Dissertation

Standardizing Clinical Predictive Modeling

Standardizing Development, Validation, and Interpretation of Clinical Prediction Models

Dissertation submitted in partial fulfillment of the
requirements for the attainment of the degree
"Doktor der Ingenieurwissenschaften"
(Dr.-Ing.)
in the scientific discipline
"Internet Technologies and Systems"

submitted to the Digital Engineering Faculty of the
University of Potsdam

by
Harry Freitas da Cruz

Hasso Plattner Institute at the University of Potsdam

Submission: September 03, 2020
Defense: April 23, 2021

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-51496
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-514960

To my father.

Disclaimers

Parts of the author's published work were partially reused in this thesis, as follows.

Chapter 2 and **Chapter 4**: Parts of these chapters are subject to IARIA copyright. According to the publisher's website, reproduction is permitted for anyone to copy, distribute, and display the copyrighted work, with no prior permission required, if the copyright notice is provided:

© 2019 IARIA. Reprinted with permission from: HF da Cruz, S. Horschig, C Nusshag, and MP Schapranow. "Knowledge Distillation of Machine Learning Models in the Prediction of Hemodialysis Outcomes". In: *International Journal on Advances in Life Sciences* 11.1 (2019), pages 33–43.

Chapter 3: Parts of this chapter are subject to IEEE copyright. According to IEEE publication rights, obtained via RightsLink®, a formal reuse license is not required for individuals working on a thesis. However, the following copyright notice should be included:

© 2019 IEEE. Reprinted with permission from: HF da Cruz, B Bergner, O. Konak, F. Schneider, P. Bode, C. Lempert, and MP Schapranow. "MORPHER–A Platform to Support Modeling of Outcome and Risk Prediction in Health Research". In: *Proc. of the 19th IEEE Intl. Conf. on Bioinformatics and Bioengineering*. IEEE. 2019, pages 462–469.

Chapter 4: Excerpts of three publications were used in this chapter. For one of the papers, the copyrights are owned by Elsevier. The appropriate rights for reuse in a thesis were obtained via RightsLink®. For another one, rights are owned by Elsevier, but authors "retain the right to include it in a thesis or dissertation, provided it is not published commercially". Copyrights to the other publication belong to SCITEPRESS. Permission for reuse was granted after written request. The follow copyright notices therefore apply, respectively:

© 2021 Elsevier B.V. Reprinted with permission from: HF da Cruz, B Pfahringer, F Schneider, A Meyer, EP Böttinger, and MP Schapranow. "Using Interpretability Approaches to Update "Black-Box" Clinical Prediction Models: an External Validation Study in Nephrology". In: *Journal of Artificial Intelligence in Medicine* (2021).

© 2019 Springer Nature. Reprinted with permission from: HF da Cruz, B Pfahringer, F Schneider, A Meyer, and MP Schapranow. "External Validation of a "Black-Box" Clinical Predictive Model in Nephrology: Can Interpretability Methods Help Illuminate Performance Differences?" In: *Proc. of the 17th Conf. on Artificial Intelligence in Medicine*. Springer. 2019, pages 191–201.

© 2019 SCITEPRESS. Reprinted with permission from: HF da Cruz, F. Schneider, and MP Schapranow. "Prediction of Acute Kidney Injury in Cardiac Surgery Patients: Interpretation using Local Interpretable Model-agnostic Explanations". In: *Proc. of the 12th Intl. Conf. on Biomedical Engineering Systems and Technologies*. Volume 5. Prague, Czech Republic, 2019, pages 380–387. ISBN: 978-989-758-353-7.

Whenever necessary, in addition, the respective co-authors' contributions were detailed in the final section of each of the aforementioned chapters.

Publications

The following original publications have been completed during the doctoral degree and this thesis relies partially on their content.

[1] HF da Cruz, B Grasnick, H Dinger, F Bier, and C Meinel. "Early Detection of Acute Kidney Injury with Bayesian Networks". In: *Proc. of the 7th Intl. Symp. on Semantic Mining in Biomedicine*. Potsdam, Germany, 2016, pages 29–36.

[2] HF da Cruz, S Horschig, C Nusshag, and MP Schapranow. "Prediction of Patient Outcomes after Renal Replacement Therapy in Intensive Care". In: *Proc. of the 3rd Intl. Conf. on Informatics and Assistive Technologies for Health-Care, Medical Support and Wellbeing*. 2018, pages 197–204.

[3] M Oleynik, E Faessler, AM Sasso, A Kappattanavar, B Bergner, HF da Cruz, JP Sachs, S Datta, and EP Böttinger. "HPI-DHC at TREC 2018 Precision Medicine Track." In: *TREC*. 2018.

[4] O Konak, HF da Cruz, M Thiele, D Golla, and MP Schapranow. "An Information and Communication Platform Supporting Analytics for Elderly Care." In: *Proc. of the 5th Intl. Conf. on Information and Communication Technologies for Ageing Well and e-Health*. 2019, pages 197–204.

[5] HF da Cruz, F. Schneider, and MP Schapranow. "Prediction of Acute Kidney Injury in Cardiac Surgery Patients: Interpretation using Local Interpretable Model-agnostic Explanations". In: *Proc. of the 12th Intl. Conf. on Biomedical Engineering Systems and Technologies*. Volume 5. Prague, Czech Republic, 2019, pages 380–387. ISBN: 978-989-758-353-7.

[6] HF da Cruz, S. Horschig, C Nusshag, and MP Schapranow. "Knowledge Distillation of Machine Learning Models in the Prediction of Hemodialysis Outcomes". In: *International Journal on Advances in Life Sciences* 11.1 (2019), pages 33–43.

[7] HF da Cruz, B Bergner, O. Konak, F. Schneider, P. Bode, C. Lempert, and MP Schapranow. "MORPHER–A Platform to Support Modeling of Outcome and Risk Prediction in Health Research". In: *Proc. of the 19th IEEE Intl. Conf. on Bioinformatics and Bioengineering*. IEEE. 2019, pages 462–469.

[8] S Datta, A Schraplau, HF da Cruz, JP Sachs, F Mayer, and EP Böttinger. "A Machine Learning Approach for Non-Invasive Diagnosis of Metabolic Syndrome". In: *Proc. of the 19th IEEE Intl. Conf. on Bioinformatics and Bioengineering*. IEEE. 2019, pages 933–940.

[9] HF da Cruz, B Pfahringer, F Schneider, A Meyer, and MP Schapranow. "External Validation of a "Black-Box" Clinical Predictive Model in Nephrology: Can Interpretability Methods Help Illuminate Performance Differences?" In: *Proc. of the 17th Conf. on Artificial Intelligence in Medicine*. Springer. 2019, pages 191–201.

[10] A Kappattanavar, HF da Cruz, B Arnrich, and EP Böttinger. "Position Matters: Sensor Placement for Sitting Posture Classification." In: *Proc. of the 8th IEEE Intl. Conf. on Healthcare Informatics*. 2020.

[11] HF da Cruz, B Pfahringer, F Schneider, A Meyer, EP Böttinger, and MP Schapranow. "Using Interpretability Approaches to Update "Black-Box" Clinical Prediction Models: an External Validation Study in Nephrology". In: *Journal of Artificial Intelligence in Medicine* (2021).

Acknowledgments

"No man is an island", by John Donne (*Devotions upon Emergent Occasions*, 1624) is an often-cited aphorism to signify our interdependence, our reliance on those around us. However, it can be metaphysically argued that we are indeed islands, but *islands to ourselves*, to the extent that no other being can really 'know' what it is to be 'one self'. As such, it makes more sense to speak in terms of *archipelagos* of interconnected islands. Granted, this may as well sound cheap and shallow, even corky, but could not be more true than for the work of elaborating a thesis.

Throughout my PhD journey, I inhabited different such archipelagos; shaken, as it were, by the movements of the 'plate tectonics' of life. Next is my attempt humble and by necessity imperfect attempt at acknowledging the people – or other islands – who have contributed directly or indirectly for this thesis.

Family, who provided me with the gift of life, and whose unwavering support never failed to furnish me with the needed peace of mind to complete yet another chapter of my academic life.

Friends, from Leipzig, Potsdam and Berlin, but rather from many different corners of the world – who provided me with the good measure of sanity in the most difficult moments. They are too many to mention by name, but the Komets know what they mean to me. Thank your for your friendship.

Colleagues and students, at the HPI for their comradeship and hands-on help during all these years. Particularly our friends from the In-Memory Databases Group, Orhan Konak, Benjamin Bergner, Florian Borchert, Milena Kraus, Cindy Perscheid, Mariana Neves (EPIC), and from the Personalized Medicine Group, Jan Philipp Sachs, Suparno Datta, Cornelia Philippson and Jasmin Cirilo. Also many thanks to our HPI students, Philipp Bode, Tom Martensen, Frederic Schneider, Siegfried Horschig and Conrad Lempert.

Advisors and mentors, particularly my first supervisor, Prof. Dr. Christoph Meinel, who welcomed me with open arms at the HPI and provided me all the needed guidance, always available to listen, help and direct. Also I thank Dr.-Ing. Matthieu Schapranow for his advice and guidance, from whom I learned so much, for which I am grateful. Furthermore, I acknowledge Prof. Dr. Erwin's Böttinger's contributions, help and advice towards the realization of this work. Finally, Dr. Christina Schröder, Dr. Eva Ehrentreich-Förster and Prof. Dr. Frank Bier, Fraunhofer IZI-BB, who were the first to believe in this project and whole-heartedly supported me in the very begining of this journey and did not let me quit altogether, even when I had made up my mind to do so. Thank you for your advice.

Project partners, who were always so generous with their time, providing invaluable insights. Specially, Dr. med. Christian Nußhag (University Hospital Heidelberg), Prof. Dr. Alexander Meyer (German Heart Center Berlin), Prof. Dr. Klemens Budde, Dr. med. Wiebke Düttmann-Rehnolt (Charité), and Dr. Rolland Roller (Deutsches Forschungszentrum für Künstliche Intelligenz).

Finally, I am deeply thankful for my loving partner Ariane Morassi Sasso, together with Yogi and Gatsby, the furrier members of the family. Morchen, thank you for being such an incredible inspiration. Thank you for being awesome. Extra credit goes to Gatsby for not licking me – and for staying alive – while writing this thesis.

Contents

List of Figures

List of Tables

Abstract

An ever-increasing number of prediction models is published every year in different medical specialties. Prognostic or diagnostic in nature, these models support medical decision making by utilizing one or more items of patient data to predict outcomes of interest, such as mortality or disease progression. While different computer tools exist that support clinical predictive modeling, I observed that the state of the art is lacking in the extent to which the needs of research clinicians are addressed. When it comes to model development, current support tools either 1) target specialist data engineers, requiring advanced coding skills, or 2) cater to a general-purpose audience, therefore not addressing the specific needs of clinical researchers. Furthermore, barriers to data access across institutional silos, cumbersome model reproducibility and extended experiment-to-result times significantly hampers validation of existing models. Similarly, without access to interpretable explanations, which allow a given model to be fully scrutinized, acceptance of machine learning approaches will remain limited. Adequate tool support, i.e., a software artifact more targeted at the needs of clinical modeling, can help mitigate the challenges identified with respect to model development, validation and interpretation. To this end, I conducted interviews with modeling practitioners in health care to better understand the modeling process itself and ascertain in what aspects adequate tool support could advance the state of the art. The functional and non-functional requirements identified served as the foundation for a software artifact that can be used for modeling outcome and risk prediction in health research. To establish the appropriateness of this approach, I implemented a use case study in the Nephrology domain for acute kidney injury, which was validated in two different hospitals. Furthermore, I conducted user evaluation to ascertain whether such an approach provides benefits compared to the state of the art and the extent to which clinical practitioners could benefit from it. Finally, when updating models for external validation, practitioners need to apply feature selection approaches to pinpoint the most relevant features, since electronic health records tend to contain several candidate predictors. Building upon interpretability methods, I developed an explanation-driven recursive feature elimination approach. This method was comprehensively evaluated against state-of-the art feature selection methods. Therefore, this thesis' main contributions are three-fold, namely, 1) designing and developing a software artifact tailored to the specific needs of the clinical modeling domain, 2) demonstrating its application in a concrete case in the Nephrology context and 3) development and evaluation of a new feature selection approach applicable in a validation context that builds upon interpretability methods. In conclusion, I argue that appropriate tooling, which relies on standardization and parametrization, can support rapid model prototyping and collaboration between clinicians and data scientists in clinical predictive modeling.

Zusammenfassung

Die Zahl der jährlich veröffentlichten Vorhersagemodelle in verschiedenen medizinischen Fachrichtungen nimmt stetig zu. Solche prognostischen oder diagnostischen Modelle helfen bei der medizinischen Entscheidungsfindung, indem sie zum Beispiel Vorhersagen zur Mortalität oder zum Krankheitsverlauf erlauben. Obwohl bereits Softwarewerkzeuge für die Entwicklung klinischer Vorhersagemodelle existieren, genügt der Stand der Technik noch immer nicht den Anforderungen klinischer Wissenschaftler. So kommt es, dass aktuelle Softwarewerkzeuge zur Modellentwicklung entweder 1) auf die Anforderungen von Datenwissenschaftlicher zugeschnitten sind und folglich Programmierkenntnisse voraussetzen, oder 2) zu generisch sind und somit den tatsächlichen Anforderungen klinischer Wissenschaftler nicht gerecht werden. Überdies wird die Reproduzierbarkeit der Modelle sowie die Durchführung und Validierung von Experimenten durch verteilte Datenbestände und Informationen, sogenannte Datensilos, stark eingeschränkt. Ähnlich verhält es sich bei der Akzeptanz von Modellen des maschinellen Lernens, welche ohne interpretierbare Erklärungen von Vorhersagen kaum gegeben sein dürfte. Eine auf diese Anforderungen klinischer Modellbildung ausgerichtete Softwarelösung kann dabei helfen, die identifizierten Herausforderungen bezüglich Modellentwicklung, -validierung und -interpretation zu bewältigen und die Akzeptanz und Nutzung unter Klinikern zu stärken. Um den Modellierungsprozess zu verstehen und zu eruieren, in welchem Ausmaß eine angemessene Softwarelösung den Stand der Technik voranbringen könnte, wurden im Zuge dieser Arbeit Interviews mit praktizierenden Modellierern im Gesundheitsbereich geführt. Daraus leiten sich funktionale und nichtfunktionale Anforderungen ab, die als Grundlage eines Softwareartefaktes für die Modellierung von Outcome- und Risikovorhersagen in der Gesundheitsforschung verwendet wurden. Um die Eignung meines Ansatzes zu verifizieren, habe ich den Anwendungsfall „akutes Nierenversagen“ im Bereich der Nephrologie in zwei verschiedenen Krankenhäusern betrachtet und validiert. Darüber hinaus wurde eine Nutzerevaluierung durchgeführt um herauszufinden, ob ein solcher Ansatz im Vergleich zum Stand der Technik Vorteile bietet und inwieweit klinische Praktiker davon profitieren können. Außerdem müssen praktizierende Kliniker bei der Aktualisierung von Modellen für die externe Validierung Ansätze zur Merkmalsselektion anwenden, da elektronische Gesundheitsakten in der Regel mehrere erklärende Merkmale enthalten. Aufbauend auf Methoden zur Interpretierbarkeit habe ich einen erklärungsorientierten rekursiven Eliminierungsansatz entwickelt. Dieser neue Ansatz wurde umfassend mit Standardverfahren der Merkmalsselektion verglichen. Daraus leiten sich folgende Forschungsbeiträge dieser Arbeit ab: 1) Entwurf und Entwicklung eines Softwareartefakts, welches auf die speziellen Bedürfnisse der klinischen Modellierungsdomäne zugeschnitten ist, 2) Demonstration seiner Anwendbarkeit für das konkrete Fallbeispiel „akutes Nierenversagen“ und 3) Entwicklung und Evaluierung eines neuen, auf Interpretierbarkeitsmethoden basierenden Ansatzes, zur Merkmalsselektion in einem Validierungskontext. Zusammenfassend ist zu folgern, dass ein geeignetes auf Standardisierung und Parametrisierung gestütztes Tool die schnelle prototypische Entwicklung und die Zusammenarbeit von Klinikern und Datenwissenschaftlern an klinischen Vorhersagemodellen unterstützen kann.

Resumo

Um número cada vez maior de modelos preditivos é publicado a cada ano em diferentes especialidades médicas. Prognósticos ou diagnósticos em natureza, esses modelos apoiam a tomada de decisão médica, utilizando dados de pacientes para prever resultados de interesse, tais como mortalidade ou progressão da doença. Embora existam diferentes ferramentas computacionais que suportam a modelagem preditiva clínica, observei que falta ainda ao estado da arte atender às necessidades de pesquisadores clínicos. Quando se trata de desenvolvimento de modelos, as ferramentas de apoio atuais ou 1) visam engenheiros de dados especializados, exigindo habilidades avançadas de programação, ou 2) atendem a um público geral, não focando, portanto, nas necessidades específicas de pesquisadores clínicos. Além disso, barreiras ao acesso aos dados em silos institucionais, dificuldade na reprodutibilidade dos modelos e tempos prolongados de experimentação e resultado dificultam significativamente a validação dos modelos existentes. Da mesma forma, sem acesso a explicações interpretáveis, que permitam que um determinado modelo seja totalmente escrutinado, a aceitação das abordagens de aprendizado de máquina permanecerá limitada. O suporte adequado de ferramentas, ou seja, um artefato de software mais direcionado às necessidades da modelagem clínica, pode ajudar a mitigar os desafios identificados com relação ao desenvolvimento, validação e interpretação de modelos. Para este fim, realizei entrevistas com profissionais de modelagem na área de saúde para entender melhor o processo de modelagem em si e verificar em que aspectos o suporte adequado de uma ferramenta poderia fazer avançar o estado da arte. Os requisitos funcionais e não funcionais identificados serviram como base para um artefato de software que pode ser usado para modelagem de resultados e previsão de risco em pesquisas em saúde. Para estabelecer a adequação dessa abordagem, implementei um estudo de caso de uso no domínio da Nefrologia para lesão renal aguda, que foi validado em dois hospitais diferentes. Além disso, realizei uma avaliação com usuários para verificar se tal abordagem oferece benefícios em comparação com o estado da arte e até que ponto os profissionais clínicos poderiam se beneficiar com ela. Finalmente, ao atualizar modelos para validação externa, os profissionais precisam aplicar abordagens de seleção de preditores para identificar aqueles mais relevantes, já que os registros eletrônicos de saúde tendem a conter vários possíveis candidatos. Com base em métodos de interpretabilidade, desenvolvi uma abordagem de eliminação recursiva de preditores orientada por explicações. Este método foi avaliado de forma abrangente em relação aos métodos de seleção de preditores mais modernos. Portanto, as principais contribuições desta tese são três, a saber: 1) projetar e desenvolver um artefato de software adaptado às necessidades específicas do domínio da modelagem clínica, 2) demonstrar sua aplicação em um caso concreto no contexto da Nefrologia e 3) desenvolvimento e avaliação de uma nova abordagem de seleção de preditores aplicável em um contexto de validação que se baseia em métodos de interpretabilidade. Em conclusão, argumento que uma ferramenta apropriada, que se baseia na padronização e parametrização, pode apoiar a prototipagem rápida de modelos e a colaboração entre profissionais clínicos e cientistas de dados na modelagem preditiva clínica.

1. Introduction

"A man on a thousand mile walk has to forget his goal and say to himself every morning, 'Today I'm going to cover twenty-five miles and then rest up and sleep'."

—Leo Tolstoy, *War and Peace (1869)*

How useful are computers in a healthcare setting? Undoubtedly, information systems have dramatically impacted the way care is delivered to patients worldwide. While initially restricted to automating administrative routines such as billing, now Electronic Health Record (EHR) support a wide range of core processes in patient care, leaving behind a massive data footprint. In this context, a similar question arises: can EHR data be useful to help improve care delivery to patients? To what extent can mathematical models learn underlying patterns in this data footprint which provide hints to healthcare providers, e.g., helping to identify deadly complications before there are obvious signs of it? This is the driving question of this thesis and provides the needed framing for the issues that will be dealt with in the subsequent chapters.

1.1. Context

The field concerned with answering the preceding question is called clinical predictive modeling. It deals with applying and developing mathematical-analytical tools to provide risk estimates of patient risk for either a) the presence of a disease (diagnosis) or b) future outcome (prognosis) [4]. Clinical Prediction Models (CPMs), prediction rules, clinical scores or more specifically prognostic models or simply *models* are "tools for helping decision making that combine two or more items of patient data to predict clinical outcomes" [5]. Such tools can be based on simple scoring systems, logistic regression or more sophisticated modeling algorithms, such as neural networks. We therefore use the term 'model' or CPM for a clinical instrument with pre-defined inputs and outputs as opposed to a 'modeling algorithm', a general-purpose mathematical construct that can be used to develop CPMs.

CPMs are widely used in the medical context. Examples of application areas include predicting patient outcomes such as mortality and hospital readmission, as well as deriving risk profiles [6]. Such models can be either diagnostic or prognostic in nature, the former referring to models which aim to identify specific patient conditions while the latter aims to predict how patients are likely to respond to a given treatment or to pinpoint disease progression [7]. They sit at the cornerstone of modern health care provision, spanning across different areas of medicine. For example, in public health, predictive models help decision makers shape public policies, e.g., to stave off epidemics building on seasonal disease trends. In intensive care, prognostic models support physicians in devising appropriate care plans according to risk profiles. By way of illustration, the widely employed clinical score for intensive care, Acute Physiology and Chronic Health Enquiry (APACHE-II), is an early instance of a clinical score which found ample acceptance in clinical practice, being included in guidelines for patient care [8].

1.2. Motivation

While predictive modeling indeed shows much promise for clinical practice, its adoption remains limited, including current research calling into question whether there is any benefit at all to the surge in clinical models, provokingly deeming 'most clinical scores useless' [9]. Dekker et al. outline three core reasons for this skepticism: (1) flaws and pitfalls in model development, (2) flaws and pitfalls in model validation and (3) lack of impact studies, i.e., assessing their use in Randomized Controlled Trials (RCTs). In addition to Dekker et al.'s assessment, I identify the opaqueness and lack of interpretability of 'black-box' Machine Learning (ML) algorithms as a further factor hindering the adoption of CPMs. As such, I am primarily concerned with challenges regarding development, validation and interpretation of prediction models, a selection of which is presented below.

