18 ATC: C03AA03, C03EA01, C09DA04, C03EA01, C09DA06, C07BB02, C09DA07, C07BB07, C09DA08, C09BA01, C09DX01, C09BA02, C09DX03, C09BA03, C09XA52, C09BA05, C09BA09, C09BA06, C09DA01, C09DA02, C09DA03, C09BA25 + (ICD-10: E876 oder ICD-10: E871 oder ICD-10: E8352 oder ICD-10: M10)
B8	binär	HCT + (Hypokal oder Hyponat oder Hypocal oder Gout)	ATC: C03CA, C03CB + ICD-10: I10, I15 + ICD-10: N393, N394
B9	binär	Sulfons + Hypertens + Inconti	ATC: C09A, C09B, C09C, C09D + ICD-10: E875
B11	binär	(ACE oder AT1) & Hyperkal	ATC: C03DA + ATC: C03DB, C09A, C09B, C09C, C09D
B12	binär	Aldonat + Potcons	ATC: G04BE03, G04BE08, G04BE09, R03DA04, R03DA07 + ICD-10: I50 + ICD-10: I95
B13	binär	PDE5 + Heartfail + Hypotens	ATC: B01AC06, B01AC08, B01AC36, B01AC56 + ICD-10: K22, K23, K24, K25, K26, K27, K28, K221, K250, K251, K252, K254, K255, K256, K260, K261, K262, K264, K265, K266, K270, K271, K272, K274, K275, K276, K280, K281, K282, K284, K285, K286 ATC: B01AC06, B01AC08, B01AC36, B01AC56, B01AC04, B01AC34, C01DX21, B01AA, B01AE, B01AF + ((ICD-10: D68, D69 + nicht ICD-10: D683, D693) oder ICD-10: I61, I62, I691, N02, R04, R31, K22, K23, K24, K25, K26, K27, K28, K221, K250, K251, K252, K254, K255, K256, K260, K261, K262, K264, K265, K266, K270, K271, K272, K274, K275, K276, K280, K281, K282, K284, K285, K286)
C2	binär	Salicyl + (UGIbleeding oder LGIbleeding)	ATC: B01AC06, B01AC08, B01AC36, B01AC56 + ATC: B01AA, B01AE, B01AF
C3	binär	Antithrombotic + ((bleedingdiathesis und kein NOTbleedingdiathesis) oder bleeding oder UGIbleeding oder LGIbleeding)	
C5	binär	Salicyl + OAC	