1.2.1. Need for Standardization in Development and Validation

For clinical prediction modeling using ML, research clinicians must undergo a lengthy process of 1) data extraction and preparation 2) modeling and 3) validation. Because these researchers are often neither statisticians nor machine learning experts, important methodological steps are ignored or not reported at all, with considerable inconsistency, e.g., in the reporting of performance metrics [10]. The alternative is to 'outsource' development to data experts, but these often lack the needed medical knowledge and might develop models that are sound statistically but questionable from a clinical perspective. Besides, model reporting usually is limited to common metrics, such as sensitivity and specificity, ignoring those concerned with calibration and clinical usefulness. Given this lack of transparency in methods and reporting, reproducibility is severely hampered. Consequently, to date, not only is the development of clinical prediction models a time-consuming endeavor but it is also potentially riddled with flaws, both from a clinical and methodological standpoint [11]. This is so because the whole process of developing clinical predictive models is highly complex, with several different parameters to be chosen and tuned.

Furthermore, even if the CPMs developed were clinically and methodologically sound, they would have to be externally validated, that is, tested on a population other than the one they were initially trained on. Only then could it be demonstrated satisfactorily that the models developed are generally applicable and not biased. Since institutions cannot share patient data with the outside unless cumbersome legal and technical hurdles are worked out, most models developed end up not being validated externally and ultimately remain within the confines of one single institution, with limited generalizability [10]. Worryingly, model validation seldom takes place at all. Evidence for this claim is a review of models for Acute Kidney Injury (AKI), out of 53 CPMs only five were externally validated [10]. Reasons for this are manifold, but most prominently, barriers to accessing data, which often lies behind institutional barriers, preclude the very possibility of a validation study in the first place.

Therefore, having a standardized approach which makes it possible to select from a wide range of configurable combinations can make it easier to develop proof-of-concept models which are more transparent and amenable to scrutiny. By the same token, if patient data does not need to leave the respective clinic, bringing algorithms to the data, not data to the algorithms, these barriers can be mitigated. As such, the first research question of this thesis is thus:

Research Question One

To what extent can the development and validation of clinical models be standardized with software support across institutional barriers?

1.2.2. Need for Model Interpretation to Uncover Biases

CPMs are routinely developed by means of commercial statistical packages such as SAS, SPSS and Stata. However, an increasing number of clinical researchers and data mining practitioners are turning to ML methods when developing prediction models. A case in point is the field of Nephrology, in which a host of research papers exploring the potentials of ML for predictive applications has been published [12]. While there is an on-going debate on whether ML models are preferable to 'non-ML' models, particularly where interpretability is concerned [13], advances in ML research warrant the exploration of such models, particularly deep learning approaches [14]. As such, methods which help to shed light onto black-box ML models are required if these models are to be accepted in clinical research.

Therefore, models should allow scrutiny by medical experts, because good model performance does not exclude the existence of data artifacts, i.e., characteristics which do not belong to the underlying population, but rather emerge for different, unrelated reasons. Therefore, to overcome a lack of trust, a prediction model should not only be internally but also externally validated to ensure the results achieved upon derivation hold true for a diverse patient population. In particular, ML models are specially prone to 'learning' dataset-specific characteristics which might fail to generalize on a wide range of cohorts, further compounding the issue of lack of trust [15].

However, model interpretation is often considered an after-thought, and not taken into account and/or reported in clinical modeling studies. This is specially of concern when it comes to ML-based models. Case in point: if not audited properly, a well-performing model trained to diagnose cancer could have learned the manufacturer's label on the image, rather than the actual cancer features' themselves. As such, the second research question of this work is thus posed:

Research Question Two

To what extent can the use of interpretability methods on black-box prediction models help illuminate model biases?

1.2.3. Need for Support in Choosing Model Features

The data footprint left by care processes in the EHR is a blessing and a curse at once: on the one hand, as more data is generated, the opportunities for finding unexpectedly useful patterns increase; on the other, the features available for modeling increase substantially, which can lead to models which overfit the data, a phenomenon termed 'the curse of dimensionality' [16]. Here, feature selection strategies can support modelers to identify the most relevant predictors for any given model, reducing bias. This is specially important for external validation studies, since a good model should perform well not only on its original population, but also on unrelated cohorts. Since it is common for the validation cohort not to have all the necessary features available, the researcher needs to choose which features to keep and which to discard, i.e., update the original model, so as to achieve good performance with the fewest number of features possible on *both* cohorts. However, it is unclear to what extent conventional feature selection methods can be useful in such validation scenarios for clinical predictive modeling, since extant work so far has focused primarily on other types of high-dimensional data, such as gene expression [17].

Furthermore, instead of conventional feature selection methods, such as statistical tests, a promising alternative is the use of interpretability methods to achieve the same goal: simpler models that perform well in the original and in the validation cohort. The rationale for this is as follows: the interpretability methods provide insight into how the model operates. If this 'knowledge' acquired by the model

reflects underlying properties of the task, one could reasonably expect that these properties would be reflected on the validation cohort as well. In this thesis, I put this hypothesis to test. Therefore, my third research question is thus derived:

Research Question Three

To what extent can interpretability methods be useful in updating models in validation studies?

1.3. Research Objectives

The overall objective of this thesis is to demonstrate how the development, validation and interpretation of CPM can be standardized. Then demonstrate it in a concrete use case and evaluate to what extent interpretability approaches can be useful in illuminating biases and helping to update models in validation studies. To achieve this overall goal, the following objectives have been identified:

- **Objective 1:** *Identify* functional and non-functional requirements for a research prototype to support standardized clinical predictive modeling.
- **Objective 2:** *Develop* a research software prototype which implements the requirements identified.
- **Objective 3:** *Demonstrate* the applicability and feasibility of components of the research software prototype on a selected clinical modeling task.
- **Objective 4:** *Assess* to what extent the evaluation, validation and interpretation of clinical models can illuminate model biases.
- **Objective 5:** *Evaluate* to what extent the use of interpretability methods can be useful in updating predictive models in an external validation cohort.

1.4. Contributions

In answering the research question and pursing the objectives laid-out previously, I developed three contributions, spanning the dimensions of software platform, case study and method evaluation, listed below.

Software Platform. The first contribution is concerned with a working research software platform that supports the predictive modeling process with respect to the dimensions of model development, validation and interpretation. To that extent, we needed to identify the needs of practitioners in the clinical modeling process in terms of actors, tasks and challenges. Once mapping has been carried out, functional and non-functional requirements were identified, which, in turn, informed the actual artifact building. The prototype consists of a fully function web application along with a Python library that automates and parameterizes routine tasks in clinical predictive modeling.

Case Study. We demonstrate the utilization of components of the software platform in a concrete case study to predict AKI in the context of heart surgery. AKI affects a significant number of cardiac surgery patients, being associated with higher mortality and complication rates [18]. The model was developed and validated using data from three different institutions. The model developed is throughly discussed, including its development, validation and interpretation. In particular, we demonstrate how the interpretability methods can help practitioners to uncover potential model biases.

Method Evaluation. The use of interpretability methods on ML-models helps to shed light on their behavior, e.g., helping to highlight important features. Identifying such important features can also help to reduce complexity of clinical models, for instance using the method called recursive feature elimination. We extended this method by combining the inputs of different interpretability methods into a composite score, and applying the models in a validation cohort. This method was compared to other standard feature selection methods with respect to its performance in the concrete case study.

1.5. Scope

The predictive modeling discussed in this work is strictly focused on the class of tasks in machine learning regarded as classification problems in the field of *supervised learning*. More specifically, we tackle binary classification task, where the output of the predictive models is a probability for presence or not of a disease or risk probability. As such, problems involving predicting a continuous variable, i.e., regression tasks, while also important in the clinical context, are not object of this work.

This was a conscious choice because 1) multi-class tasks can be easily transformed into equivalent binary classification tasks using, e.g., one-hot encoding and 2) binary classification tasks cover a wide range of use cases in medicine. Furthermore, in this thesis, we do not tackle deep learning algorithms, since the core object of analysis are not the algorithms themselves, but rather the process of clinical predictive modeling itself. Nevertheless, the topics discussed in this thesis, particularly those referring to interpretation could be also extended to deep learning algorithms.

1.6. Outline

This thesis is structured as follows. Chapter 2 provides the necessary background, considering the modeling process, modeling algorithms, model performance metrics, and interpretability. This sets the common ground for the three contributions outlined above, namely, research software platform, case study and method evaluation, in Chapter 3, Chapter 4 and Chapter 5 respectively. Each of these chapters has its own listing of methods, results and discussion, therefore I deliberately did not include an extra 'Methods' chapter, since it would lead to unnecessary repetition. Finally, Chapter 6 provides a summary of the contributions achieved with this work, while revising research questions and objectives, as well as delineates future avenues of research.

Table 1.1.: Overview of top-level thesis structure, including research questions, objectives, chapters and contributions to help navigate this document. Abbreviation: RQ=Research Question, RO=Research Objective.

Question	Objective	Chapter	Contribution	Pages
RQ1	RO1 RO2	Chapter 3	**Software Platform**	20 - 56
RQ2	RO3 RO4	Chapter 4	**Case Study**	58 - 84
RQ3	RO5	Chapter 4	**Method Evaluation**	86 - 102

2. Background

"Let us not forget that the causes of human actions are usually infinitely more complex and more various than we are in the habit of explaining them afterwards, and are seldom clearly outlined."

—Fyodor Dostoyevsky, *The Idiot (1879–80)*

How are clinical predictive models born? In this chapter, the required background concerned with the field of clinical predictive modeling with respect to development, validation and interpretation will be covered in greater depth. This background knowledge helps to frame this work's contributions.

2.1. Defining Clinical Predictive Model

Model development entails the activities concerned with answering a given medical question by means of mathematical tools to diagnose a condition or predict an outcome using patient-related data. As such, models are said to be either diagnostic or prognostic in nature [4]. In this work, we are primarily concerned with prognostic models, more specifically the ones in which the outcome of interest is binary, which is often employed in clinical research [19].

Formally, given a number of patient characteristics, such as demographics, vitals and medications, represented by a feature vector $X \in (x_1, x_2, ..., x_n)$, a model is defined by a function f that maps those features to the likelihood or probability y' of a patient outcome or diagnosis y, often a binary marker, as defined in Equation 2.1:

$$f(X) = y' \mid y' \in [0...1] \tag{2.1}$$

2.2. Clinical Predictive Modeling Process

Different publications have set out to outline a general framework for this purpose [4, 6, 7]. Building on the intersections of previous work and the insights gleaned from expert interviews, I derived a summarized view of the process, depicted in Figure 2.1 using the Business Process Modeling Notation (BPMN) 2.0 notation [20].

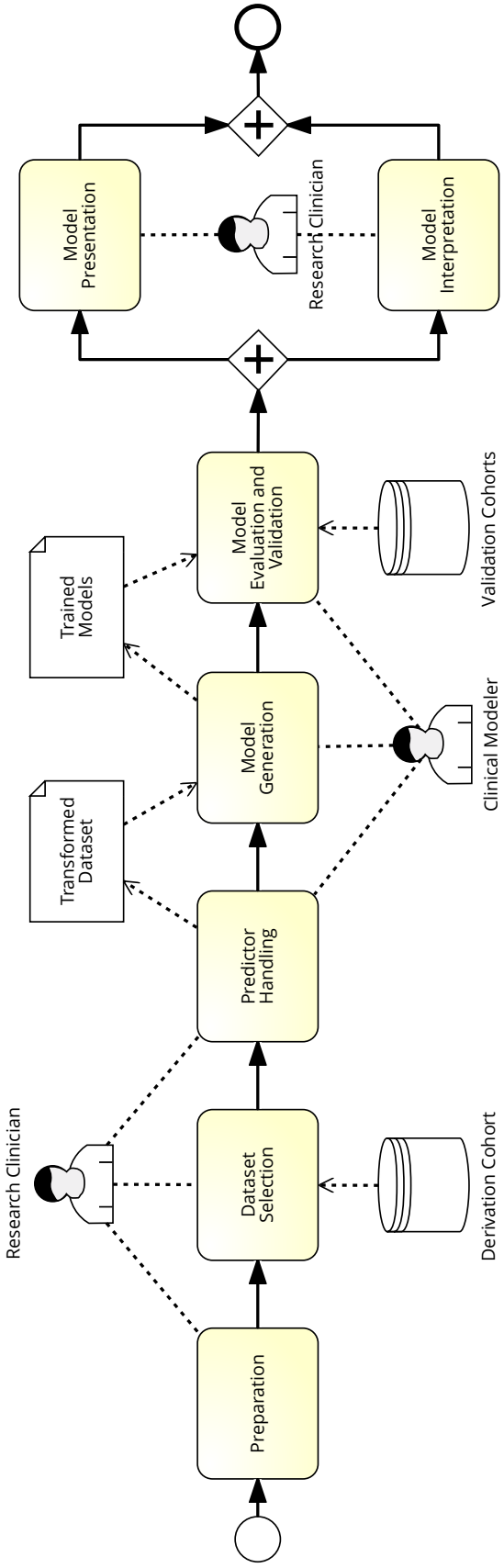


Figure 2.1: Steps involved in clinical modeling illustrated by Business Process Modeling Notation (BPMN) 2.0 diagram.

The following sections provide a summarized overview of the different process steps identified.

2.2.1. Preparation

The goal of clinical modeling is to develop an accurate and clinically useful instrument. To this end, critical questions have to be addressed before modeling itself can start. First, the endpoints to be analyzed, such as an event (mortality, readmission, etc.) or disease (stroke, liver injury, etc.) must be defined. Second, researchers must pinpoint the target population, that is, for what type patients the model will be applied (elderly patients, diabetic patients, chronic cases and the like). Third, target users of the model must be specified, defining, e.g., if it is target at general practitioners, at nursing staff, specialist physicians, etc. The answer to those questions ultimately guide the data selection step and also influences how variables shall be transformed prior to modeling [6]. For example, models targeted at specialist users may make use of advanced biomarkers, while those aimed at low-threshold clinical settings could rely on more readily available parameters.

2.2.2. Dataset Selection

Once the initial requirements have been established in the preparation phase, one needs to analyze available data sources with respect to their suitability to answer the questions defined. While primary data is usually preferred, i.e., data collected specifically for the given purpose, often clinical researchers have to rely on secondary sources, i.e., administrative data, for modeling. Further, depending on the purpose of the model, that is, diagnosis or detection vs. prognosis or future incidence, different types of cohorts are applicable. In general, diagnostic models can be developed with cross-sectional data, while prognostic models usually require longitudinal cohort data [6]. In addition, apart from ensuring that the modeling cohort is representative of the target population, a distinction must be made between development and validation datasets. Ideally, to properly assess generalizability, validation cohorts should come from external study populations. When these are not available, a subset of the data at hand can be held out for internal validation, e.g., via bootstrapping (sampling with replacement).

2.2.3. Predictor Handling

Selected datasets often possess a high number of candidate predictors. To ensure the final model is user-friendly, these must be carefully assessed with respect to statistical significance levels, e.g., p-values, as well as cross-correlation between predictors, in which case a predictor does not add new information. Different techniques for feature selection exist, such as t-tests, mutual information gain or principal component analysis and should be applied at the researcher's discretion. Furthermore, as a rule, the predictors available in the selected datasets must be pre-processed prior to modeling. For example, continuous predictors can be turned into categorical items or binarized. Similarly, different categorical items might be combined into a binary predictor. Further, depending on the modeling algorithm, numerical predictors should be normalized. Finally, clinical datasets are usually plagued by missing data, which might comprise the model. Therefore, care must be taken to either 1) remove from modeling predictors which are too often missing or 2) use appropriate imputation techniques to mitigate this effect [6].

2.2.4. Model Generation

In this step, the clinical modeler tries out different models, aiming to find the that best fits the available data. Published literature on clinical predictive modeling often makes use of logistic regression and Cox proportional harzards model [10]. At the same time, machine learning approaches are

increasingly finding their way into this research area [21]. Regardless of the underlying modeling algorithm, this task can be substantially time-consuming, since different predictor combinations should be examined, i.e., using the full set of predictors or subsets by means of backward elimination or forward selection [22]. In the case of machine learning algorithms, the problem is further compounded by the need to optimize the algorithm's hyperparameters, for example using techniques such as random or grid search, a computationally expensive task [23]. Finally, even though this optimization step might improve the model's performance, it potentially increases its propensity to overfitting the data. Overfit can be assessed by means of Akaike and/or Bayesian information criteria [24].

The ability to compare multiple candidate models side by side is vital for predictive modeling. In a similar vein, models are usually iteratively developed, in multiple cycles with different versions. It becomes necessary, therefore, to track changes to model parameters and underlying data/features to allow for informed comparisons. In effect, such changes can only be tracked if the models can be formally defined by means of a standard modeling language that supports a reproducible, machine-readable specification.

2.2.5. Model Evaluation and Validation

The models thus trained need to be evaluated with respect to how well they accomplish the given task. Likewise, results achieved must be validated, i.e., confirmed or dispelled, with a dataset not previously seen by the model. When a subset of the derivation dataset itself is used for this purpose, validation is said to be 'internal'. Conversely, external validation is carried out on a dataset from a different cohort, study or institution, which is the prefer ed approach to validate predictive models [10]. A number of different standard metrics exist to evaluate and validate CPMs, most prominently discrimination and calibration metrics. The first refers to a model's ability to discriminate between an event taking place or not (e.g. death) or between a diseased or healthy patient. Metrics such as sensitivity, specificity, and Area Under the Receiver Operating Characteristic (AUROC) are frequently employed in this context. The latter, calibration, refers to how well predicted risk matches observed risk for different equally-sized risk deciles. This is an important dimension to assess, because even if a model presents good discrimination metrics, it might overestimate risk for patients that actually present a low-risk profile. Calibration is typically visualized by means of calibration or reliability curves [6].

To evaluate model performance, tools should enable practitioners to easily visualize a set of standard metrics, such as AUROC, precision, recall for discrimination, as well as reliability curves and/or Hosmer-Lemeshow test for calibration purposes. While internal validity can be established by using the derivation cohort itself, external validity must be ensured, e.g., by providing an easy mechanism to exchange predictive models across institutional barriers and the ability to update existing models with the information of a new cohort.

2.2.6. Model Interpretation

While ML models potentially provide better discriminative performance when compared to linear approaches such as logistic regression, they typically do not allow scrutiny by non-ML experts and are therefore deemed to be black-box or present high opacity, a property arising from the internal mechanics of certain algorithms [25]. It is essential for doctors to be able to scrutinize an algorithm's decision so as to allow the most appropriate course of action, particularly when expert intuition disagrees with the model's output [26]. As ML-experts develop ever more complex and accurate models, the distance between clinic and the scientific computing servers grows ever wider. In effect, as stated by Ridgeway "an interpretable model that is actually used is better than one that is more accurate but sits on a shelf" [27]. Different approaches exist which are targeted at conferring interpretability to existing black-box models [13]. Model interpretability can be achieved at the global and local level.

The first refers to the relationship between target and predictors for the model as a whole, while the second seeks to examine single prediction instances [15].

IT-support tools should provide intelligible explanations for its predictions. This is of particular importance when it comes to ML-based algorithms, usually not easily interpretable. These explanations can take the form of feature-based importances, which provide some degree of insight into a model's behavior by assigning different magnitudes (coefficients) to the model's predictors.

2.2.7. Model Presentation

Finally, for actual use in clinical routine, models need to be properly presented, e.g., by means of an easy to calculate score that can be readily operated by clinical users in the point of care. Regression formulas, paper-based scores and nomograms are typically employed to this end. Increasingly often, these clinical instruments are made available to practitioners in the form of mobile apps or are calculated directly within clinical software. This is particularly relevant when it comes to ML-based CPMs, given the usual larger numbers of predictors used and the more sophisticated mathematical operations involved, which could be cumbersome or impractical to calculate manually [4].

Once models have been properly validated and interpreted, they can be considered for deployment. IT-tools should enable practitioners to derive easy-to-use scores based on the models developed. Ideally, the tool itself should provide the means to automatically calculate the prediction scores from the model given the appropriate input. Furthermore, the tool should make it possible to seamlessly integrate the model into existing clinical systems.

2.3. Modeling Algorithms

In the following sections, details on the modeling algorithms utilized throughout this thesis will be provided. Indeed, researchers make use a number of modeling algorithms, such as logistic regression or ML approaches, outlined below.

2.3.1. Logistic Regression

Logistic Regression (LR) is widely used for clinical prediction model development. It provides fast training time and easy-to-interpret coefficients for each model feature. For the sake of illustration, in a univariate logistic regression model, the probably that an input vector X can be assigned to the default class (or $y = 1$, i.e., AKI onset) is given by Equation 2.2 also known as logit function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{2.2}$$

The parameters β_0 and β_1 need to be estimated via an optimization procedure. This algorithm seeks to obtain coefficients β_i for each input feature in order to produce a binary output while minimizing the error between predicted and actual class membership using maximum-likelihood estimation [28].

Owing to its simplicity, however, LR tends to perform worse when compared to more sophisticated algorithms such as Multilayer Perceptron (MLP) or Random Forest (RF). Critically, LR is built upon the assumption of linearly correlated inputs and outputs. This is potentially an issue, since in a medical context one cannot necessarily assume linear relationships.

Hyperparameters. One of the key hyperparameters to be tuned for LR refers to the regularization strength. Model performance upon validation can be improved by penalizing large coefficients, potentially reducing overfitting. As such, model sparsity is improved by a strong regularization, typically defined as λ. Another key parameter to tune is the the type of penalty for the regularization,

namely L1 (lasso) or L2 (ridge). Since utilizing L1 penalty shrinks the coefficients of less importance to zero, some features might be removed altogether, a desirable property when dealing with wide datasets.

2.3.2. Decision Trees

Intuitively, a decision tree classifier seeks to define the best possible way to split a given training set in two sets, such that these sets are as homogeneous as possible. In other words, such that they belong to the same underlying 'class', e.g., such that all (or most) patients in any given set are diseased or non-diseased, or positive or negative. The question then becomes how to define this 'best split' [29]. The choice of how to split a sample depends on how well a given feature separates – or discriminates – between those two classes. Ideally, each of the sets should be 'pure', i.e., perfectly homogeneous: all positive examples from one set should be contained in the other with no negative examples.

There are different choices on how to calculate node impurity, such as error rate (proportion of 'wrong' examples), entropy, derived from information theory [30], Gini index, or Gini impurity[31]. Considering a classification problem with a number of different classes C and i as a given class, Equation 2.3 provides the Gini impurity G of a set, where $p(i)$ is the class distribution probability of class i. In practice, finding homogenous splits does not come easy: the trees are grown by recursively testing different split criteria and assessing the impurity of the resulting splits or leaf nodes until a desired stopping criteria is achieved or no more splitting is possible.

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \tag{2.3}$$

Hyperparameters. When modeling with decision trees, practitioners can, e.g., choose different impurity measures. Since trees can grow arbitrarily complex and thereby overfit the data, regularization parameters can be applied, as such defining a maximum tree depth [32]. Also, a weight can be defined for each of the classes, if one is considered to be more important, e.g., in the case where there is class imbalance.

2.3.3. Gradient-boosting Decision Trees

Gradient Boosting Decision Trees (GBDT) are an ensemble classification approach, i.e., one in which several simple classifiers are combined to tackle a complex task by majority voting. In this specific case, the ensemble is made up of several small-depth decision trees, which are then trained in iterative steps [32]. In binary classification using GBDT, the goal is to derive a function F to classify an input vector X in an additive fashion, such that following the numerical optimization proposed by Jerome Friedman [33]:

$$F^*(X) = \sum_{m=0}^{M} f_m(X) \tag{2.4}$$

In Equation 2.4, f_m are termed base learners, with $f_0(X)$ being an initial guess, chosen in such a way as to miminize the aggregated loss while keeping the preceding base learner fixed. For GBDT, these weak learners are tree stumps, i.e., tree with small depth, also termed 'tree stumps'. This process is repeated iteratively using gradient descent. In each of the iterations, the influence of the new weak learner can be modified by an weight factor, or gradient step size, or learning rate, before it is added to the previous iteration [34].

Hyperparameters. Given that GBDT rely on decision tree stumps as weak learners, they support all the parameters that also apply to decision trees. The learning rate can also be fine-tuned for GBDT. Smaller values for the learning rate, i.e., smaller than one, generally lead to a larger number of composite trees, since the gradient descent procedure will need more iterations to converge. Additionally, modelers can also define the highest number of invidual learners beforehand. This can be advisable for very large datasets.

2.3.4. Random Forests

The RF algorithm builds an ensemble of multiple trees in order to get a more accurate and stable prediction in comparison to an approach that relies on single decision tree. The ensemble's constituent trees utilize a random subset of the features available to split the nodes to be classified [32]. As a result of 'pooling' or majority voting of individual predictions, characteristically, RF are less prone to overfitting than regular decision trees. RF relies on bagging or bootstrap aggregation, i.e., sampling with replacement, to select samples of the training data, in an effort to reduce variance in the prediction function [35]. Hastie et al. formalize the concept in Algorithm 1 [35].

Given a set of constituent trees b where $b \in \{1, \ldots, B\}$, we denote the overall class prediction of the random forest rf over all B trees for input x by $\widehat{C}^B_{rf}(x)$. Accordingly, if we denote the class prediction of the bth constituent tree by $\widehat{C}_b(x)$, the classification output of the RF model is given by Equation 2.5:

$$\widehat{C}^B_{rf}(x) = majority\ vote\{\widehat{C}_b(x)\}^B_1 \tag{2.5}$$

Algorithm 1: Training a Random Forest

Input: Training Data
Result: Ensemble of Trees
for $b = 1$ *to* B **do**

(a) Obtain bootstrap sample of size N from training data;

(b) Grow tree T_b to the bootstrapped data, applying these steps recursively, until minimum node size n_{min} is reached:

i. Select m variables at random from the available p variables;

ii. Pick the best variable/split point among m;

iii. Split the node into two daughter nodes;

end
Return Ensemble of Trees $\{T_b\}^B_1$;

Hyperparameters. The RF algorithm tends to perform well even without extensive tuning, what may explain its wide popularity [35]. In addition to the usual hyperparameters for decision trees, such as tree depth, the library employed exposes a number of hyperparameters that can be tuned specifically for RF. They include, e.g., the number of constituent trees (or estimators), i.e., B from Algorithm 1, number of variables m to split a node and the minimum number of leaves required to split an internal node.

Table 2.1.: Confusion matrix for the predictions of a binary classifier. A similar structure applies for non-binary classifiers, with n sup 2 number of total cells, where n is the number of different labels.

		Actual Outcome	
		Diseased	*Healthy*
Predicted	*Diseased*	*TP*	*FP*
	Healthy	*FN*	*TN*

2.4. Model Performance Metrics

Once models have been developed, their performance needs to be assessed by means of different metrics. Roughly, one can distinguish between discrimination, calibration and clinical usefulness metrics [36].

2.4.1. Discrimination

Discrimination helps researchers to assess the extent to which the model can discriminate between patients that have will have the outcome from those that will not, i.e., True Positives (TPs) and True Negatives (TNs), respectively. Any classifier is bound to make mistakes. As such, a model might indicate a patient as sick who actually is not sick, i.e., a False Positive (FP) and conversely miss out on a patient who is sick by labeling him not sick, i.e., a False Negative (FN). The relationship between TP, TN, FP, FN is often illustrated by means of a so-called confusion matrix such as depicted in Table 2.1.

For binary classifiers, the combination of those indicators (TP, TN, FP, FN) gives rise to metrics such as AUROC, precision, recall and F-1 scores which are routinely utilized for evaluating CPMs [36].

Sensitivity and Specificity. Sensitivity, also called recall, and True Positive Rate (TPR) can be understood as a measure of completeness, i.e., to what extent are diseased patients correctly characterized as such by the model? This metric of is particular interest in diagnostic models, since it quantifies the percentage of disease patients correctly identified. It can be thus calculated:

$$Sensitivity = \frac{TP}{TP + FN} = 1 - FNR \tag{2.6}$$

Conversely, specificity or True Negative Rate (TNR), refers to the proportion of negatives correctly labeled as such, i.e., how selective is the test? In other words, what percentage of healthy patients are correctly deemed non-diseased by the model? This metric can be calculated as follows in Equation 2.7:

$$Specificity = \frac{TN}{TN + FP} = 1 - FPR \tag{2.7}$$

In practice, there is often a trade-off between sensitivity and specificity that must be weighted out depending on the concrete scenario. A model with high sensitivity will be very good at *ruling in* disease. As such, it is often desirable when the harms costs associated with a false positive are high, e.g., screening programs in cancer. On the other hand, a highly specific model excels at *ruling out* disease. As such, such tests are often employed after a highly sensitive test for confirmation, e.g., an biopsy following a cancer screening [37]. It should be noted that both are independent of disease prevalence.

Diagnostic Odds-Ratio. Given the often necessary trade-off between sensitivity and specificity, a global measurement of model/test performance that allows easy comparison between different models is desirable. The Diagnostic Odds Ratio (DOR) makes use of the concept of odds ratio applied to diagnostic scenario [38]. It is defined by the odds-ratio of a test being positive in the diseased

population relative to the odds of its being positive in the non-diseased, i.e., healthy population. It is defined by Equation 2.8 or alternatively using sensitivity and specificity Equation 2.8, being independent of disease prevalence. The value for DOR ranges from 0 to infinity (the higher the better). Because of its definition, it must noted that DOR may rise steeply as sensitivity or specificity approaches 100%. A test with specificity of 99% and sensitivity > 90% has a DOR of more than 500 [39].

$$DOR = \frac{TP}{FN} \Big/ \frac{FP}{TN} \tag{2.8}$$

$$DOR = \frac{Sensitivity}{(1 - Sensitivity)} \Big/ \frac{1 - Specificity}{Specificity} \tag{2.9}$$

Area Under the Receiver Operating Characteristic Curve. Calculating a model's sensitivity and specificity requires defining a threshold. If the predicted probability is higher than a given threshold, the test will consider the patient disease and non-diseased otherwise. Since threshold-setting directly affects the counts of TP, FP, etc. a way of visualizing these key metrics for different thresholds is required. This can be achieved by means of a Receiver Operating Characteristic (ROC) Curve. It plots the percentage of false negatives or False Positive Rate (FPR) against the percentage of true positives or TPR for the whole range of cut-off thresholds, thus making it easy to compare models across different cut-offs. In probabilistic terms, it defines the likelihood that a patient who is diseased (a positive) will rank higher than one who is not diseased (a negative).

Furthermore, if one calculates the area below the curve, it offers a summary global metric of the model's discriminative capabilities, the so-called AUROC. As such, the higher the AUROC, the better the model. A perfect model would have a 1.0 AUROC while a model that has no predictive skill would score 0.5 (perfect diagonal).

Positive and Negative Predictive Value. Positive Predictive Value (PPV), precision, depends on prevalence and seeks to answer the question: what is the chance that a patient who tested positive in fact has the disease? Thus, it can be understood as a measure of the quality or exactness of the model [40]. It depends on disease prevalence, meaning, if prevalence increases, precision also increases [41].

2.4.2. Calibration

Calibration metrics help practitioners to assess the extent to which model risk predictions agree or disagree with observed – or real – risk rates in the real cohort. A widespread approach for visualizing calibration is by plotting predicted probabilities against observed outcome incidence or graphical method or calibration curve. Another approach to obtain a quantitative rather than qualitative assessment is by means of calculating alpha and beta calibration via curve fitting. Finally, calibration can be improved with techniques such as Platt's scaling or sigmoid calibration.

Calibration Curve. Since real risk rates are not known in advance, i.e., one tries to learn them from the data, practitioners proceed as follows. First, outcome incidence is calculated across cohort deciles, or actual outcome probability. Second, for each of those deciles, the predicted probability, i.e., the model's output is computed. By plotting actual probabilities against the mean of predicted probabilities within deciles, one can construct a so-called *calibration curve*, which makes it possible to visualize if the model is well-calibrated [42].

Alpha and Beta Calibration. Besides this graphical representation, by fitting a line to the thus obtained calibration curve, this line's intercept and slope offer a quantified measure of the model's

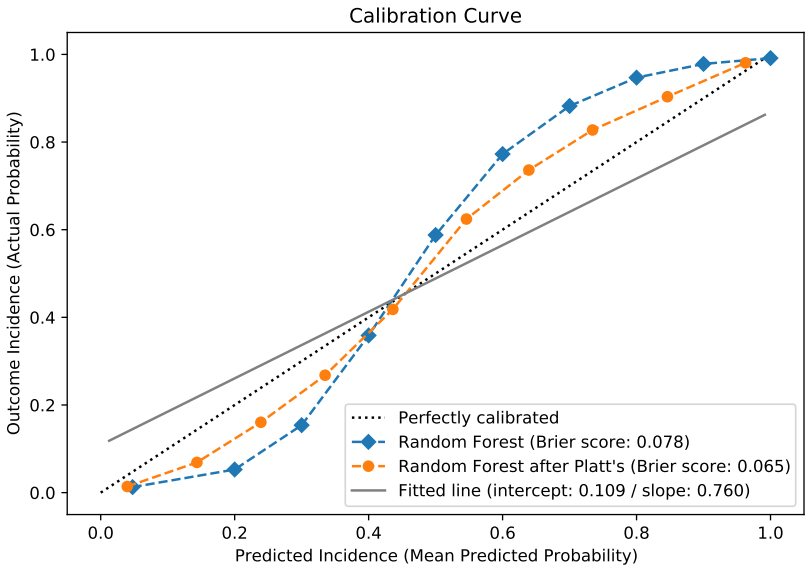


Figure 2.2.: Classification example showing different classifiers along with fitted calibration line. Note that after applying Platt's scaling, the Random Forest classifier achieves a lower Brier score.

calibration, also known as alpha and beta calibration, respectively. Alpha calibration or calibration-in-the-large reflects the extent to which predictions are systematically too or too high. Beta calibration helps to measure the spread between predicted and actual outcome probabilities across the different risk deciles. A beta coefficient lower than 1 suggests a classifier that labels low risk patients too low and high risk patients too high [36]. Accordingly, a perfectly calibrated classifier would always be consistent with the observed outcome across all risk deciles, that is, alpha=0 and beta=1. Another approach to quantify calibration is to use the Brier score, or the mean squared error of the model's prediction [43].

Platt's Scaling. Furthermore, if the model is not well-calibrated, i.e., over or under-predicting risk, it is possible to calibrate it by adjusting the predicted probabilities. One widely spread approach to achieve this is Platt's scaling, particularly useful when the distortion of predicted probabilities are sigmoid-shaped [44]. Essentially, this method relies on fitting a logistic regression on the predicted probabilities, i.e., outputs of the classifier. As such, it makes strong assumptions on the distribution of predicted probabilities (parametric). Another non-parametric option is isotonic regression, which however requires a large number of samples to avoid overfitting ($>$ 1000). When fewer samples are available, Platt's method is often preferred. Noteworthy is that the calibration procedure will not lead to changes in summary metrics of performance, such as AUROC, but may well affect others, such as Brier score and log loss [36].

Figure 2.2 adapted from the documentation of scikit-learn [45] illustrates the concepts above with a RF classifier trained on an example classification task. Note that applying Platt's scalling helps to improve calibration, leading to a lower Brier score. Nevertheless, calibration slope is smaller than one, revealing that the classifier tends to predicts too low for lower risk deciles and conversely for higher ones, possibly implying model overfit.

2.4.3. Clinical Usefulness

While it can be argued that good discrimination and calibration metrics lead to better models, in practice they offer little guidance as to whether the models are actually useful to clinical decision making itself, when benefits must be assessed against risks, i.e., when consequences matter [46]. To illustrate, comparing two models with the same AUROC, the one with higher sensitivity might be preferred if the clinical harm caused by missing out on a positive case is higher than that of misdiagnosing a healthy case. Therefore, a method of weighting out the possible harms of false positives or false negatives against the benefit of having more true positives and true negatives – or net benefit – is desirable.

Net Benefit. Consider the concrete example of risk of AKI after a surgical procedure such as heart surgery. A recommended strategy to prevent AKI in these settings is the optimization of blood pressure, using fluid management and vasopressor agents [47]. However, such patients' circulatory system is already in a compromised condition and administering more vasopressors could potentially an increased mortality risk. Therefore, if the probability of AKI for the patient is very low, e.g., 5% the treating physician could refrain from blood pressure optimization to prevent AKI. On the other hand, if AKI probability is higher, the physician might decide in favor of this course of treatment in spite of risk associated with treatment.

The different probabilities provided by a model yield different counts of TP and FP depending on the chosen cut-off point. The threshold probability from which treatment would be considered, denoted by p_t, varies depending on the clinical use and depending on the treating physician and patient preferences. A measure of Net Benefit (NB) has been introduced by Vickers et al. on the basis of the theoretical relationship between the threshold probability of outcome and the relative value of false negatives and false positives [48]. NB can thus be calculated as follows:

$$NB = \frac{TP}{n} - \frac{FP}{n}\left(\frac{p_t}{1-p_t}\right) \tag{2.10}$$

Decision Curve. NB can be calculated for any given threshold p_t. Plotting NB across different threshold probabilities gives rise to an instrument called decision curve. It makes it possible for practitioners to readily visualize the range of thresholds for which there is a net benefit in acting upon the model's recommendation. Still, the model should be compared to either competing models or alternative courses of action. As such, for the decision curve, one should also consider the case when no patients are treated (no patients will develop AKI, treat none) and the case when treatment is always dispensed (all patients will develop AKI, treat all) [46]. Decision curve analysis helps to assess these trade-offs graphically using the relative weight of harms vs. benefits to define the threshold in different courses of action [48]. Conveniently, a summary measure of net benefit – Area under the Net Benefit Curve (ANBC) – can be calculated via trapezoidal rule [49].

2.5. Interpretability Methods

The enhanced performance with ML algorithms is often achieved at the expense of model interpretability. The ability to explain and interpret decision is a key requirement in medical applications. For ML applications, Lipton places particular focus on identifying decision boundaries and ascertaining the influence of specific feature for improved interpretability [50]. Approaches have been developed to achieve interpretability of black-box models, such as the classification vectors approach by Baehrens et al. and the Locally-Interpretable Model-agnostic Explanations (LIME) by Ribeiro, Singh, and Guestrin [51, 52]. In particular, Katuwal and Chen applied the LIME technique for achieving interpretability of random forests for predicting ICU mortality, with accuracy of 80 % [53]. Still in the

medical domain, Hayn et al. quantified the influence of individual features on particular decisions made by a random forest in clinical modeling applications [54].

2.5.1. Defining Interpretability

With respect to ML models and results, Doshi-Velez defines interpretability as "the ability to explain or to present in understandable terms to a human" [55]. In contrast, Lipton sees interpretability as a "non-monolithic concept" which encompasses a host of "distinct ideas" [50]. Expanding on these ideas, a fledging community of researchers, deemed Fairness, Accountability, Transparency (FAT) academics, emphasizes, amongst others, explainability as one of the core principles for accountable algorithms [56]. This principle establishes that algorithmic decisions should be intelligible to end-users in "non-technical terms". For the purposes of this thesis, we define interpretability as a *property of machine learning algorithms and their outputs which allows scrutiny by medical experts*. Under scrutiny, we mean the ability of doctors to 1) easily ascertain the 'reasoning' behind an algorithm's decision, 2) identify the most relevant features for the model's output and 3) illuminate possible biases within the model.

For the purposes of this thesis, we define interpretability methods as "tools which quantify or visualize feature effects or feature importance", describing how features contribute to the predictions of the model globally or locally [57]. This task is typically achieved by means of surrogate models or method-based approaches.

2.5.2. Global Surrogate

Global surrogates seek to distill the knowledge captured by a black-box ML model into a more interpretable model. In this method, also termed mimic learning, a simple, more interpretable or *student* model relies on the predicted probabilities of the original or *teacher* model. As such, instead of having the actual outputs as training target, the student model is trained on the predicted probabilities of the teacher model while retaining the same original input features [58]. Formally, given a prediction model defined by $f(x, y) = y'$, where x is the model input and y' its output for a true label y, we train a mimic model $g(x, y') = y'_*$ where $g \in G$, i.e., a class of interpretable models. As such, the mimic model is obtained by minimizing $\sum_{i=1}^{N} ||y'_i - y'_{i*}||^2$ for N training samples. It is worth noting that the student model is only as accurate as its teacher.

Training the student model on the predicted probabilities makes it possible to derive a more understandable, but comparatively well-performinig prediction model. In fact, under certain circumstances, the student model could generalize better than the more complex, teacher model [58]. Arguably, the interpretable model attenuates training data irregularities, which could have a detrimental effect on the teacher model's generalizability. The logic implemented for the mimic learning approach is shown in pseudo-code in Algorithm 2 [58].

Algorithm 2: Mimic Learning with Surrogate Model [58]

Input: ML Model, Training Dataset and Test Dataset
Result: Sorted mimic regression coefficients
Obtain Predicted Probabilities of ML Model on Training dataset;
Train Mimic Model on Predicted Probabilities and Training dataset;
Apply trained Mimic model on Test dataset;
Obtain Mimic Model regression coefficients on Test dataset;
Sort Regression Coefficients;
Return Regression Coefficients;

2.5.3. Local Surrogate

This approach differs from global surrogate to the extent that the focus of explanation is one single prediction instance, therefore operating locally as opposed to globally. The LIME method makes use of a more interpretable model, e.g., linear regression, to explain the behavior of a black-box algorithm when applied to a given sample, i.e., a specific patient instance [52]. Intuitively, LIME seeks to explain how the model's prediction changes when the input changes. Given an instance x, in our case a surgical patient, LIME generates a number of 'perturbed' samples weighted by their distance to x and fits an interpretable model to these new samples.

As per Equation 2.11 the explanations are given by minimizing a loss function $\mathcal{L}$ that measures how well the local surrogate g belonging to a class of interpretable models G approximates our model f in the vicinity of the instance of interest defined by π_x. The loss function is further penalized by model complexity $\Omega(g)$. As such, the explanations are given by the regression coefficients of the surrogate model, which are deemed to be locally but not globally faithful. To obtain a global understanding of the model's behavior, this method provides the so-called submodular pick, in which explanations are chosen that have the highest explanation coverage, thereby offering some insight into the model's global behavior. The method parameterizes how many explanations should be considered. We chose to generate explanations for 25% of the data used for training. We then computed the mean absolute value of each feature's contribution across all explanations. Given that LIME is prone to unstable explanations because of the randomness of the perturbed samples, we excluded features that appeared in less than 10% of the explanations.

$$\xi(x) = argmin_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.11}$$

2.5.4. Method-based Feature Importance

Many algorithms, such as logistic regression and similar linear models, provide a straight-forward way to obtain feature importances, e.g., via the calculated feature weights. For tree-based methods, the algorithm seeks to find features that split the data as homogeneously as possible based on a measure of impurity, such as Gini impurity or information gain. A feature can be therefore considered more relevant if it decreases impurity for a given split. As such, to estimate the relative importance of a feature, one can add up impurity decreases over all nodes that include that feature weighted by the proportion of samples which are split at the given node. Accordingly, nodes closer to the top of the tree will be considered more important.

For ensemble methods, decreases in impurity are averaged over all constituent trees, i.e., mean decrease in impurity [59]. In the case of random forests, one can calculate the importance of a variable X_m over all N_T trees as defined in Equation 2.12, where $p(t)\Delta i(s_t, t)$ is the weighted decrease in impurity over all nodes t which include X_m. In Equation 2.12, $v(s_t)$ is the variable used in split s_t and $p(t) = N_t/N$, i.e., the proportion of samples reaching t. A disadvantage of the mean decrease in impurity method is that is algorithm-dependent, i.e., can only be used with tree-based approaches.

$$Imp(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T: v(s_t) = X_m} p(t)\Delta i(s_t, t) \tag{2.12}$$

2.5.5. Shapley Values

This method utilizes insights from game theory to obtain feature contributions. In this context, the "players" can be construed as the different model features, while the model's prediction represents the "reward" desired [60]. Intuitively, one can build coalitions of players (features), which, jointly, contribute towards obtaining the reward (prediction). For any given coalition of features, there will

be a difference between the actual prediction for a given instance and the average prediction for all instances. As such, the Shapley value of the j-th feature ϕ_j is the average marginal contribution of a feature value considering all possible feature coalitions when that feature value is added to the coalition. As the number of features values increases, obtaining the exact contribution might become computationally costly, given the multitude of possible coalitions. To circumvent this limitation, Štrumbelj et al. developed a method to obtain an approximation $\hat{\phi}_j$ with Monte-Carlo sampling [61] using M total samples.

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}(x^m_{+j}) - \hat{f}(x^m_{-j}) \right) \tag{2.13}$$

In Equation 2.13, $\hat{f}(x^m_{+j})$ is the prediction for an instance x in which a random number of features have been replaced by values of a random data point z, while retaining the value of feature j, i.e., x^m_j. In a similar fashion, $\hat{f}(x^m_{-j})$ represents the case in which x^m_j is also obtained from the random data point z. Specifically, in this work, we utilized SHapley Additive exPlanations (SHAP), an even more computationally efficient alternative to estimating Shapley values, particularly in the case of tree-based methods [62].

3. Software Platform for Clinical Predictive Modeling

"Manuscripts don't burn."

—Mikhail Bulgakov, *The Master and Margarita (1967).*

To what extent can a complex process be standardized? Is it any useful to attempt to standardize it? In this chapter, I outline the challenges inherent to the clinical modeling domain and present a platform for rapid prototyping tailored to the specific needs of clinical modeling – Modeling of Outcome and Risk Prediction for Health Research (MORPHER). I argue that a move towards hybrid solutions, i.e., a mix of cloud and on-premise infrastructure, constitutes a viable way to reduce the time needed to develop, validate, and interpret clinical predictive models in a standardized, reproducible fashion.