C6	binär	Antiplatelet + CVD + OAC	ATC: B01AC + ICD-10: I200, I201, I208, I209, I21, I22, I23, I24, I25, I65, I63, I66, I672, I693, I739, I74, I75 + ATC: B01AA, B01AE, B01AF
C7	binär	Ticlopidine	ATC: B01AC05
C10	binär	NSAID + OAC	ATC: M01A, N02BA + ATC: B01AA, B01AE, B01AF
C11	binär	NSAID + Antipatelet + kein PPI	ATC: M01A, N02BA + ATC: B01AC + nicht ATC: A02BC
D1	binär	TCA + (Glaucom oder Repol oder Urinary Retention)	ATC: N06AA, N06CA01 + (ICD-10: H402 oder ICD-10: I44, I472, I45 oder ICD-10: N44, N33) (ATC: N05AA01, N05AH02, N05AF01, N05AB02, N05AC04, N05AA03, N05AF05) + ICD-10: N44, N33
D3	binär	(AP Sel oder Clozapin) + Urinary Retention	ATC: N06AB + ICD-10: E871
D4	binär	SSRI + Hyponat	(ATC: N05A + nicht ATC: N05AH02 + nicht ATC: N05AH04) + ICD-10: G20, G21, G22, G23, G3185, G3182
D6	binär	(Parkinson oder Lewy)	ATC: N04A + ATC: N05A
D7	binär	Antichol Antipark + Ap	ATC: N04A + ICD-10: F05
D8	binär	Antichol Antipark + Delir	ATC: N04BA, N04BC + ICD-10: G250
D13	binär	Dopa + Tremor	ATC: C01AA05 + ICD-10: N18
E1	binär	Digoxin + Chronic Renal	ATC: B01AE + ICD-10: N18
E2	binär	Dthromb + Chronic Renal	ATC: B01AF + ICD-10: N18
E3	binär	Fxa + Chronic Renal	ATC: M01A + ICD-10: N18
E4	binär	NSAIDrenal + Chronic Renal	ATC: M04AC01 + ICD-10: N18
E5	binär	Colchicine + Chronic Renal	ATC: A10BA02 + ICD-10: N18
E6	binär	Metformin + Chronic Renal	ATC: N05AB04, A03FA01 + ICD-10: G20, G21, G22, G23
F1	binär	(PCP oder MCP) + Parkinson	ATC: N04A, B03AA, B03AB, B03AD, N02A, C08DA01, C08DA51 + ICD-10: K590
F3	binär	(Antichol Antipark oder Oral Iron oder Opiod oder Verapamil) + Constipation	ATC: R03DA + ICD-10: J44 + nicht ATC: R03A, R03BA, R03BB
G1	binär	Theophylline + COPD + kein COPD Combi	ATC: H02AB, H02BX + ICD-10: J44 + nicht ATC: R03BA, R03AK
G2	binär	GC Syst + COPD + kein GC Inh	ATC: R03BB + ICD-10: H402, N40, R33
G3	binär	Antimusc + (Glaucom oder Urinary Retention)	ATC: M01A, N02BA + nicht ATC: M01AH + ICD-10: K22, K23, K24, K25, K26, K27, K28, K221, K250, K251, K252, K254, K255, K256, K260, K261, K262, K264, K265, K266, K270, K271, K272, K274, K275, K276, K280, K281, K282, K284, K285, K286 + (nicht ATC: A02BC oder nicht ATC: A02BA)
H1	binär	NSAID + kein Coxib + (UGIbleeding oder LGIbleeding) + (kein PPI oder kein H2)	ATC: H02AB, H02BX + ICD-10: M15, M16, M17, M18, M19, M47
H5	binär	GC Syst + Osteoarthritis	ATC: M01AH + ICD-10: I200, I201, I208, I209, I21, I22, I23, I24, I25, I65, I63, I66, I672, I693, I739, I74, I75
H7	binär	Coxib + CVD	ATC: M01A, N02BA + ATC: H02AB, H02BX + nicht ATC: A02BC
H8	binär	NSAID + GC Syst + kein PPI	

H9	binär	Oral Bisphospho + UGIbleeding Uro Antimusc + (Glaucom oder Urinary Retention)	ATC: M05BA04, M05BA06, M05BA07, M05BB + ICD-10: K22, K23, K24, K25, K26, K221, K250, K251, K252, K254, K255, K256, K260, K261, K262, K264, K265, K266
I1	binär		ATC: G04BD + ICD-10: H402, N40, R33
I2	binär	Alphablocker + (Orthostase oder Syncope)	ATC: G04CA, C02CA04, C02CA06, C02KD01 + ICD-10: I951, R55
J1	binär	Sulfonylorea + DM2	ATC: A10BB01, A10BB02, A10BB12 + ICD-10: E11
J2	binär	Thiazolidene + Heartfail	ATC: A10BG, A10BD05, A10BD06, A10BD09, A10BD12 + ICD-10: I50
J3	binär	Betablocker + (DM1 oder DM2) + Hypoglyc	ATC: C07 + ICD-10: E10, E11 + ICD-10: E160, E161, E162
J4	binär	Oestrogens + (VTE oder BRCA)	ATC: G03AA, G03AB, G03C, G03EA, G03EB, G03F, G03HB, L02AA + ICD-10: I26, C50
K1	binär	Benzos + Fall	ATC: N05CD, N05BA, N03AE + ICD-10: Z9181, R296, S01, S02, S09, S22, S32, S52, S62, S82, S92
K2	binär	AP + Fall	ATC: N05A, ICD-10: Z9181, R296, S01, S02, S09, S22, S32, S52, S62, S82, S92
K3	binär	Vasodil + Orthostase	ATC: C02DB01, C02LG01, C02LG51, C02DB02, C02LG02, C02DC01, G04CA, C02CA04, C02CA06, C02KD01, C08, C09 + ICD-10: I951
K4	binär	Zsub + Fall	ATC: N05CF + ICD-10: Z9181, R296, S01, S02, S09, S22, S32, S52, S62, S82, S92
M1	binär	(Antichol Antipark oder Uro Antimusc) + TCA	ATC: N04A, G04BD + ATC: N06AA, N06CA01