3.1. Introduction

As we have seen in Chapter 2, the clinical predictive modeling process is substantially complex, requiring a number of iterative steps involving collaboration between different actors. In the following, I provide an overview of the the challenges faced by practitioners in these three areas, i.e., development, validation and interpretation and investigate to what extent adequate software support can be beneficial.

3.1.1. Motivation

Machine learning is rapidly becoming a mainstay in research and industry. Particularly for clinical predictive modeling, these approaches are being increasingly applied, as evidenced by the growth in the number of related publications. While different computer tools exist that support rapid prototyping, I posit that current approaches are lacking in the extent to which the needs of research clinicians are addressed. This leads to an increase in the time needed for development and validation of such models. Indeed, the clinical modeling process is riddled with challenges that ultimately affect the models' clinical credibility and therefore the extent to which predictive models are used in practice [5]. These challenges are related to flaws and pitfalls in model development as well as validation and the lack of impact studies [9]. Since the assessment of model impact usually requires setting up a prospective study, including a RCT, MORPHER is concerned primarily with the challenges pertaining to model development, validation and interpretation.

3.1.2. Challenges to Development

With respect to model development, three main challenges can be ascertained. They are concerned with tool support, development standardization and consistency in reporting. These challenges will be addressed in detail in the following sections.

Lack of Tailored Clinical Modeling Tools. Medical researchers who develop CPM models are often burdened by clinical responsibilities and cannot afford to spend valuable patient time coding data extraction and model training routines. Furthermore, since most clinicians are not trained statisticians or machine learning scientists, important methodological aspects are often overlooked. To circumvent these issue, professional statisticians or ML experts are often engaged in the modeling process, but lacking the medical background, repeated interactions are necessary until satisfactory results can be achieved. Current tools supporting clinical modeling mitigate those issues to a certain extent, e.g., by providing visual environments, but are found lacking in central aspects.

Lack of Standardization in Development. Model development is an inherently iterative process. Therefore, different combinations and tests of a multitude of adjustable parameters are necessary to achieve satisfactory model performance. Examples are algorithms and their hyperparameters, data configurations, validation strategies (e.g. split validation vs. bootstrapping), data imputation strategies, normalization, among others. As such, the search space and possible combinations can grow widely. This fact calls for a standardized approach in order to make it possible to precisely ascertain what steps have been followed when modeling. Standardizing the individual steps also make it possible to keep track of different experiments across iterations.

Lack of Consistency in Reporting. The number of CPMs published has been growing steadily. However, key details on how these models have been developed, what methods have been applied and how are often lacking. This is a particularly acute issue concerning performance reporting, meaning a model's statistical power. A large number of publications rely on discrimination as the sole metric of evaluation, which might convey a biased assessment of a model's capabilities. Other metrics exist, such as calibration and clinical usefulness, that contribute towards a more complete evaluation. In this context, guidelines such as Transparent Reporting of a Multivariable Prediction Model for Individual Prognosis or Diagnosis (TRIPOD) provide guidance on what to report in a predictive model publication [7].

3.1.3. Challenges to Model Validation

While a significant number of new predictive models are continuously being published, specially since the advent of ML approaches, only a handful of them are validated externally. The reasons for this phenomenon are manifold, partially stemming from a systematic novelty bias in academic publishing. For the purposes of this thesis, the challenges are mostly concerned with barriers to data access, reproducibility and iterative experiments.

Barriers to Data Access. For privacy protection reasons, data for validation in hospitals often lies beyond the reach of external researchers aiming to validate their models. In order to allow collaboration across institutional borders, apart from legally challenging and time-consuming efforts to allow data access to third parties, one can pursue a strategy of 'bringing algorithms' to data instead.

Barriers to Reproducibility. No standard for development exists which establishes a common, reproducible standard for clinical predictive modeling, thereby posing a hindrance for reproducibility of experiments. Next to a lack of a standard on a description level, a consistent environment for execution is also required, e.g., concerning libraries and software versions. A related issue is concerned with data standards: they should either have been used at derivation, so that validation is seamless, or a detailed specification of predictors must be available.

Barriers to Model Iteration. The iterative nature of model development also extends to validation. This process might require updates and changes on both ends, i.e., derivation and validation model, each giving rise to a new iteration. Given that models need to be adjusted to fit a given population over different experiments – or iterations – relying on having to manually running model updates every time might prove unfeasible.

3.1.4. Challenges to Model Interpretation

The gains in performance provided by ML models often come at the cost of interpretability. As a result, the adoption of ML-based models in the clinical setting has been met with skepticism in the medical community. To a significant extent, the reason for this skepticism can be traced back to the lack of transparency in the predictions generated by these complex algorithms. Therefore, it remains important to guarantee access to intelligible explanations, which allow model predictions to be scrutinized by medical experts.

Access to Intelligible Explanations. Explanations for complex machine learning models can be broken down into two perspectives. First, a practitioner might be interested to obtain a global understanding of a model's inner workings, e.g., in terms of what features most contribute towards its predictions. Second, a local perspective, i.e., scrutinizing predictions at the sample, patient-level is also desirable, since a model's averaged behavior might not be representative of how individual patients are handled by it.

Access to Explanation Comparisons. Multiple methods for deriving explanations exist. Given that different methods place focus on a particular aspect of how models work, their explanations might differ. Therefore, it becomes necessary not only to have access to different such methods, but also to obtain a comparison across different methods. This enables a more balanced view of what features are important, e.g., because multiple methods indicate them as relevant.

Access to Evaluating Explanations. While explanations provide an insight into model behavior, the possibility to act upon the insights, meaning the ability to change the original model in response to a medical expert's feedback remain not addressed. This usually entails adjusting the original scripts and submitting the explanations again for evaluation. Adequate support can speed up this process.

The remainder of this chapter proceeds as follows. First, in Section 3.2, I outline related work in this field, with an overview of software support for clinical predictive modeling. Then, in Section 3.3, details on the requirements engineering approach, and technologies utilized for the development are provided. Section 3.4 outlines functional and non-functional requirements derived from interviews with experts in the field. In the same section, I present the architecture and implementation of tool MORPHER which aims to address these requirements. Finally, in order to evaluate how MORPHER contributes to the existing body of knowledge, in Section 3.5 I present the results of a functional comparison between this work and other related approaches, assessment of expert interviews and performance in a simulated clinical modeling task for Chronic Kidney Disease (CKD).

3.2. Related Work

As the basis for related work comparison from a functional perspective, I conducted literature research on tools that can be used to support clinical predictive modeling. I used different permutations of the following search terms: 'machine learning', 'prediction model', 'predictive model', 'predictive modeling', 'predictive modelling' (British spelling), 'data mining' and 'prediction' in combination with the terms 'tool', 'toolkit', 'software', 'framework', 'workbench' and 'platform', e.g., all articles containing the full phrase (all of the words) 'predictive modeling platform', 'data mining framework' or 'prediction framework'.

In the assessment carried out, software tools were included which can be used to allow a clinical researcher to develop machine-learning based predictive models. Tools were excluded which might be used in the ML industry but are not yet part of peer-reviewed publications. Furthermore, tools targeted at natural language processing, text mining, bioinformatics, patents, as well tools focused on other domains, such as chemistry or articles in languages other than English were excluded.

The literature research carried out revealed a wide array of IT tools that are currently employed to support the clinical modeling process. They range from statistical packages to visual modeling environments, either for general purposes or with a biomedical focus. I assessed them with respect to their degree of compliance with the requirements gathered as laid out in Table 3.1. These tools can be broken down into the following categories:

- **Statistical Packages**. While extensively utilized in practice, statistical packages usually run on the developer's own machine, i.e, are not targeted at the specific needs of clinicians and are ill-suited to handle large amounts of data. Examples are SAS [63], SPSS [64], Stata [65] or others.
- **Mathematical Software**. Software such as Matlab [66] and Octave [67] provide statistical routines that could be used to develop prediction models. However, similarly to ML toolkits these solutions can only be utilized by expert users and reproducibility is not necessarily guaranteed.
- **ML Software Toolkits**. ML toolkits, though powerful, require expert knowledge for handling. Furthermore, the freedom developers have makes it harder to establish standards for sharing models. Toolkits such as scikit-learn [45], h2o [68], Keras [69], Pytorch [70], along with the myriad of ML R packages [71] fall under this category.
- **Visual Modeling Suites**. Visual ML suites like Weka [72], Orange [73] and RapidMiner [74], make model development and sharing substantially easier, but data integration remains an issue, performance reporting is not standardized, and medical knowledge is not integrated in a structured fashion. In addition, a combination of Jupyter Notebooks and a ML toolkit has been increasing applied in research [75].
- **Biomedical Software**. Biomedical software usually runs on a central local server and as such are detached from the outside world if sensitive data is being handled. Even though they automate model development to a considerable extent, sharing results across institutional borders remains cumbersome. Examples of this category are tools such as tranSMART [76], ATLAS [77] in combination with the PatientLevelPrediction (PLP) R package [78], MLBCD [79], and ExplICU [80].

Even though currently different tools exist which enable rapid prototyping of ML models, they address the challenges laid-out only to a limited extent. For example, current tools offer no readily-available support in the methodological aspects pertaining to modeling, for example in the form of a standard set of performance metrics and help in handling predictors. Besides, visual modeling tools make it possible to a certain extent to ascertain what steps have been followed in modeling. However, these tools are general-purpose and do not cater to the specific needs of clinical modelers, e.g., clinical usefulness metrics are not provided. Furthermore, extant IT-tool support provides no easy way to seamless share models across institutions, i.e., the possibility to take algorithms to researchers, not data. Likewise, only a subset of these tools support medical standards that could facilitate model validation.

3.3. Methods

In the following, I provide details on the approach pursued for requirements engineering. I also outline the technologies used in this work along with a typical user scenario. Finally, I present how the tool assessment was carried out.

3.3.1. Requirements Engineering

I utilized elements of the design thinking methodology to carry out a workshop with subject-matter experts for clinical predictive modeling and machine learning [81]. This expert panel was made up of

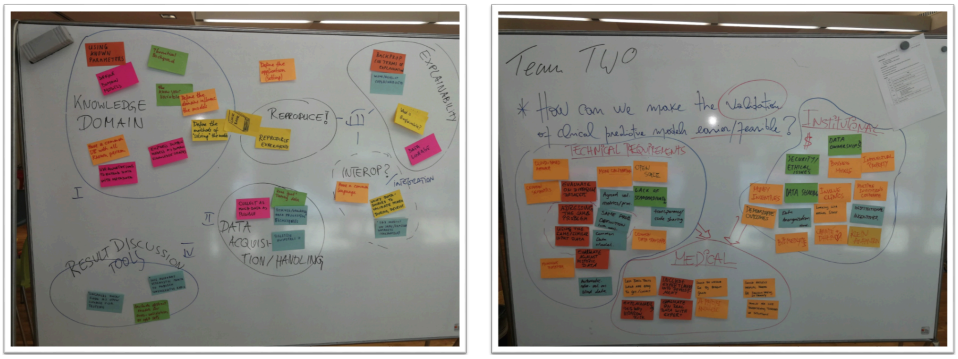


Figure 3.1.: Brainstorming session with workshop participants. They were divided in mixed groups, i.e., both with technical and clinical focus, and tasked with specific design challenges.

researchers linked to the BigMedilytics Horizon2020 project[1] and invited practitioners. In total, the panel included three medical doctors (general medicine, nephrology and cardiology) and six machine learning professionals, all of which routinely conduct predictive modeling research. The experts were divided in two multidisciplinary groups. The first group was tasked with answering the following question: "What should a computer tool provide to make clinical predictive modeling easier and help avoid common mistakes?". The second group addressed the question "How can a computer tool help make validation of clinical predictive models more feasible?". Each group then brainstormed in separate group session.

In a subsequent step, the results achieved in each group were jointly discussed and summarized. Then, I distilled the workshop results into *personas* and *user stories*. Personas are representative of categories of users of a given solution and stories synthesize user aims which can be translated into software requirements. This generated a catalog of user stories, which established the foundation for functional and non-functional requirements in accordance with ISO/IEC/IEEE 24765:2017 [82]. The requirements thus identified are listed in Table 3.1.

3.3.2. Personas

Two main stakeholders can be ascertained who are actively involved in the clinical modeling process, the **research clinician** and **clinical modeler**. The former usually participates actively in the early and late phases of the process, providing guidance on the research questions and cohort selection, as well as validating and interpreting results from a clinical standpoint. A high degree of clinical knowledge is required in this role, since it impacts data collection, defining predictors, and establishing key clinical requirements. The latter is more concerned with the analytical aspects of the process, being responsible for the statistical modeling itself. While deeper understanding of mathematical modeling is expected, some degree of clinical exposure helps to ensure seamless communication between the stakeholders.

3.3.3. User Story Mapping

In order to guide the implementation of the first MORPHER prototype, I utilized the User Story Mapping (USM) methodology [83]. Building upon the insights gathered during the expert workshop referred to in Section 3.3 above, I outlined important steps in the user's journey in the tool. Figure 3.3 displays the result of user story mapping for each aspect, modeling, validation and interpretation.

[1] https://www.bigmedilytics.eu/

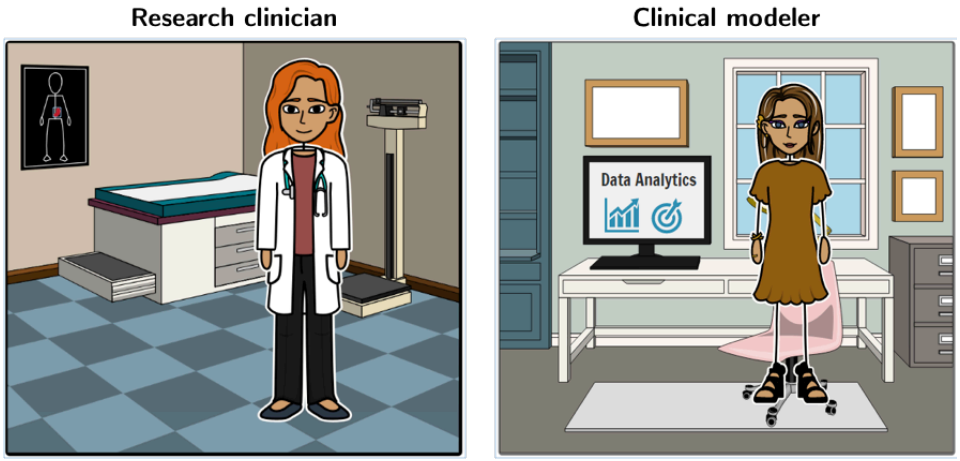


Figure 3.2.: Identified predictive modeling process personas. Research clinicians (medical specialists focused on research) and clinical modelers (data scientists specialized on health) collaborate to develop prediction models.

In summary, Researcher A selects a cohort and train models, upon which she evaluates the model's performance. A second actor, Researcher B, can then select that particular model and apply it to a new cohort, while visualizing the pertinent metrics (discrimination, calibration, clinical usefulness). Both researchers also have the possibility to inspect the generated model to interpret it.

Notably, the story mapping conducted has evolved over time, so Figure 3.3 represents a comprehensive yet not exhaustive subset of the users' journey in the tool. Also, not every single item of the story mapping could be successfully implemented. In particular, the items displayed under 'backlog' will be included in future iterations of the tool.

3.3.4. Technologies Utilized

The platform developed relies on substantial existing work. Though the technologies have been already elucidated throughout Section 3.4.2, I provide here a brief overview.

Web Applications. In the development of both MORPHER Cloud and Local, I applied the Model-View-Controller (MVC) design pattern, used to bring together the user interface with the underlying data models via so-called controllers [84]. Here I made use of Python web technologies, such as Flask, to develop the web applications, along side Javascript frameworks JQuery and Vue. For the local instances, I additionally used Docker containers for to make deployment easier.

Database Backend). MORPHER Cloud relies on Postgres as a database relational backend. Specifically, I made use of the document-based extensions of this database to allow extensibility of the different data resources. The execution environment makes use of an In-Memory database and keeps manages the task execution in a separate database.

User Authentication. In order to enable users to switch seamlessly between the local and cloud applications, I adopted the Single Sign On (SSO) approach, building upon existing work (HPI Identity Provider[2]). The user authentication flow follows the Security Assertion Markup Language (SAML) standard [85].

Execution Environment. I utilized the existing Analyze Genomes Worker Framework as the basis for the execution environment, where model training, validation and interpretation takes place [86]. Originally developed for genome analysis, I created MORPHER jobs which are fully pluggable within this framework, thereby making it compatible with the existing infrastructure. I utilized the existing

[2] https://accounts.analyzegenomes.com/idp/

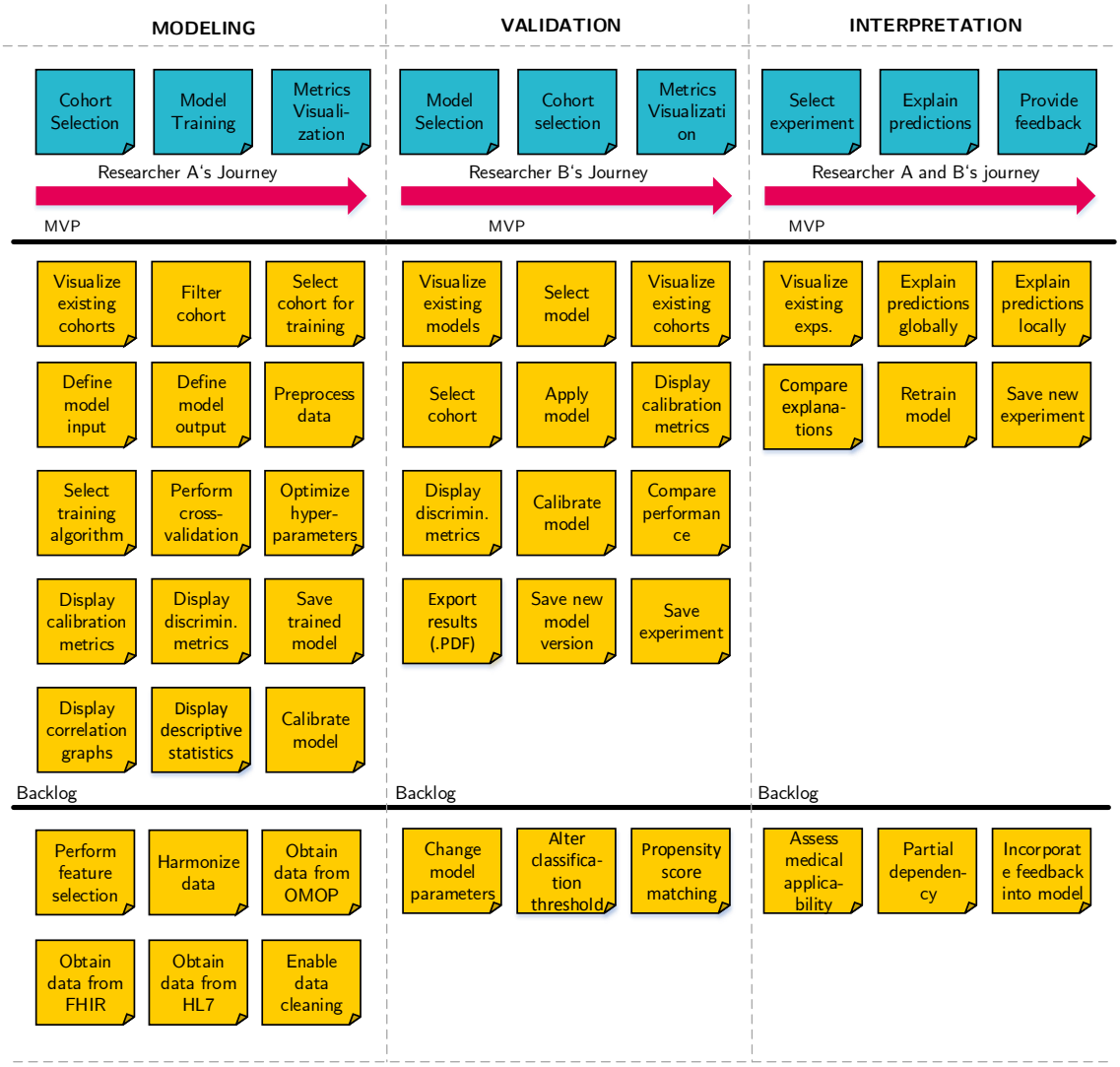


Figure 3.3.: User story mapping conducted. Concepts: Algorithm=a mathematical object for modeling, e.g., decision tree, etc.; Model=an algorithm trained on a given data set; Experiment=applying a given model on an existing cohort. Backlog represents features not implemented in the prototype. Abbreviations: MVP: Minimum Viable Product.

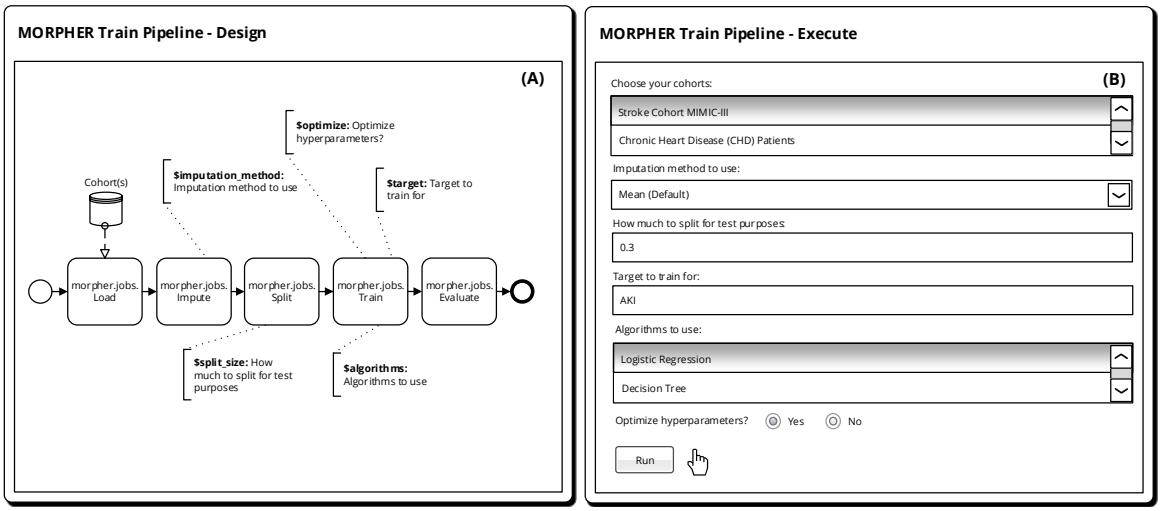


Figure 3.4.: Exemplary depiction of MORPHER train pipeline implemented via the Analyze Genomes Worker Framework [86]. Model developers can define pipelines graphically using Business Processing Modeling Notation 2.0 (A). Each step of the pipeline is an executable Python script that can be parameterized in a later step (B).

Worker Framework for definition and execution of specific ML tasks on a scalable distributed computing infrastructure. The Worker Framework automates scheduling and parallel asynchronous execution of such jobs on distributed computing nodes. All data pertaining to task scheduling and execution is stored in a persistent database, so that the statuses of long-running tasks, such as hyperparameter optimization, can be tracked and performance statistics retrieved.

Machine Learning. For the ML pipelines, I employed existing ML libraries, such as Pandas [87] for data manipulation, scikit-learn [45] for ML models, fancyimpute [88] for imputation. Additionally, I included specialized ML algorithms, such as XGBoost [89] and LightGBM [90].

3.4. Results

3.4.1. Software Requirements

The software requirements are intrinsically linked to the clinical modeling process itself and the identified challenges. Therefore, these requirements are the direct result of literature research carried out and the insights obtained from the expert workshop. In the following, I cover both functional and non-functional requirements.

The functional requirements were mapped to the steps for clinical predictive modeling and are laid out in detail in Table 3.1. The practitioners interviewed also pointed out a number of non-functional requirements. For one, they stressed the need for research clinicians and model developers to collaborate more closely in order to reduce the time needed for model evaluation. As such, tools would to need to be friendly to non-ML, i.e., clinical experts. That entails the necessity for visual model development, meaning without to the need to write code, install packages, etc., for example by means of a drag and drop interface. In addition, given that ML predictive modeling usually includes computationally intensive operations, such as hyperparameter optimization, the software employed should take advantage of the parallelization capabilities of the underlying hardware, be it cloud-based or on-premise.

Table 3.1.: Functional and non-functional requirements for development and validation of Clinical Prediction Models. Abbreviations: F=Functional Requirements; NF=Non-Functional Requirements.

User Stories or Requirements			Personas or Stakeholders	
Item	Step	Description	Research Clinician	Model Developer
F1	Preparation	**Candidate Predictors:** Provide a pre-defined list of candidate predictors derived from expert knowledge of the specific domain, e.g., in the form of predictors used in similar studies.	✓	
F2		**Medical Knowledge Graphs:** Utilize medical knowledge graphs to guide model development, which can inform how to transform predictors or assess clinical meaningfulness of results.	✓	✓
F3		**Use of Best Practices:** Provide assistance in adopting best practices while developing the models, such as how to choose a significant cohort, perform data transformation, and interpret results.	✓	✓
F4	Dataset Selection	**Cohort Exploration:** Enable interactive exploration of a given cohort, including descriptive statistics, such as standard deviation, min-max values, without coding necessary.	✓	
F5		**Common Data Formats:** Enable the use of harmonized or standard data formats for model development based on an existing initiatives, such as OMOP or FHIR.		✓
F6		**Common Data Semantics:** Provide harmonized semantics for model variables, e.g., how to define a given outcome or phenotype, consistency in extracting variables, agreed upon units of measurement, etc.	✓	✓
F7	Predictor Handling	**Support in Coding of Predictors:** Provide different ways on how to code categorical and continuous predictors, using label binarization, one-hot encoding, threshold setting and the like, since this directly affects model performance.		✓
F8		**Data Transformation and Pre-processing:** Provide data transformation tools to ensure common formats and semantics and data pre-processing techniques, e.g., normalization/scaling, as well as single (mean) and multiple imputation with k-NN or other methods.		✓
F9		**Feature Selection:** Utilize and compare different feature selection approaches, such as stepwise selection, t-tests and mutual information.		✓
F10	Model Generation	**Model Comparison:** Enable comparison of multiple candidate models side-by-side, i.e., provide different modeling algorithms out-of-the-box.		✓
F11		**Version Control:** Allow versioning to track data and model changes so as to enable comparisons across different versions, in order to ascertain the impact of changes to data, hyperparameters or algorithms.		✓
F12		**Formal Characterization:** Enable machine-readable description of models to ensure reproducibility and trackability, e.g., via Business Process Modeling Notation.		✓
F13	Model Evaluation and Validation	**Internal Validation:** Provide cross-validation and/or train-test split validation methods that enable a first assessment of a model's performance.		✓
F14		**External Validation:** Provide the possibility of running a given model upon new data with minor modifications if underlying data consistency is ensured, i.e, sharing of models across institutions seamlessly.	✓	✓

User Stories or Requirements			Personas or Stakeholders	
Item	Step	Description	Research Clinician	Model Developer
F15		**Standard Metrics:** Provide an extensible set of standardized metrics for discrimination such as c-statistic, ROC curve, precision, recall, etc., calibration and clinical usefulness.	✓	✓
F16	Model Interpretation	**Intelligible Explanations:** Provide explanations for blackbox models that are easily understandable by non-ML experts both at the global (model) and local (prediction) level.	✓	✓
F17		**Feature-based Importances:** Provide a quantification of feature importances for each model predictor to ascertain its role in the prediction.	✓	✓
F18		**Comparison of Interpretability Methods:** Provide unified view of different interpretability methods.	✓	✓
F19	Model Presentation	**User-friendly Scores:** Provide scores that make the application of the model easier, e.g., using paper calculations or nomograms.	✓	
F20		**Computer-based Frontend:** Provide developed scores by means of a web browser or smartphone app.	✓	✓
F21		**Integration into Clinical Software:** Enable seamless integration of the model into care workflows by means of contextual integration into existing clinical software.	✓	✓
NF1	Non-Functional	**Non-Expert User Friendliness:** Enable also non-expert level users to develop and validate models.	✓	
NF2		**Visual Model Development:** Enable visual development of predictive models, e.g, via drag and drop of pre-existing building blocks	✓	
NF3		**Parallelization Capabilities:** Enable parallelization of time-consuming tasks, e.g., hyperparameter optimization.		✓

3.4.2. Software System Architecture

The following sections provide detailed information on the architecture of the MORPHER platform, which makes use of Web and ML technologies to support clinical predictive modeling, comprised of different layers targeted at application, platform, execution and data concerns. The architecture is depicted as a Fundamental Modeling Concepts (FMC) block diagram in Figure 3.5. In the following, after briefly outlining my approach, each architectural component will be described in further detail.

Hybrid Approach (Cloud and On-Premise)

MORPHER enables research clinicians and model developers to create, share and validate CPMs using a combination of a cloud-based approach with on-premise, i.e., local, MORPHER instances running on composable docker containers [91]. The MORPHER cloud instance provides the necessary functionality to design CPMs with a visual interface using BPMN 2.0 notation. Models thus developed can be shared with other researchers/institutions to either 1) refine the model, 2) re-train it using existing data or 3) validate it using own local data. Data on model performance is provided in a standardized and centralized fashion. Given that the execution environment and trained models are equivalent, the tool enables reproducibility of experiments and validation of CPMs without significant overhead, as long as that data on both ends follow the same formats, e.g. Observational Medical Outcomes Partnership (OMOP) [92].

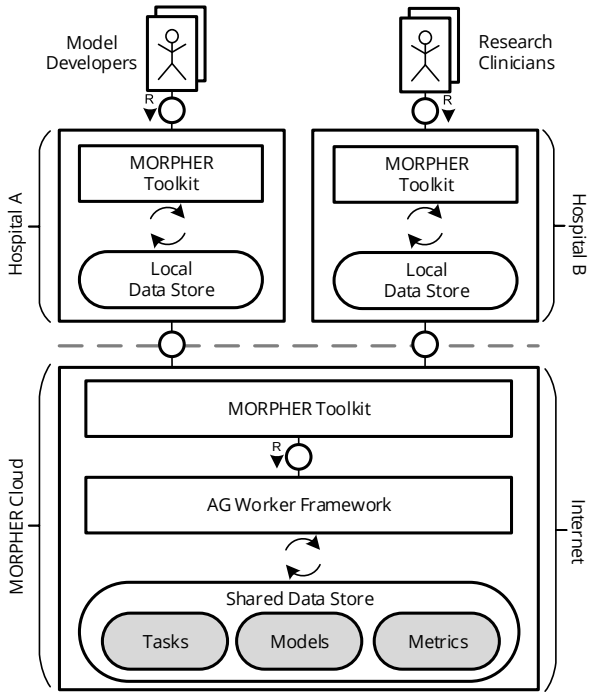


Figure 3.5.: MORPHER architecture depicted as a Fundamental Modeling Concepts (FMC) block diagram. Different institutions can collaborate exchanging models and metrics while keeping local data private.

MORPHER Cloud

This layer is concerned with providing the core application components, which are accessed by users via the Web interface, namely, both model developers and research clinicians. The Web frontend relies on HTML5/JQuery [93]. The application server backend is based on Python Flask [94]. Model developers are responsible for specifying CPMs in terms of loading, preprocessing, train and evaluate steps. Each of the processing steps can be visually specified by means of pipelines using standard BPMN 2.0 notation, which can be parameterized and executed. The Model Training and Evaluation component offers the Representational State Transfer (REST) interfaces that activate the ML functionalities exposed by the MORPHER Framework to train and evaluate different machine learning models using a standard set of metrics.

Local Data Stores

Data that is stored locally refers almost exclusively to the cohort data used for model development and validation. The different cohorts used as the basis for model development and validation are stored locally to safeguard privacy. Clinical researchers can import data from various sources and apply user-defined filters and transformations for predictor handling, such as binarization, normalization and imputation. It must be noted that advanced data preprocessing, such as pivoting, feature extraction from time-series data and data cleaning are not currently supported. An effort that varies starkly from case to case, this task does lend itself easily to automation and must be conducted in a previous step. MORPHER supports data in Comma-Separated Values (CSV) format, direct database connections with a user-defined database connection and a Structured Query Language (SQL) query.

Shared Data Store

Data that is stored centrally is concerned with task execution, i.e., model training itself, the thus trained models and performance metrics. Trained models are stored a structured fashion using jsonpickle [95]. In this way, models can be later retrieved and instantiated in order be applied on a different cohort, for validation purposes. Not only trained models, but also relevant metadata such as input/output and the algorithm's hyperparameters are available. Along with model metadata, data pertaining to true labels, predicted labels, and predicted probabilities after model applying the model are also stored in the repository. This enables flexible generation of model metrics, including discrimination, calibration and clinical usefulness. User data for login and audit purposes are also stored centrally. The database backend currently relies on Postgres, using a combination of relational and document-based (JSON) objects.

Analyze Genomes Worker Framework

As the execution environment for model training, evaluation and interpretation, we built upon existing work carried out by Schapranow et al. [86]. The AnalyzeGenomes.com platform provides a runtime environment for BPMN-based process models initially designed for genomic data processing pipelines [86]. It consists of a design environment supporting graphical modeling and an execution environment, which can run arbitrary program code. Each step of the modeled process pipeline is defined in corresponding Python script implemented with the MORPHER Toolkit. Amongst others, it supports common tasks involved in data loading, preprocessing, training and evaluation of multiple ML algorithms.

MORPHER Toolkit

The tasks to be executed are defined via this component. It is a Python package that handles the most common tasks involved in data loading, preprocessing, model training and evaluation of multiple ML algorithms. For data handling, such as label binarization or one-hot encoding, the package utilizes the library Pandas [87]. Since medical data is often very sparse, this package encapsulates routines for single and multiple imputation, e.g., k-NN, using the library fancyimpute [88]. Model training is provided by the ML library scikit-learn [45]. Currently, the framework supports the algorithms decision tree, random forests, gradient-boosting decision tree, logistic regression and multilayer perceptron, among others. More algorithms can be added with ease following the defined software interfaces. Furthermore, the software interface is designed to support multiple ML backends, i.e., though I use scikit-learn at this point, the Toolkit are not dependent on it.

In model training, another important aspect is hyperparameter optimization. The Toolkit provides it by performing grid search across a range of different parameters, optimizing via cross-validation of AUROC. Consequently, by providing a simplified for a number of routine tasks involved in modeling based on existing utilities, this component can be easily utilized on its own, for scripting purposes, or embedded in fully-fledged applications. Table 3.2 provides an overview of the functionalities implemented and the options to researchers via the MORPHER Toolkit Application Programming Interface (API).

Table 3.2.: Implemented features provided via MORPHER Toolkit

Step	Functionality	Options
Dataset Selection	Retrieving Data [87, 96]	Loading CSV Data Loading SQL Data
	Inspecting Data [87, 97]	Summary Statistics Ordinary Least Square Regression P-values
Predictor Handling	Data Imputation [45, 88]	Mean Imputation Soft imputation k-Nearest Neighbors
	Data Sampling [98]	Synthetic Minority Over-sampling Technique Random Over Sampler Random Under Sampler Cluster Centroids
	Data Encoding [45]	Label Binarizer Label Encoding One Hot Encoding Ordinal Encoding
	Data Scaling [45]	Standard Scaler Robust Scaler Normalizer Quantile Transformer
Model Generation	Model Training [45, 89, 90]	Decision Tree (DT) Random Forest (RF) Logistic Regression (LR) Multilayer Perceptron (MLP) Gradient Boosting DecisionTree (GBDT) Elastic Net (EN) Support Vector Machine (SVM) Naive Bayes XGBoost LightGBM
	Internal Validation [45]	k-Fold Cross Validation Test-Split Validation
Model Generation	Hyperparameter Tuning [45]	Exaustive Grid Search

Step	Functionality	Options
	Calibration [45]	Isotonic Calibration Platt's Scaling
Model Evaluation and Validation	Discrimination Metrics [45]	Area Under the ROC Curve Area Under the Precision-Recall Curve Precision, Recall, F-Score Diagnostic Odds-Ratio
	Calibration Metrics [45]	Calibration Slope Calibration Intercept Brier Score
	Clinical Usefulness [48, 99]	Net Benefit (Treated) Net Benefit (All / None)
	Plotting [45, 100]	Receiver Operating Characteristic (ROC) Curve Precision-Recall (PR) Curve Calibration Curve Net Benefit Curve
Model Interpretation	Feature Importance [59, 52, 62, 58]	Method-based Feature Importance Local Interpretable Model-Agnostic Explanations (LIME) Shapley Values Mimic Learning
	Plotting [100]	Feature Importance Plots Summary Feature Importance (Weighted Explanations) Interpretability Heatmap

3.4.3. System Implementation

In this section, I provide details on system implementation, including database design, class diagrams and sequence diagrams.

Database Design

The appropriate implementation of the requirements identified required a suitable database design. I employed a hybrid solution comprising both a relational and an object-based component reliant on JavaScript Object Notation (JSON) that could support the flexible nature of the platform. In brief, I outline the main entities that make up the platforms database design, which are the following:

Cohorts. Predictive modeling work relies critically on a given patient cohort. Basic information such as name and user (owner) are supplemented by dynamic information in JSON format, such as features, transformations, filters and the like. The cohort data itself is stored in Comma-separated Values (CSV) format with a link for this entity. Upon cohort upload, summary information on the cohort is stored in a separate JSON file, removing the need for generating those statistics every time the user wants to explore the cohort with the package Pandas Profiling [101].

Models. The output of a training a ML-algorithm is a model, which must be stored in the database for posterior use, e.g., model validation. The trained binary object itself is stored in the database as JSON, allowing it to be retrieved and re-instantiated at a later stage. Meta-level data such as features used, model target, and hyperparameters is stored in a JSON field 'params'.

Experiments. The act by the user of training, validating or interpreting a model is understood at the database level as an 'experiment'. Additionally, this entity also keeps track of key model performance metrics for later retrieval (discrimination, calibration and clinical usefulness). Therefore, it is possible to retrieve the metrics of training or validating any combination of model and cohort.

Predictions. In order to plot many of the model evaluation diagrams of the platform, such as ROC and calibration or clinical usefulness curve, it is necessary to record the model predictions (predicted probabilities and labels) for each cohort sample, i.e., patient, along with the expected label. This makes it possible to calculate deviations from the expected outcomes, which can then be provided to users upon model evaluation.

Other Entities. A central aspect of MORPHER is enabling collaboration and sharing. For this purpose, the platform needs to keep track of users, shared cohorts and shared models, as well as roles. As such, a user can generate a model that can be shared with a second user to validate it on his own local cohort.

Figure 3.6 depicts the Entity-Relationship diagram of MORPHER. Note that much of the inherent complexity is abstracted by means of JSON attributes, such as 'params', which make it possible to extend the data stored as new features are added to the application. For instance, if new metrics for a given model were to be added to the platform, the database structure would remain the same, only a new property added to the JSON definition in the 'params' attribute of the 'experiments' entity.

Class Diagram

The class diagram of MORPHER Cloud closely matches the data model depicted in Figure 3.6, therefore being reflected via Object-Relational Mapping (ORM) in application classes by means of SQL Alchemy [96]. As such, for the class diagram, I focus on the MORPHER Toolkit, the component that automates routine clinical modeling tasks. The Unified Modeling Language (UML) class diagram depicted in Figure 3.7 illustrates more closely how the Toolkit is embedded into the existing AG Worker Framework and how it can be easily extended.

The core of the Toolkit is the `MorpherJob`, containing services shared by all sub-classes, such as access to the Cloud API. This job is compliant with the class signature of the Worker Framework jobs,

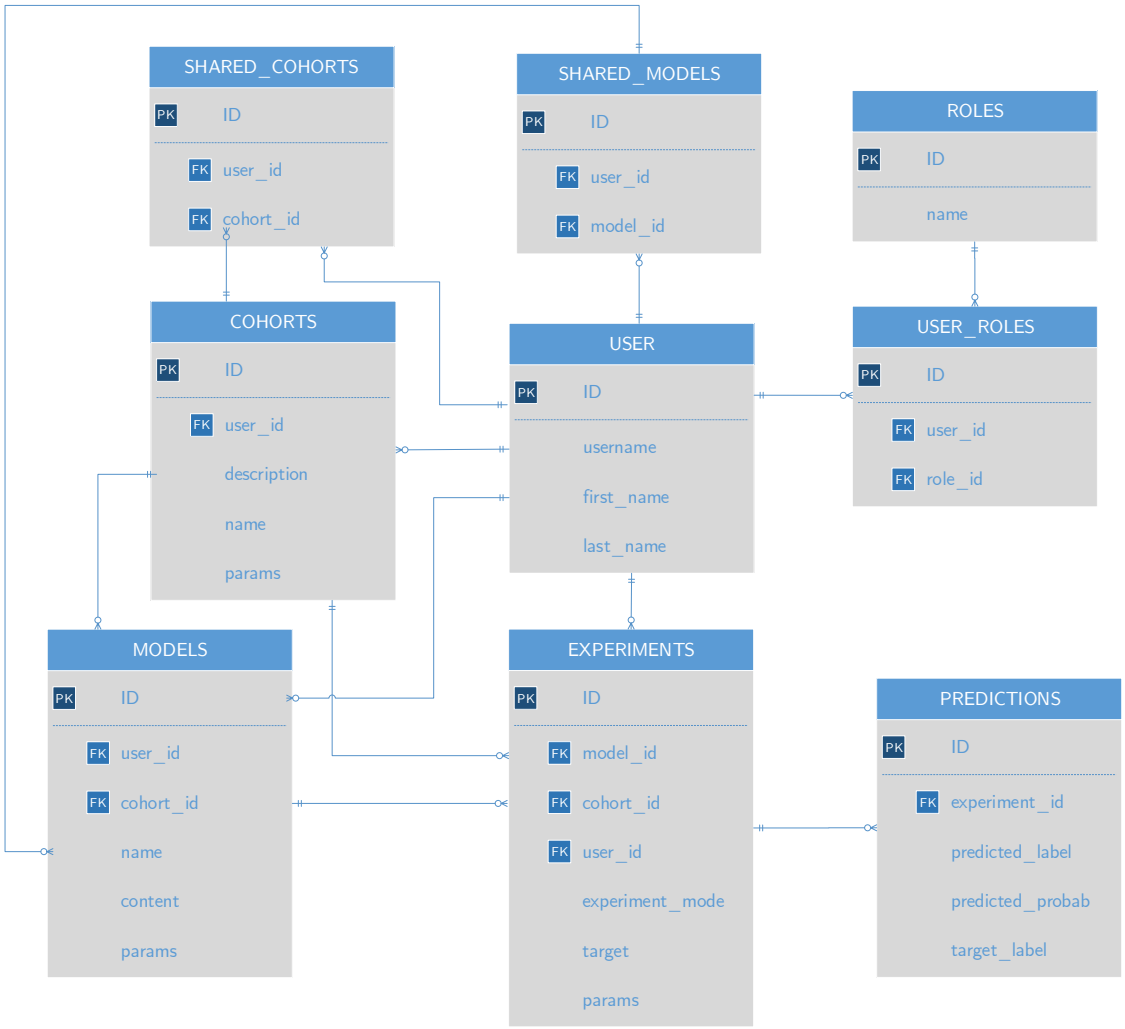


Figure 3.6.: Entity-Relationship diagram of MORPHER database. Flexible extensions of the data model are possible via JSON attributes, ensuring continuous platform development with limited impact to the data model's structure.

Listing 3.1: Implementation of LightGBM algorithm in MORPHER Toolkit. kwargs construct provides the means to pass on keyword arguments to the constructor, e.g., particular hyperparameters.

```
from mopher.algorithms import Base
import lightgbm
"""
    Base class defines methods such as
    fit(), predict() and predict_proba()
"""
class LightGBM(Base):
    def __init__(self, **kwargs):
        clf = lightgbm.LGBMClassifier(**kwargs)

    ''' call Base constructor '''
    super().__init__(clf, **kwargs)
```

i.e., it inherits from that class. Since the Worker Framework can dynamically load external libraries compatible with its API, one is able to define an arbitrary number of jobs. This job-based approach makes it possible to implement at the code-level the steps identified clinical modeling process (cf. Figure 2.1).

For example, the Job `Train` makes use of ML models defined in the `algorithms` module for model generation. These algorithms inherit from a base class, which offers the functionality for model fitting and predicting in a centralized fashion. As such, for extending the collections of algorithms, it suffices to inherit from `Base` and provide the custom code based on any existing implementation. For example, the algorithm LightGBM [90] can be included in the Toolkit by using the code in Listing 3.1.

The algorithms thus defined can be instantiated at runtime, according to the parameters passed on to the method `execute`. A similar approach is followed by the `Impute` and `Explain` jobs, where the custom classes are defined and can be instantiated according to parameterized method calls.