START-Kriterien:

0: nicht vorliegend, 1: vorliegend

Name	Ausprägungstyp	Erklärung	Operationalisierung
A1	binär	AF + kein OAC	ICD-10: I481, I482, I4891 + nicht ATC: B01AA, B01AE, B01AF
A2	binär	AF + kein OAC + kein Salicyl	ICD-10: I481, I482, I4891 + nicht ATC: B01AA, B01AE, B01AF + nicht: ATC: B01AC06, B01AC08, B01AC36, B01AC56
A3	binär	CVD + kein Antiplatelet	ICD-10: I200, I201, I208, I209, I21, I22, I23, I24, I25, I65, I63, I66, I672, I693, I739, I74, I75 + nicht ATC: B01AC
A5	binär	CVD + kein Statin	ICD-10: I200, I201, I208, I209, I21, I22, I23, I24, I25, I65, I63, I66, I672, I693, I739, I74, I75 + nicht ATC: C10AA, C10BA, C01BX
A6	binär	(Syst Heartfail oder CAD) + kein (ACE oder AT1)	ICD-10: I501, I509, I200, I201, I208, I209, I21, I22, I23, I24, I25, I65, I63, I66, I672, I693, I739, I74, I75 + nicht ATC: C09A, C09B, C09C, C09D
A7	binär	CAD + kein Betablocker	ICD-10: I200, I201, I208, I209, I21, I22, I23, I24, I25 + nicht ATC: C07
A8	binär	Syst Heartfail + kein Betablocker Appr	ICD-10: I501, I509 + nicht ATC: C07AB07, C07AB12, C07AB02, C07AG02
C1	binär	Parkinson + kein Dopa	ICD-10: G20, G21, G22, G23 + nicht ATC: N04BA, N04BC

C2	binär	Depression + kein Non TCA	ICD-10: F32, F330, F331, F332, F333, F338, F339 + nicht ATC: N06AB, N06AP, N06AX
C3	binär	Dementia + kein AChE	ICD-10: F00, F01, F02, F03, G30, G31 + nicht ATC: N06DA ICD-10: K22, K23, K24, K25, K26, K27, K28, K221, K250, K251, K252, K254, K255, K256, K260, K261, K262, K264, K265, K266, K270, K271, K272, K274, K275, K276, K280, K281, K282, K284, K285, K286 + nicht ATC: A02BC
D1	binär	(UGIBleeding oder LGIBleeding) + kein PPI	ICD-10: M05, M06 + nicht ATC: L04AX01, L04AX03, L04AA13, L04AD01, A07EC01, P01BA, L04AC03, L04AC07, L04AB, L04AA24, L01XC02, M01CB01, M01CB03, M01AA, M01AB, M01AC, M01AE, M01AG, M01AH, M01AX01, N02BA
E1	binär	RA + kein DMARD	ICD-10: M80, M81, M82 + nicht ATC: M05BA, M05BB, M05BX, H05AA02
E4	binär	Osteoporosis + kein (Bisphospho oder Antires oder Teriparatide)	ICD-10: M10 + nicht ATC: M04AA
E6	binär	Gout + kein XO1	ICD-10: E10, E11 + ICD-10: R80, N06, N391 + nicht ATC: C09A, C09B, C09C, C09D
F1	binär	((DM2 oder DM1) + Proteinuria) + kein (ACE oder AT1)	