Example Implementation

The different jobs available in the Toolkit can be used in standalone mode, i.e., without an instance of the AG Worker Framework. As such, they can be combined at will within a pipeline. Let us assume a patient cohort is available and we want predict a target outcome using a range of algorithms such as LR, Decision Tree (DT) and GBDT. Further, we want to conduct split internal validation using a test size of 20% of the available data. Finally, missing data shall be imputed using k-Nearest Neighbors (kNN). This task might require several lines of code by for all the intermediate steps. With MORPHER, it can be substantially simplified by chaining the Toolkit jobs. Listing 3.2 illustrates the approach.

Sequence Diagram

In order to illustrate how the Local and Cloud instances of MORPHER can support the work of two researchers, I make use of a Unified Modeling Language (UML) sequence diagram, depicted in Figure 3.8. First, a researcher in Clinic A uploads her cohort *locally* while making metadata available to the cloud instance. The Cloud then triggers a request for training for the MORPHER Local instance in Clinic A. A model is then trained locally and the processing result, i.e., the model binary and model performance (metadata) is sent back to the Cloud via a REST API and shared to researcher in Clinic B. In a second step, a researcher in Clinic B proceeds similarly and uploads his cohort metadata to

Listing 3.2: Demo implementation of a predictive model for diabetes using the MORPHER Toolkit jobs. Note that the dictionary 'results' contain model metrics for the trained algorithms.

```
import morpher
from morpher.config import algorithms, imputers
from morpher.jobs import *
from morpher.metrics import *

''' define the input file and target variable '''
filename="cohort.csv"
target = "diabetes"

''' First we load, impute and split the dataset in train and test '''
data = Load().execute(filename=filename)
data = Impute().execute(data, imputation_method=imputers.KNN)
train, test = Split().execute(data, test_size=0.2)

''' Then we train the given algorithms on the training set '''
models = Train().execute(
    train,
    target=target,
    algorithms=[
        algorithms.LR,
        algorithms.DT,
        algorithms.RF,
    ]
)

''' and evaluate them on the test set '''
results = Evaluate().execute(test, target=target, models=models)
```

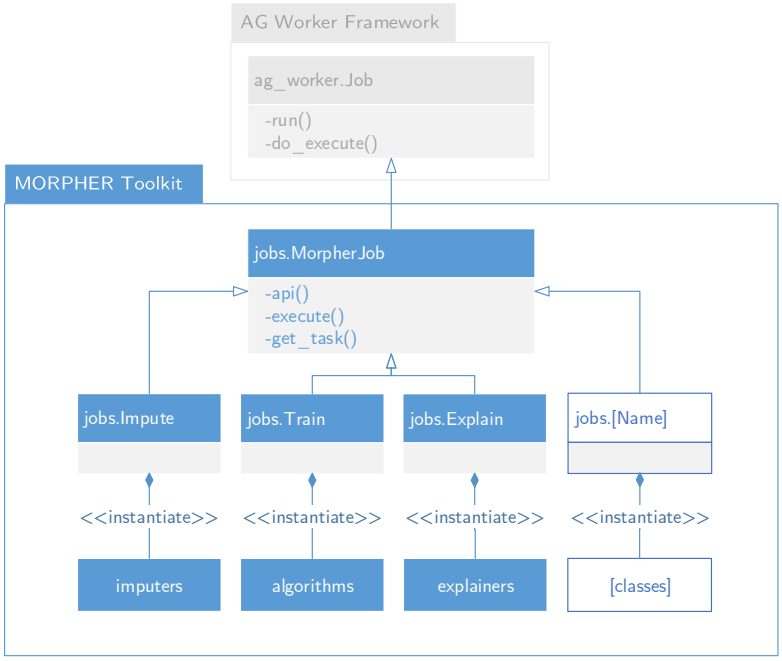


Figure 3.7.: Unified Modeling Language (UML) class diagram of MORPHER Toolkit. MORPHER Jobs inherit from the base Worker Framework class Job, thereby making it possible for these jobs to be executed in the existing Worker Framework infrastructure.

the cloud while keeping her physical data locally. Now, instead of training a new model, she simply retrieves the model trained by researcher A and requests a validation, which also happens in the local computing infrastructure). Likely, this validation generates metadata (i.e., performance metrics), including discrimination, calibration and clinical usefulness. Finally, researcher in clinic B can requests the model developed to be interpreted using local data.

The communication between the Cloud and Local instance takes place via Advanced Message Queuing Protocol (AMQP) supported by the AG Worker Framework infrastructure. More specially, the Cloud instance runs a message broker – Redis [102] – which queues processing requests from registered worker nodes running Celery [103]. Processing nodes can only register within the message broker by means of a secret key, to prevent attacks. The MORPHER Toolkit jobs can be executed locally by Celery. The Toolkit, in turn, communicates the result back to the Cloud instance via REST.

Cohort Metadata

The cohort metadata that is sent from local nodes over to the cloud contains only textual, non-identifying information. The Cloud node exposes API which are accessed by the Local clients. These local clients are authenticated via JSON Web Token (JWT). The information that is sent over to the cloud about the local cohort is limited. The variable `params` contains data such as the local filename, column names, and any cohort transformations. The server responds with the `id` of the newly created cohort.

When a `train model` command is issued from the cloud to train a model with a local cohort, it is picked up by the local worker node. The local worker node then starts the local training pipeline for the specific cohort.

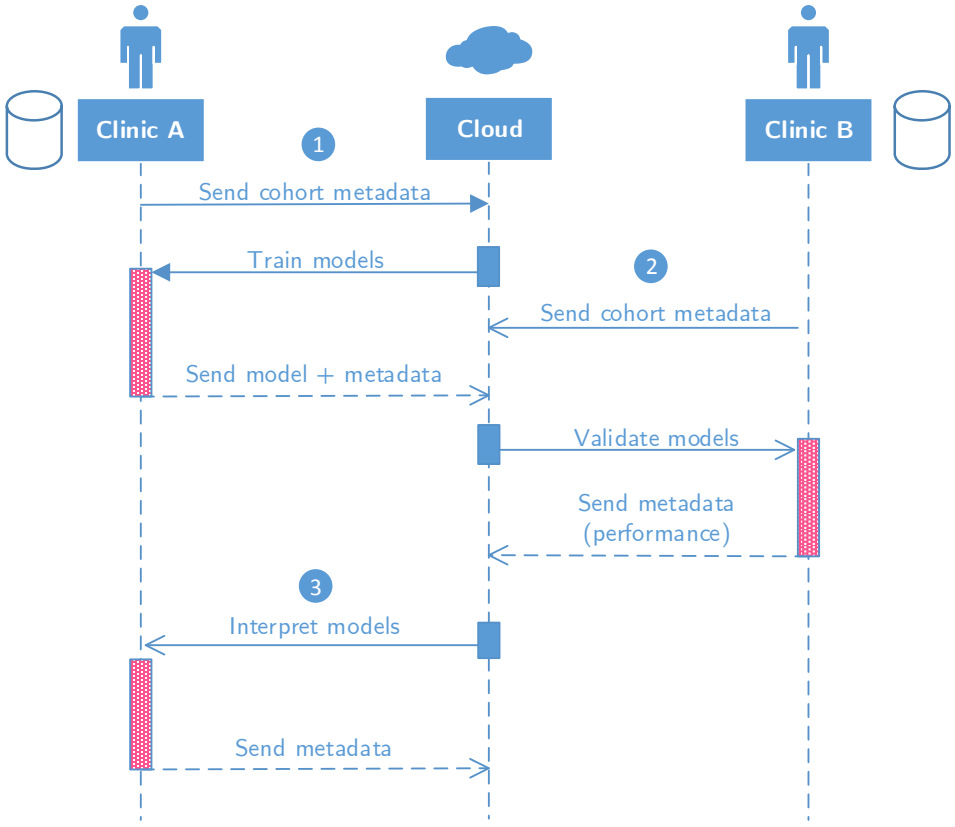


Figure 3.8.: Unified Modeling Language (UML) sequence diagram displaying how MORPHER can support the workflow of two researchers. The colored bar highlight processing that takes place locally. Cloud and local instances communicate by means of AMQP.

User Journey

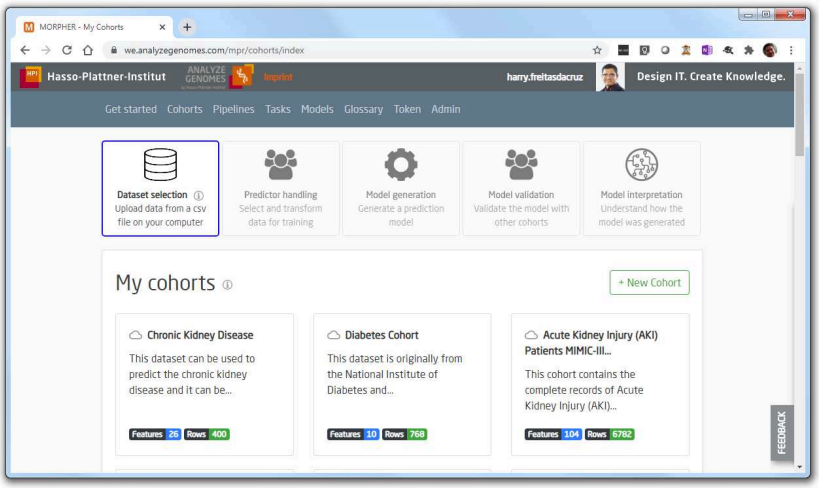
The user journey in MORPHER follows the clinical modeling steps laid in Section 2.2, i.e., dataset selection, predictor handling, model generation, validation and interpretation. The user can select from a list of available cohorts, which have been previously either 1) uploaded or 2) shared with the respective user. Note that if the cohort is a local cohort, i.e., residing in a remote local, only textual information about the specific cohort is available. Then, the user can select from a list of available cohorts, which have been previously either 1) uploaded or 2) shared with the respective user. Note that if the cohort is a local cohort, i.e., residing in a remote local, only textual information about the specific cohort is available. After selecting the respective cohort, the user can explore the respective features, or define feature transformations. This action is then followed by model generation, in which parameters can be manually specified, such as imputation method, algorithms, etc. Optionally, an automated mode cycles through all available options. After model generation, the user can inspect model metrics and verify feature importances with interpretability methods. Figure 3.9 shows the first three steps in exemplary screenshots.

3.5. Evaluation

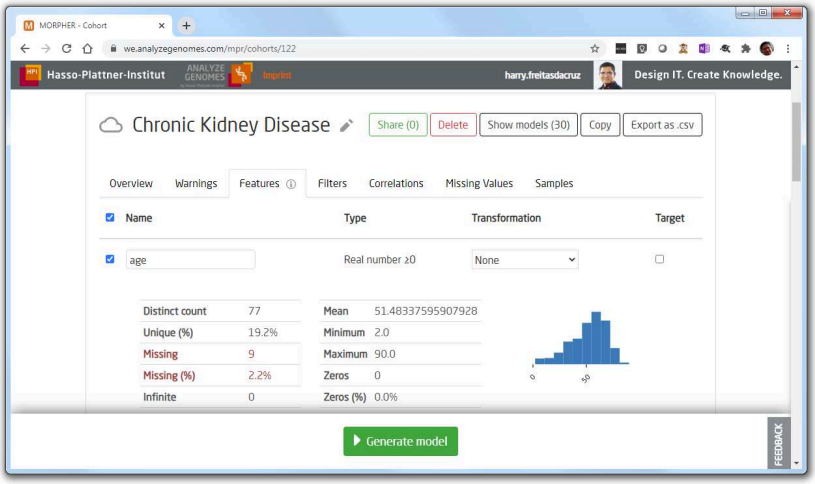
In this section, I evaluate the proposed platform from three perspectives. First, I carry out a functional comparison to existing approaches in clinical predictive modeling taking into account the requirements laid out. Second, I analyze how the platform performs in comparison to one selected tool from the list of related work considering different dimensions. Finally, by means of semi-structured interviews with experts, I seek to assess technology acceptance with regards to the platform. In the following, each aspect will be described in detail, including study design, study subjects and results. A joint discussion is then presented in Section 3.6.

3.5.1. Functional Perspective

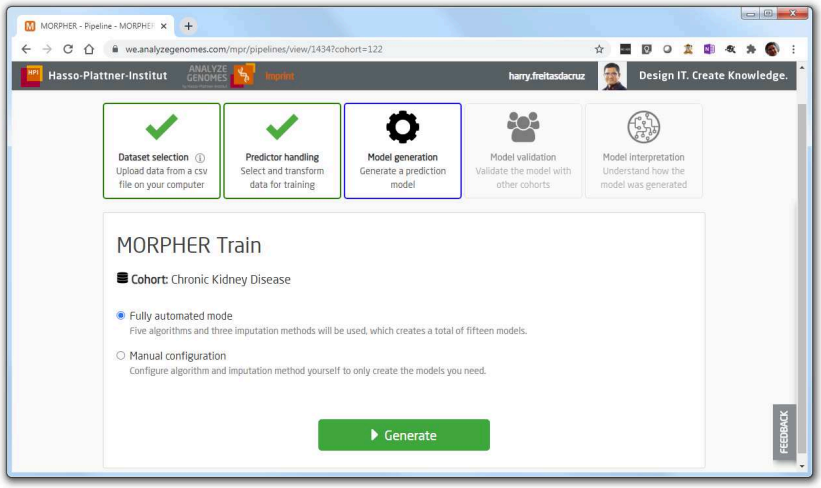
In the final assessment for the functional perspective, software tools were included which allow a clinical researcher to develop machine-learning based predictive models without the need to code, i.e., enabling visual use, thereby excluding mathematical software, ML software toolkits and biomedical software that require coding skills, such as the PLP R package. Their degree of compliance with the requirements gathered was indicated by a solid, half-filled and empty circle as depicted in Table 3.4. Some of these tools, such as ML suites, make it possible to rapidly prototype models, yet they do not cater to the specific needs of clinical modelers. In addition, extant IT-tool support provides no easy way to seamless share models across institutions.



(a) Cohort selection screen. User sees available cohorts.



(b) Cohort exploration screen. Descriptive statistics for selected column.



(c) Model generation using fully automated mode or manual configuration.

Figure 3.9.: Exemplary screenshots from MORPHER displaying the user journey. Note that the respective clinical predictive modeling steps reflected on the tool itself.

Table 3.3.: Overview of selected REST APIs from MORPHER Web with the corresponding endpoints, request and responses. Requests are placed by Local clients via JWT authentication.

Endpoint	Request	Response

cohorts/create
Creates a new local cohort.
	`{`
	`"name":'...',`
	`"description":'...',`
	`"parameters": params,`
	`"user_id": current_user.id`
	`}`
		`{`
		`"cohort_id": cohort_id`
		`}`

cohorts/user
Get a list of cohorts from current user.
	`{`
	`"user_id": user_id`
	`}`
		`[`
		`{`
		`"name":'...',`
		`"description":'...',`
		`"parameters": params,`
		`"user_id": user_id`
		`}, ...`
		`]`

cohorts/[id]
Updates a given cohort, e.g., share cohort with other users.
	`{`
	`"cohort_id": cohort_id,`
	`"params" : params,`
	`"name": name,`
	`"description":description,`
	`"shared_users": [users],`
	`}`
		`{`
		`"OK" or "Not Authorized"`
		`}`

Table 3.4.: Tools surveyed. Abbreviations: GP-General Purpose, BSP-Biomedical Software Platform. Legend: ●–Fully supported; ◐–Partially supported; ○–Not supported.

Category	Tool	Functional Requirements																					Non-Functional		
		Preparation			DS Selection			Pred. Handling			M. Generation			Eval. & Validation			Interpretation			Presentation			NF		
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	NF1	NF2	NF3
GP	Statistical Packages [63, 64, 65]	○	○	○	●	○	○	●	●	●	●	○	○	●	●	○	○	●	○	○	○	○	○	○	●
	ML Toolkits [45, 68, 69, 70, 104]	○	○	○	◐	○	○	●	●	●	●	○	○	●	●	○	○	●	○	○	○	○	○	○	●
	Scientific Computing [74, 67, 105]	○	○	○	●	○	○	●	●	●	●	○	◐	●	●	○	○	●	○	○	○	○	○	◐	●
	ML Suites [74, 72, 73, 75]	○	○	○	●	○	○	●	●	●	●	○	●	●	●	○	○	●	○	○	○	○	◐	◐	●
BSP	tranSMART [106]	◐	○	○	●	●	○	○	○	○	◐	○	○	●	○	○	○	○	○	○	○	○	○	○	○
	ATLAS [77]	●	○	◐	●	●	○	○	○	○	●	○	○	●	◐	◐	○	○	○	○	○	○	◐	◐	○
	MLBCD [79]	○	○	○	◐	○	○	○	◐	●	○	○	○	●	○	◐	○	◐	○	○	○	○	◐	●	●
	ExplICU [80]	◐	○	○	◐	○	○	○	○	○	○	○	○	●	○	◐	○	◐	○	○	◐	○	◐	○	○
	PLP R package [78]	●	○	◐	○	●	○	○	○	○	●	○	○	●	●	◐	○	◐	○	○	○	○	○	○	◐
	MORPHER	●	○	◐	●	○	○	◐	◐	◐	●	●	●	●	●	●	●	●	●	○	○	○	●	●	●

3.5.2. Clinical Modeling Task

To compare my platform to a similar state-of-the-art competitor, I designed a clinical modeling task to be carried out within a specific time frame. I then extract a number of metrics from the task such as user activity, as well as task completion and correctness metrics. For the test, I selected the tool RapidMiner Go (RMG) [3], a recently published automated version of RapidMiner [14].

Study Design. To conduct the assessment, we designed a clinical modeling task to be carried out within a set time frame of one hour / 60min. The task entailed a given dataset which should be preprocessed and a number of modeling algorithms to be trained followed by model evaluation and interpretation. The users where then required to fill out a report which was developed to cover relevant items suggested by the TRIPOD criteria [7]. To help analyze the results and avoid possible biases, I additionally required the participants to answer questions regarding their previous knowledge and experience conducting clinical predictive modeling. Figure 3.10 provides an overview of the proposed study design. The subjects were thoroughly informed on the research study and consented to have their answers analyzed for this thesis via an electronic consent form on Google Forms.

Study Subjects. The study participants included a cohort of Digital Health and IT-Systems Engineering master program students Hasso Plattner Institute with little to no exposure to clinical predictive modeling. Also, the study subjects had no previous experience with either tool, nor with the clinical task itself. A pre-selection identified 91 candidate subjects, who were contacted via e-mail. The final cohort was composed of 12 subjects (13% response rate), out of which half were assigned to MORPHER and half to RMG, via stratified randomization using Graphpad [107].

Modeling Task. The clinical modeling task entailed developing a number of CPMs for CKD. The data utilized the fully anonymized Pima Indians dataset openly available on-line, therefore no data restrictions apply [108]. The task was divided into sections following the clinical process modeling steps laid out in Figure 2.1, namely, dataset selection, predictor handling, model generation and evaluation and model interpretation. The subjects received a document containing a comprehensive description of the task. They then utilized an on-line questionnaire on Google Forms to provide their answers to questions posed. The modeling task questionnaire is available in the Appendix and was structured according to each process step, totaling 25 items. The items covered questions such as number of predictors, imputation strategy, imputation strategy, feature importance, etc.

The total time alloted for the task was 60min, with 15min for general question answering, 15min for watching the introductory video (tutorial), 15min to perform the task on the tool and finally 15min for filling out the form. Subjects were instructed to finalize the questionnaire after 15min, even if not completely answered.

Evaluation Metrics. The evaluation metrics utilized address three aspects. The first is concerned with *user activity* while performing the clinical modeling task. For this, I measured time elapsed for modeling, number of mouse clicks and mouse moves, distance covered by the mouse and an overall index of mouse activity and number of keyboard keys pressed. In order to record this information automatically, I developed a Java application using JNativeHook[4] which ran on the subject's computer. Subjects were instructed to start and end the monitoring tool while utilizing the respective clinical modeling tool. The second metric sought to assess how many of the total items on the task questionnaire could be answered according to each process step, i.e., *task completion*. Finally, the answers provided were compared to a reference in order to assess *task correctness*, i.e., out of the items reported how many of them were correct.

Results. The results obtained for the user activity metrics are depicted in Figure 3.12 as violin plots. In addition to the information present in a box plot, violin plots also present the probability density

[3] https://go.rapidminer.com/am/
[4] https://github.com/kwhat/jnativehook

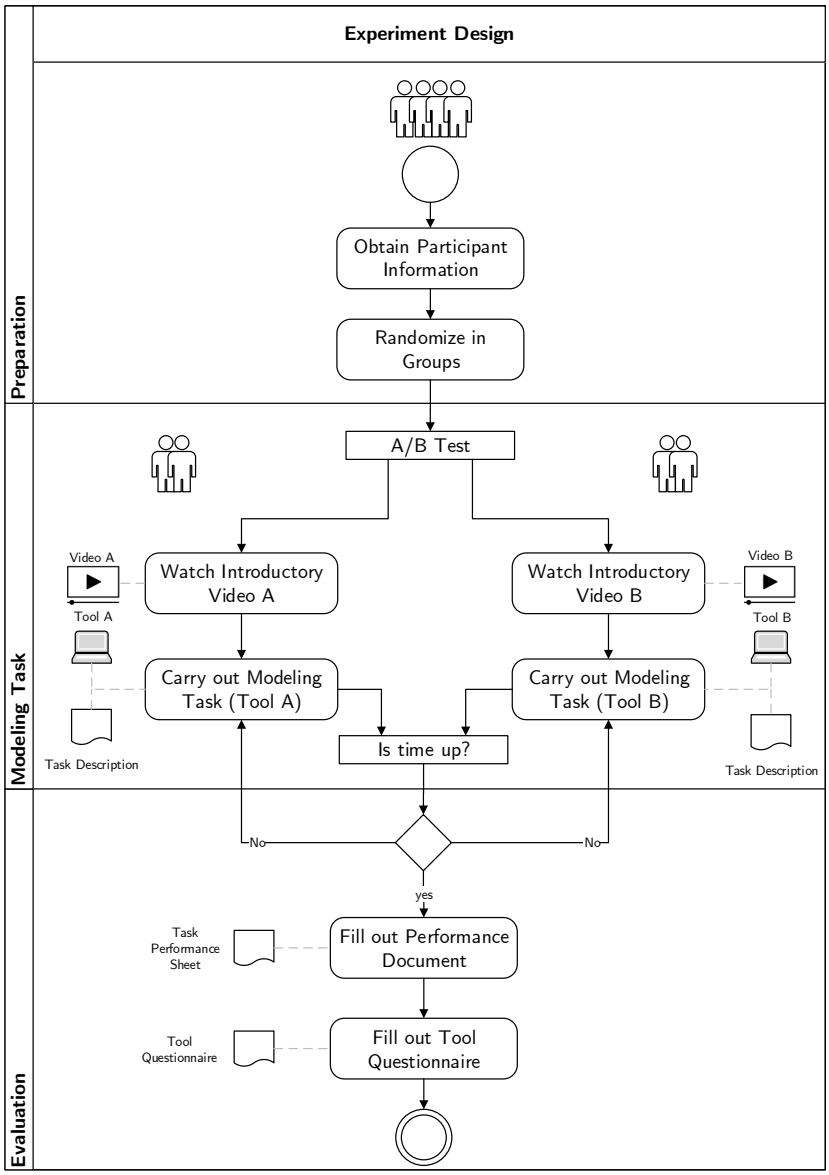


Figure 3.10.: Characterization of the study design as UML diagram. Before the clinical modeling task, users are randomized, and each group is assigned either MORPHER or another tool. The users first watch a video explaining how to use the respective tool, i.e., a tutorial, and then receive a document with task description. The whole study procedure was capped at one hour.

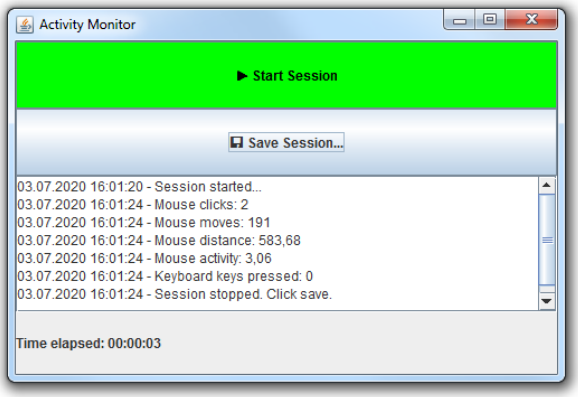


Figure 3.11.: Activity monitor developed for the user test. It records user activity when active. Subjects were asked to provide the monitoring report generated by the tool.

of the data at different values, smoothed by a kernel density estimator [109]. Note that the metrics 'mouse moves' and 'mouse distance', as measured by the Euclidean distance in pixels between two points, are provided in natural log scale (ln) for better visualization.

In the study cohort under analysis, time elapsed for MORPHER had a wider range variance than RMG, albeit with a similar median (white dot) at 7.5 and 6.5 minutes respectively. The interquartile range (black bar) for mouse clicks and mouse distance differed substantially across tools, to the detriment of MORPHER. With respect to mouse moves, MORPHER presents more uniformity in comparison to RMG, i.e., lower variance and extreme values. However, mouse activity (mouse distance in pixels (px) / mouse moves) had both highest frequency (probability density) between 5 and 10px/move. Finally, regarding keyboard input, the range of variance was markedly higher for MORPHER, with substantially smaller keyboard input necessary for RMG.

A normality test for each of the metrics using Shapiro-Wilk test returned normality for all variables except time elapsed and keyboard keys pressed at $\alpha < 0.05$. However, small samples often pass the normality test, since these tend to have low power to reject the null hypothesis, i.e., that data is normally distributed [110]. Therefore, following Collins et al.'s recommendations, I opted to utilize the non-parametric Mann-Whitney U statistical test for calculating p-values, which takes into account medians rather than means of independent samples [111]. Under this test, the null hypothesis assumes distributions of both populations are equal. At $\alpha = 0.05$, the null hypothesis can be rejected, i.e., differences are statistically significant, for mouse clicks and keyboard keys pressed.

With respect to performance at modeling task, namely completion and correctness, the results obtained are reported in Figure 3.13 for each of the clinical modeling process steps and tools respectively. Task completion measured how many of the total items in each step could be answered (regardless of being correct or not). Results for predictor handling and model evaluation were similarly complete, with subjects using MORPHER completing approximately 20% fewer items in the interpretation step as compared to RMG, 64% vs. 22% respectively. As for the model generation step, MORPHER subjects were able to answer 100% of the items required vs. 88% in RMG. As to task correctness, MORPHER presented substantial advantages with respect to model generation (100% vs. 63%), while performing comparatively worse in model interpretation, 64% vs. 86% for RMG. While predictor handling was similar for both tools, MORPHER presented an advantage with respect to data selection and model evaluation. Similarly to user activity, given the limitations of the sample size, which hampers the use of parametric tests, I utilized the Mann-Whitney U statistical test to measure statistical significance. At the selected significance level, $\alpha < 0.05$ no significant results could be obtained. Table 3.5 shows the results obtained for each metric/process step.

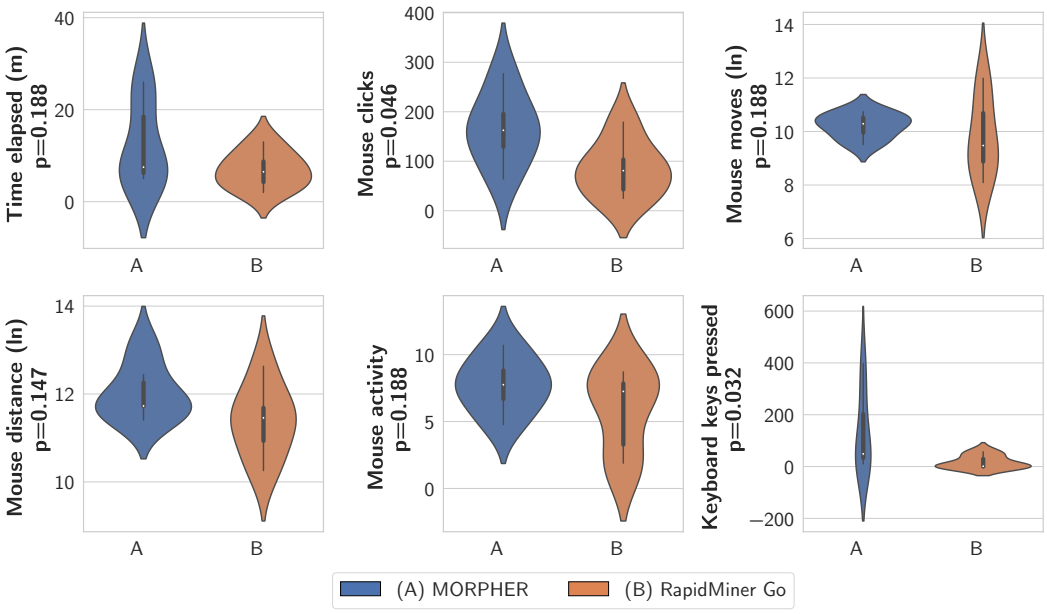


Figure 3.12.: User activity in comparison for both tools, including time elapsed, mouse clicks, mouse moves and distance, both natural log-scale (ln), mouse activity and keyboard keys pressed for N=12 (6 each tool). For $\alpha < 0.05$, mouse clicks and keys pressed are statistically significant.

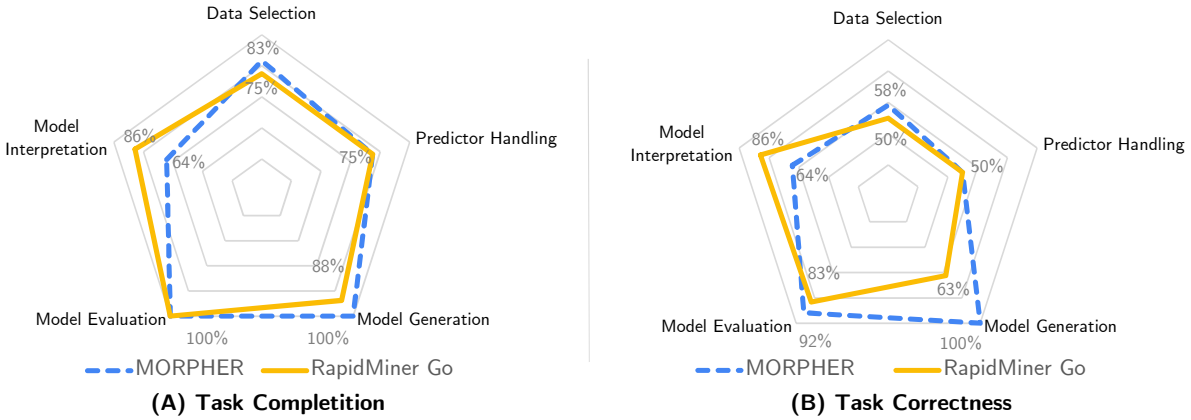


Figure 3.13.: Task completion (A) and correctness (B) for both tools along the different process steps for N=12 with 6 for each tool.

Table 3.5.: Statistical testing of modeling task results utilizing Mann-Whitney U test comparison MORPHER and RapidMiner Go. Null hypothesis assumes that the two samples come from the same distribution. At $\alpha = 0.05$ no significant results could be obtained.

P-value Metric	Dataset Selection	Predictor Handling	Model Generation	Model Evaluation	Model Interpretation
Task Completion	0.402	0.465	0.168	0.202	0.283
Task Correctness	0.256	0.339	0.090	0.192	0.342

3.5.3. Technology Acceptance

To assess the extent to which one could use the developed platform in actual clinical practice, I carried semi-structured interviews with practitioners coming both from a data science and clinical research. Additionally, drawing the on the Unified Theory of Acceptance and Use of Technology (UTAUT), we evaluated both the performance expectancy and effort expectancy of my approach [112].

Study Design. In order to qualitatively assess the degree to which the solution developed could be accepted by potential users, I carried out a number of semi-structured interviews limited to a time frame of 60min. In these, subjects had the opportunity to interact with the tool assisted by the interviewer, reply to questions and pose questions themselves as needed, following extant literature recommendations [113] on qualitative research. Informed consent was obtained by participants via an electronic formular. Participants were thus asked, how could this tool:

- Question 1: support clinical predictive modeling? This was aimed to elicit highlights and positive aspects identified by the interviewees, i.e., positive aspects, and/or advantages.
- Question 2: be detrimental to clinical predictive modeling? Question targeted at understanding potential pitfalls and short-comings not currently addressed., i.e., negative aspects, and/or disadvantages.
- Question 3: be improved to reflect actual needs? Item focused on identifying unmet needs, important in the context of clinical predictive modeling.

The interview responses were then clustered according to the clinical predictive modeling process steps (cf. Section 2.2) and repeated mentions of the some topic/aspect were counted. The interview section of the study was followed by a structured on-line questionnaire aimed at capturing potential users' perspectives in a structured fashion, available in the Appendix.

Study subjects. Study subjects comprised a cohort of experts either working as research clinicians or data scientists with varying levels of experience with clinical predictive modeling, allowing a balanced view of the platform. The interview cohort include 17 subjects in total (N=17) with 7 reporting a predominantly data scientist background and 10 reporting a predominantly clinical research and/or medical background. They were interviewed in person and via Zoom. Interviewing subjects for acceptance or usability studies is always associated with time and cost investment. A trade-off must be then met between effort and utility. As guide for such studies, Nielsen & Landauer developed a mathematical model in which maximum cost/benefit ratio at four interviews, with costs still paying off at 16 interviews, at which point marginal utility is decreased [114].

Evaluation Metrics. Utilizing two of the independent variables of Venkatesh et al.'s UTAUT, I sought to investigate prospective user technology acceptance in terms of two constructs: Performance Expectancy (PE) and Effort Expectancy (EE).

On the one hand, PE seeks to ascertain how the users expect the given tool to perform. It is subdivided into three other subconstructs. The first one is *perceived usefulness*, which establishes whether practitioners would be willing to use the tool develop in future projects of their own. The second is *job fit*, which seeks to assess to what extent the system capabilities improves job performance. Finally, *relative advantage* establishes the degree to which a given tool is better than its precursor [112].

On the other, EE aims to capture users' perceptions on how complex it would be use the tool, since a specific learning curve must be overcome before a user can be said to be proficient in tool use. Similarly to PE, its constituent subconstructs are three-fold. First, *perceived ease of use* helps to assess users' perceptions with regards to how effortless they perceive the tool to be. Second item refers to *complexity*, aimed at eliciting the extent to which a system is hard to use and/or understand. Finally, *ease of use* captures the degree to which subjects believe the tool is ease to use. Each of the metrics

has a number of items associated with it, which have been presented to test subjects in a Likert-type scale [112].

Table A.2 provides an overview of each construct and metric along with the attending items.

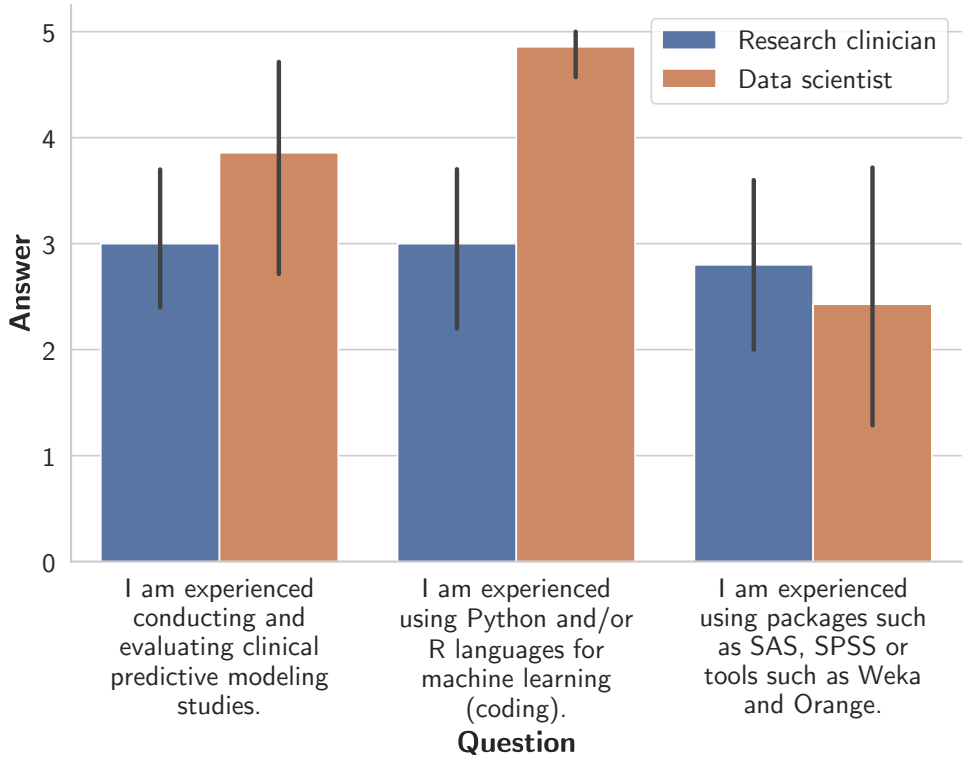


Figure 3.14.: Profile of subjects interviewed (N=17). Data scientists consider themselves on average to be more knowledgeable in the topic.

Results. In the following, I report on the results of the semi-structured interviews with experts and the results obtained concerning technology acceptance questionnaire.

Table 3.6 provides the result of the semi-structured interviews clustered by process step. The numbers in circle refer to how often that specific item was mentioned across the interviews. With respect to *Question 1*, i.e., positive aspects, in particular research clinicians highlighted often the fact that MORPHER is easier to use in comparison to other tools, considering it a valuable tool for use in health research. Additionally, the possibility to support collaboration between research clinicians, develop quick prototypes and enable model validation between clinics was considered relevant by at least three interviewees. Data scientists, conversely, highlighted other aspects, such as the availability of comprehensive graphs for evaluation. Regarding *Question 2*, i.e., negative aspects, the item most often cited during the interview by data scientists was the lack of knowledge of clinical predictive modeling by clinical researchers (six times). Conversely, this was cited by research clinicians themselves only half as often (three times). In particular, other aspects not related to the process (NR) in Table 3.6 featured prominently during the interviews. Examples are legal challenges on storing models and cohorts, i.e., data privacy, and difficulty interpreting the results of model performance. When it comes to suggestions, i.e, *Question 3*, a summary of the statistical methods used, e.g., following TRIPOD, was comparatively often mentioned during interviews (three times) followed by the need of a video tutorial on how to use the tool. Data scientists, in turn, highlighted the need for mode advanced forms

of data pre-processing and formatting and, among others, the possibility to use a 'fully-automated mode'. This mode of operation would completely automate modeling with default parameters, which could be optionally fine-tuned.

Table 3.6.: Expert interview results. Numbers in circles illustrate how often the item was mentioned. Abbreviations: DS=Dataset Selection, PH=Predictor Handling, MG=Model Generation, EV= Evaluation and Validation, MI=Model Interpretation, NR=Not Related to Process.

Question	Item	Step	Research Clinician	Data Scientist
Question 1	Easier to use in comparison to other tools	MG	(4)	–
	Cohort exploration features are helpful	DS	(3)	–
	Supports the collaboration of data scientists and medical experts	MG	(3)	–
	Increases transparency (options for modeling available)	MG	(1)	–
	Glossary feature is helpful	MG	(1)	–
	Possibility to learn while using the tool	NR	(1)	–
	Provides reference on existing models	EV	(1)	–
	Helps researchers keep an 'experiment journal' while modeling	MG	(1)	–
	Comprehensive graphs for evaluation	EV	–	(2)
	Provides ready-to-use baselines for further algorithm development	EV	–	(1)
	Useful tool for use in health research, including own use cases	NR	(4)	(1)
	Enables validation across clinics	EV	(4)	(1)
	Can be used to generate model prototypes quickly	MG	(3)	(1)
Question 2	Difficulties interpreting the results	MI	(2)	–
	Limited usefulness for practicing doctors	NR	(2)	–
	Fewer parameters in comparison to other tools	MG	(1)	–
	Lack of transparency in the methods used	MG	(1)	–
	Does not support cohort extraction features	PH	(1)	–
	Lack of trust from IT departments	NR	(1)	–
	Makes it easier to make mistakes while modeling	MG	–	(2)
	Limited handling of categorical data, e.g., imputation	PH	–	(2)
	Lack of data quality control/analysis	DS	–	(1)
	Lack of knowledge on clinical predictive modeling by medical researchers	NR	(3)	(6)
	Legal challenges on storing models and/or cohorts (data privacy)	NR	(3)	(2)
	Different terminology than that used by doctors	NR	(1)	(1)
Question 3	Summary of statistical methods used, e.g. using TRIPOD	EV	(3)	–
	Provide summarized statistical information for cohorts	DS	(1)	–
	More elaborate transformations for features (threshold binning, etc.)	PH	(1)	–
	Cohort matching function for model validation	PH	(1)	–
	Improve validation metrics: graph types, mean values, etc.	EV	(1)	–
	Include a warning that only anonymized data can be uploaded	DS	(1)	–
	Functionality to download models for further refinement	MP	(1)	–
	Provide different internal validation set-ups, cross-validation, etc.	EV	–	(1)
	Include more advanced forms of model validation and graphs	EV	–	(1)
	Provide functionalities for federated learning	MG	–	(1)
	Provide help on what models/settings to use	MG	–	(1)
	Provide fully-automated mode (auto-parameter setting)	MG	–	(1)
	Advanced data handling for data pre-processing and formatting	PH	–	(2)
	Support for deep learning algorithms	MG	–	(1)
	Recommendation engine for modeling settings	MG	–	(1)
	Perform model interpretation during training	MI	–	(1)
	Short video and/or tutorial on how to use the tool	NR	(2)	(1)
	Support for other data import files, such as SPSS and Excel	DS	(1)	(1)

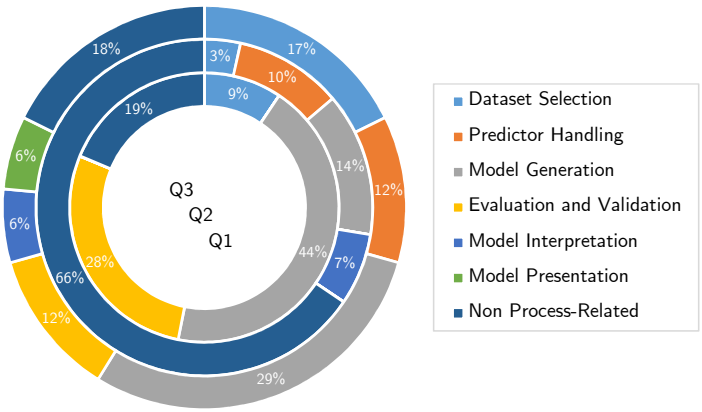


Figure 3.15.: Compiled categorized interviews. Abbreviations: Q1-Q3: Questions 1 to 3 respectively, from the inner to outer circles.

Furthermore, Figure 3.15 displays how often each of process steps were mentioned throughout the interviews, by combining the counts for each item in Table 3.6 for both research clinicians and data scientists. This is an attempt at capturing what process steps seem to be most relevant for the interviewed subjects. Positive aspects regarding model generation were mentioned 44% of the time, followed by evaluation and validation (28%) and non-processed related issues (18%). For negative aspects, non-process related issues dominated the interview items with 66% of items referring to such issues. Model generation also featured prominently in suggestion for improvement, followed by non-process related issues with 18% and predictor handling, as well as evaluation and validation, both with 12%.

Finally, Figure 3.16 and Figure 3.17 display the different subconstructs for PE and EE, respectively, as diverging stacked bar charts, an often recommended visualization approach for Likert-scale items [115]. To assess the internal consistency of each subconstruct, I applied Cronbach's alpha, which should optimally be higher than 0.7 [116]. A higher than 0.7 Cronbach's alpha was obtained for all subconstructs in both PE and EE.

Regarding PE, perceived usefulness, job fit and relative advantages have been to a large extent well-received by the respondents, with the bulk of answers leaning towards the positive end of the scala (strongly agree). Some respondents, however, were much more critical, particularly with respect to relative advantage, where a higher proportion of neutral and negative responses can be ascertained. When it comes to EE, perceived ease of use, complexity, and ease of use were seen favorably, with most answers lying in the positive spectrum (agree and strongly agree). For reporting and consistency, coding for the subconstruct 'complexity' was reversed, with 'completely disagree' coded as a favorable answer and so forth. In the final analysis, one item of this subconstruct removed from the analysis, because of an error in the on-line formular.

3.6. Discussion

Discussion shall focus on the three aspects covered in the evaluation, namely, comparison to current approaches with respect to the functional perspective and clinical modeling task, as well as technology acceptance.

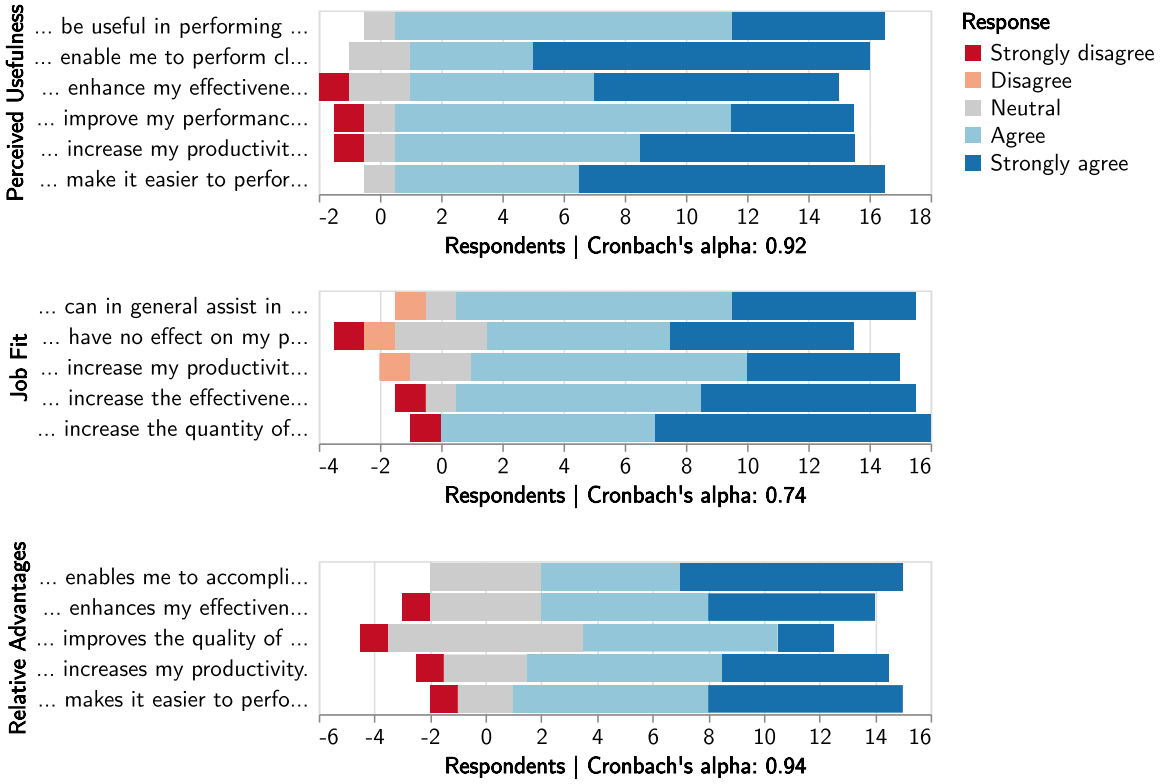


Figure 3.16.: Results obtained for the construct Performance Expectancy (PE). Evaluates the extent to which the tool is likely to fulfill user's performance expectations.

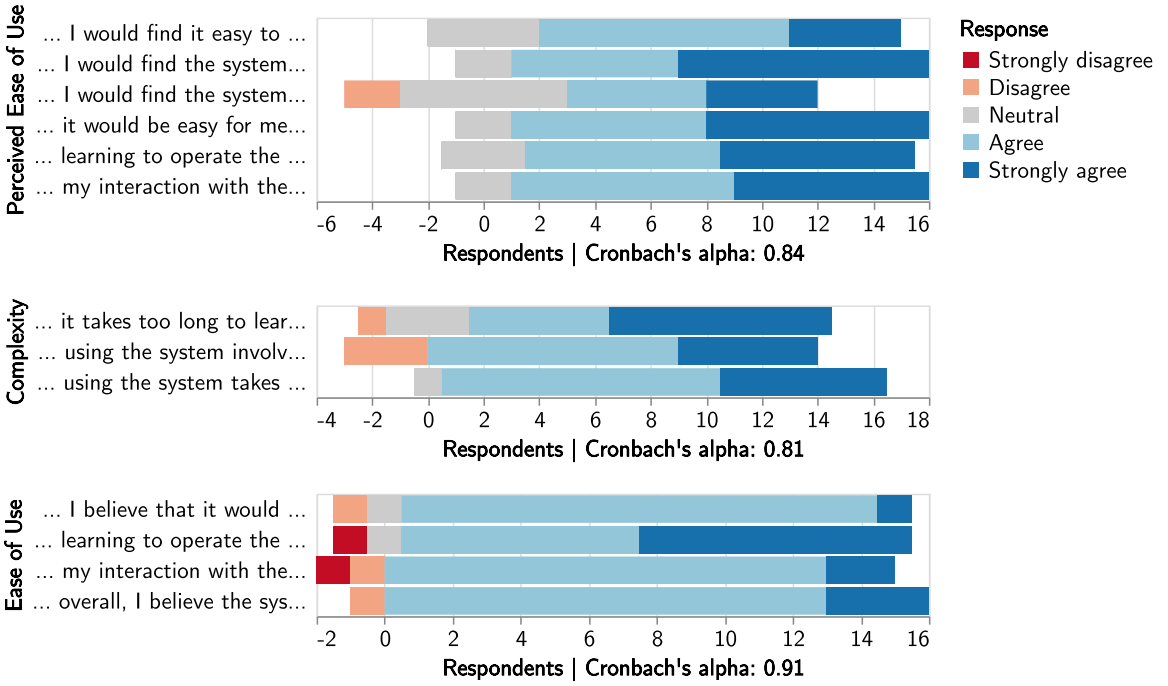


Figure 3.17.: Results obtained for the construct Effort Expectancy (EE). Evaluates the how the user perceives the effort required to learn and to use the tool .

3.6.1. Functional Perspective

Examining the landscape of IT support for clinical predictive modeling as laid out in Table 3.4, researchers seem to be faced with a trade-off: on the one hand, general-purpose tools, feature-rich, but requiring high levels of technical expertise; on the other, biomedical software that automates to a certain extent modeling tasks but fail to address critical aspects of the process. Statistical packages are currently the de facto standard for model development, but using them in a collaborative context, e.g., for model validation could prove cumbersome. Additionally, support for data standards is scarce or non-existent in those tools, except the ones provided by the OHDSI initiative[5] such as ATLAS and PLP package [78].

ML Suites such as RapidMiner manage to automate prototyping to a considerable extent, relying on easy-to-use, visual interfaces. Focused on general practitioners, rather than clinical modelers, these tools might require a steeper learning curve than my approach. Furthermore, these tools offer no seamless way to share models across institutions, though admittedly this could be achievable by means of manual file exchange. Even in such a scenario, comparing model metrics would require manual summarization, unlike my approach, in which metrics across different validation experiments are kept in a centralized fashion. Moreover, even though biomedical software does present considerable advantages to practitioners, they typically do not offer the tools needed for ad-hoc data handling, such as imputation, scaling and the like. Exception to this observation is the tool MLBCD, which offers automatic pivoting of features [79]. Finally, the use of the AG Worker Framework automates job execution and utilization of computational resources via paralelization of long-running tasks, which cannot be easily achieved unless custom extensions are used, such as Weka-Parallel [117].

[5] https://www.ohdsi.org/

In summary, the use of on-premise and cloud resources, i.e., a hybrid approach with MORPHER, makes it possible to retain crucial information on model metrics, algorithms hyperparameters and the model themselves centrally while keeping sensitive data, i.e., patient data, within the confines of the participating institutions. Without a similar set-up, manually keeping records of individual metrics and exchanging models via e-mails and data privacy issues substantially hamper collaboration. As such, tool developers should conceive of IT solutions that encompass the broader scope of clinical modeling needs in collaborative environments.

3.6.2. Clinical Modeling Task

Overall, the results obtained during the user activity assessment seems to indicate a clear advantage of RMG over MORPHER. While the statistical significance of the results could not be established for all user activity metrics, differences in mouse clicks and keyboard keys pressed seem to be particularly significant ($p < 0.05$). This is explained by the fact that RMG is a highly automated ML tool, where very little freedom is given to the user to parameterize the experiments. This makes the tool indeed very convenient to use, but at the expense of transparency. Conversely, MORPHER automates the predictive modeling process only to a certain extent, requiring manual input for certain parameters, such as imputation methods. In effect, this was a conscious design decision: the TRIPOD guidelines recommend that researchers report on a number of key decisions made, such as validation strategies, imputation methods, etc. As such, while MORPHER is somewhat more laborious to use, notably with respect to keyboard input, users can expect more transparency, even if connected to a slower learning curve.

This aspect is more clearly illustrated in the results obtained in the metrics task completion and task correctness. Except for the step 'model interpretation', MORPHER was at least equivalent or better, most noticeably with respect to 'model generation' when evaluating task correctness. The results poor results obtained by MORPHER with regards to interpretation can be explained by the fact that the feature importances were presented in different user views within the tool, as opposed to RMG, where feature importance information is contained in one page. Future iterations of MORPHER should take this account. It is worth noting, however, that while RMG provides only one interpretability method, MORPHER provides several, including a comparison of methods.

3.6.3. Technology Acceptance

In the interviews conducted, I reached the numbers recommended by Nielsen & Landauer [114]. As the number of interviews increases from the optimum (four), the same aspects re-emerge across interviews, as illustrated by the counts in Table 3.6. Also considering users' perceptions on performance and effort expectancy, results indicate that research clinicians tended to view the tool's use more favorably than data scientists. This is likely evidence of the fact that MORPHER is geared towards rapid prototyping of CPM, therefore lacking advanced configuration options desirable from a data scientist's perspective. Admittedly, striking the right balance between automation and freedom of configuration a complex undertaking. In particular, data scientists highlighted the potential of making mistakes by researchers who might not be knowledgeable in the field of clinical predictive modeling. While this is a valid criticism that applies to any support tools, it must be noted that MORPHER is oriented towards the TRIPOD guidelines, in an effort to achieve more transparency. At any rate, following this feedback, I implemented a 'fully automated mode', with optional manual configurations (cf. Figure 3.9).

Furthermore, aspects not directly related to the clinical modeling process itself, featured prominently in the interviews, as illustrated by Figure 3.15. These merit extended discussion, most notably regarding two aspects. First, lack of knowledge on clinical predictive modeling seems to be a concern for both data scientists and research clinicians, which could hamper utility and adoption of the tool.

Indeed, this aspect is indeed a concern and has been reflected in methodological shortcomings in publications in the field [9]. This is mitigated by MORPHER in that standardized pipelines are available (and reusable) which have been developed and validated by data scientists in collaboration with research clinicians. Further, ample use of video tutorials explaining the basics of the field, especially with regards to evaluation of CPMs could be embedded in the tool itself. The current version of MORPHER supports extensive glossary items that provide quick explanations of importance concepts, along with the pertaining literature references.

Second, issues pertaining to data privacy and trust also played a substantial role during the interviews. More specifically, research clinicians and data scientists alike raised concern with whether data stored in MORPHER complies with privacy regulations, be it clinical data or the models themselves. With regards to data uploaded to the cloud instance of MORPHER, it is up to the user to make sure that the data is anonymized or at least k-anonymized, and a proper warning has been included to that effect. For the metadata that is uploaded from the local to the cloud instance (cf. Figure 3.5), it comprises only descriptive statistical data, which is usually also required for peer-reviewed publications. However, the Fundamental Law of Information Recovery postulates that "overly accurate" answers to too many questions may completely compromise privacy [118]. Therefore, summary statistics is not entirely exempt of privacy risks. Similar criticism can be leveled at storing and making available *trained* ML models: these models have theoretically the capacity to 'memorize' information on the training data and be thereby potentially vulnerable to privacy attacks [119]. In order to overcome those pitfalls, the field of privacy-preserving data analysis seeks to establish 'differential privacy', making it possible to learn useful information on the population but nothing on the individual [120]. This field of study and the accompanying precautions and strategies to achieve differential privacy lies beyond the scope of this dissertation. Nevertheless, production-grade deployment of a similar solution should include an extensive treatment of those aspects.

3.6.4. Limitations

In this section, I outline some of the limitation inherent to the platform and the evaluation studies developed, i.e., limitations on the functional view, clinical modeling task and technology acceptance.

Functional View. A number of limitations apply to my platform (MORPHER). First, this work is concerned with prognostic classification models. Time-to-event prediction models are also relevant in research and will be implemented in future iterations. Second, in its current state, the platform still does not fully address the myriad of issues pertaining to data preparation prior to modeling. Complex tasks such as pivoting and feature extraction from time-series data must be executed outside the platform. A similar issue applies to datasets used for validation: collaborating institutions should make sure that the trained models are either 1) based on standard formats in the first place, such as OMOP, or 2) that the validation dataset is matched to the derivation dataset in predictor definitions and format. Given the arbitrary complexity of this task, currently the platform does not support it. Alternatively, advanced users can extract the required data from a OMOP-compliant database the package `InspectOMOP` [121]. Finally, we addressed only functional requirements concerning modeling, validation and model interpretation, but other areas are equally important, such as model presentation and integration into clinical care, which shall be addressed in future work.

Clinical Modeling Task. Studies such as the one proposed usually cannot avoid spurious effects resulting from selection bias [118]. I sought to mitigate this effect by sampling from a somewhat homogeneous population without advanced knowledge on predictive modeling itself, but one cannot rule out the influence of other factors not controlled for, such as familiarity with IT tools in general. Particularly for MORPHER, variance in the results seemed higher than for the the control group (RMG). This effect is likely further exacerbated by the small sample size of the study. Even calculating the required sample size is difficult, since the overall population of clinical researchers cannot be

estimated accurately. Furthermore, as demonstrated by the statistical tests conducted, significance at $\alpha = 0.5$ could not be established for all the metrics utilized, thereby casting doubt on the validity of the results obtained. Another limitation is the nature of the clinical modeling task itself: time constraints required a simulated use case, real-world usage beyond the scope of a constrained test might reveal a different pattern altogether. This underscores the need for further testing in future work.

Technology Acceptance. While also subjected to similar selection bias effects, the wide spectrum of experience and expertise in the interviews carried out contributed positively to the range of feedback obtained. However, given the limited scope of the interview (60min), it can be argued that the time was not sufficient for the subjects to develop a truly informed opinion of whether they could accept the technology or not. A scenario in which the tool had been continuously used over an extended period might provided a less biased estimate. Further, inherent to the interview instrument is the idea of 'courtesy bias', in which interviewees tend to provide positive assessments because just they do not want to 'offend' those seeking their option [122]. Regardless of this effect, the interviews carried out provided valuable insights into how the tool can be further developed.

3.7. Conclusion

Clinical predictive modeling is a complex undertaking that requires adequate tool support. An examination of existing related work revealed that different solutions address the needs identified to varying degrees, with gaps concerning particularly model development and validation. While challenges remain concerning aspects such as data harmonization, privacy-preserving data mining and the potential vulnerability of ML models to privacy attacks, MORPHER seeks to mitigate some of these deficiencies. It facilitates the development and validation of CPMs making use of a visual interface for modeling that is based on a standard notation (BPMN 2.0) that can be executed in distributed computing infrastructure. MORPHER's virtualized architecture leveraging Docker can be run either on-premise or on the cloud, providing the appropriate infrastructure for model exchange and collaboration between research clinicians and model developers. As such, the tool can help expedite the model prototyping and validation while making a contribution towards standardization and reproducibility.

Building on the insights gained during development and evaluation, implications for future development of tools support clinical predictive modeling include the following:

Provide better support for data harmonization. While there is a growing trend towards standardization in EHR data, this is a long process, which requires conversion of existing datasets. These datasets could be potentially used for predictive modeling and model validation if comprehensive data harmonization tools are integrated into existing solutions. MORPHER offers support for a couple of basic data transformation, such as imputation, encoding, etc. but does not support complex scenarios, such as concept mapping.

Improve predictive modeling education. The interviews revealed that concepts related to the development and especially evaluation of predictive models are not readily accessible to researchers. While MORPHER integrates an on-screen glossary and a step-by-step process, predictive modeling tools present an opportunity for targeted learning if they include an 'educational mode'.

Strike a balance between configuration and automation. This aspect goes hand-in-hand with the previous one: while automation hides complexity from the user, transparency is also required, especially when it comes to publications. As such, tools should offer automation, but optional configuration. In MORPHER this is achieved by means of an optional 'fully-automated mode'.

Provide more transparency with regards to data privacy. In MORPHER, the only information that leaves the local instances are cohort information that can't be in any way traced to individual patient

that and serialized ML models. However, this is not readily evident and relies on trust. Predictive modeling tools should provide users with the possibility to verify that privacy cannot be harmed, for example by means of third-party security certifications.

Support federated learning. A growing body of research is now focusing on learning over distributed datasets in the form of federated learning [123]. User-facing predictive modeling tools should also provide functionalities that enable this approach of distributed learning. While not currently implemented in MORPHER, new Jobs could be added to the Toolkit to address this scenario with relative ease.

Code Availability

The cloud version of the platform is accessible on-line [6]. The machine learning component, MORPHER Toolkit, is openly accessible via GitHub [7].

Credits

Next to my contributions in conducting requirements engineering, designing, developing and evaluating the technical artifacts discussed, the following credits are due with respect to the work presented in this chapter. Orhan Konak and Benjamin Bergner contributed towards evaluation and comparison to other tools. Philipp Bode implemented extensions to the AG Worker Framework to support MORPHER Jobs. Conrad Lempert implemented improvements to make the MORPHER Web interface more user-friendly. Matthieu-P. Schapranow served as technical advisor and reviewed the final manuscript of the MORPHER paper [1].

[6] https://we.analyzegenomes.com/mpr
[7] https://github.com/hpi-dhc/morpher-toolkit

4. Case Study: Acute Kidney Injury

"I asked the doctor; he said he couldn't live more than three days. But can they be sure?"

—Leo Tolstoy, *Anna Karenina (1878)*

Can machine learning help predict complications? In this chapter, I make use of the functionalities provided by MORPHER to address the development, validation and interpretation of a predictive model for a concrete use case, tackling prediction of Acute Kidney Injury (AKI) following a surgical procedure. An introduction will be followed by related work, methods, results and discussion sections. In each of these sections, issues pertaining to development, validation and interpretation are dealt with. This chapter illustrates the issues mentioned in the earlier parts of this work.

4.1. Introduction

Patients suffering with chronic or acute heart disease can end up requiring surgery to help manage or cure this condition. In particular, surgeries utilizing a cardiopulmonary bypass are particularly deleterious for the kidneys, potentially giving rise to AKI. Indeed, AKI affects up to 30% of all patients after cardiac surgery. Among others, this intervention has been linked to a series of complications, such as increased mortality, comorbidities and adverse outcomes for patients [18].

Identifying patients at high risk for developing AKI *before* the surgical intervention can assist care providers in adopting targeted renal-protective strategies, such as increasing renal blood flow and avoidance of medications that can impact the kidneys [124]. There is little consensus on what drugs can effectively prevent AKI onset in heart patients. However, early detection can be relevant for perioperative patient management and clinical trial recruitment [125]. Therefore, a number of studies have been targeted at developing accurate risk scores and CPMs for this disease.

Previous work dealing with the task of predicting heart surgery-associated AKI take into account biomarkers and/or clinical data before, during and after the surgical intervention. Particularly concerning biomarker-based approaches, measurements of interest are usually taken after the surgery [126], thus posing barriers for use prior to the intervention. In this work, I derive a CPM which utilizes only preoperative variables in order to predict the onset of AKI. This CPM is based on clinical data collected prior to surgery from MIMIC-III [127], here referred to as *derivation cohort*. For this task, I compare the performance of different prediction algorithms, including both easily interpretable, as well as complex ensemble approaches.

ML algorithms are prone to learning particular features of a dataset, which might not be transferable to other datasets. Therefore, to investigate how the CPM developed performs on completely unseen data, I discuss the results of applying this model on two external *validation cohorts*, one from a German and another from an American university hospitals, German Heart Center (DHZB) and Mount Sinai Hospital. It is worth mentioning that, while essential, validation of prediction models is only rarely performed in the literature [9].

ML techniques, e.g., deep learning, tend to perform better than linear modeling approaches when it comes to discriminative performance, but often fall short of providing adequate interpretability.

Indeed, striking the right balance between predictive performance and interpretability is key for deploying clinical predictive models in practice [128]. Therefore, in this work, I make use of different interpretability approaches to provide interpretable explanations for the model predictions and discuss to what extent these results reflect medical knowledge.

This chapter's experimental set-up is illustrated by the Business Process Modeling Notation (BPMN) diagram in Figure 4.1.

4.2. Related Work

The following sections outline related work regarding the development, validation and interpretation of prediction models for AKI.

4.2.1. Model Development

Current work on AKI prediction after heart surgery relies on serum or urine biomarkers, such as neutrophil gelatinase-associated lipocalin [129]. Concerning prediction models using electronic health data, an early approach has been the Cleveland score [130] derived from a large cohort of open-heart surgery patients, which was followed by the publication of the AKICS score based on a cohort of Brazilian patients [131]. In a multicentric, multinational study, Mehta et al. developed a score using the National Cardiac Surgery Database of the Society of Thoracic Surgeons (STS), achieving satisfactory results [132]. Prediction models developed until now have been largely based on logistic regression, with the AKICS score presenting the best performance upon derivation. Even considering possible overfitting effects, the AUROC of most models range from 0.74 to 0.84 [133].

More and more, ML techniques are also being utilized for AKI prediction. Thottakkara et al. utilized a number of ML techniques, with Generalized Additive Model as the best-performing for AKI prediction in a large patient cohort (N=50,318) with AUROC=0.858 [134]. With a substantially smaller cohort (N=212), Legrand et al. achieved AUROC=0.760 using a super learner estimator [135]. In a similarly sized cohort of heart surgical patients (N=212), Eyck et al. were able to achieve AUROC=0.834 [136]. An ensemble of learners achieved the best result in a cohort of North American patients with AUROC=0.740 with N=25,521 [137]. Flechet et al. achieved AUROC=0.84 using random forests on a general surgery multicenter cohort (N=2,123) for AKI stage 2-3 [138]. Furthermore, in a cohort of Korean patients (N=2,010), Lee et al. reported AUROC=0.78 with gradient boosting machine (XGBoost) [139]. More recently, Tomašev et al. within Google DeepMind used Recurrent Neural Networks (RNN) on a large-scale cohort of US veterans (N=703,782), achieving [140] AUROC=0.83. In contrast, my model based on GBDT achieved AUROC=0.90 (N=6,782).

4.2.2. Model Validation

As a rule, model performance obtained on the validation cohort is usually poorer when compared to that of the derivation cohort, with models often showing "disappointing accuracy" on new cohorts [143]. Since the publication of those scores, a literature review recommended the Cleveland score, as it is the most frequently validated score [133]. However, the Cleveland score presented poor discrimination metrics when validated in a Chinese cohort, suggesting limited generalizability for populations not predominantly Caucasian [142]. The Simplified Renal Index (SRI) strived to achieve a succinct set of predictors, but ultimately performed worse than the Cleveland score on a validation cohort of the Mayo Clinic [141]. It is worth noting that excepting the work of Flechet et al. and Tomašev et al. none of the other works offered an external validation of their ML models.

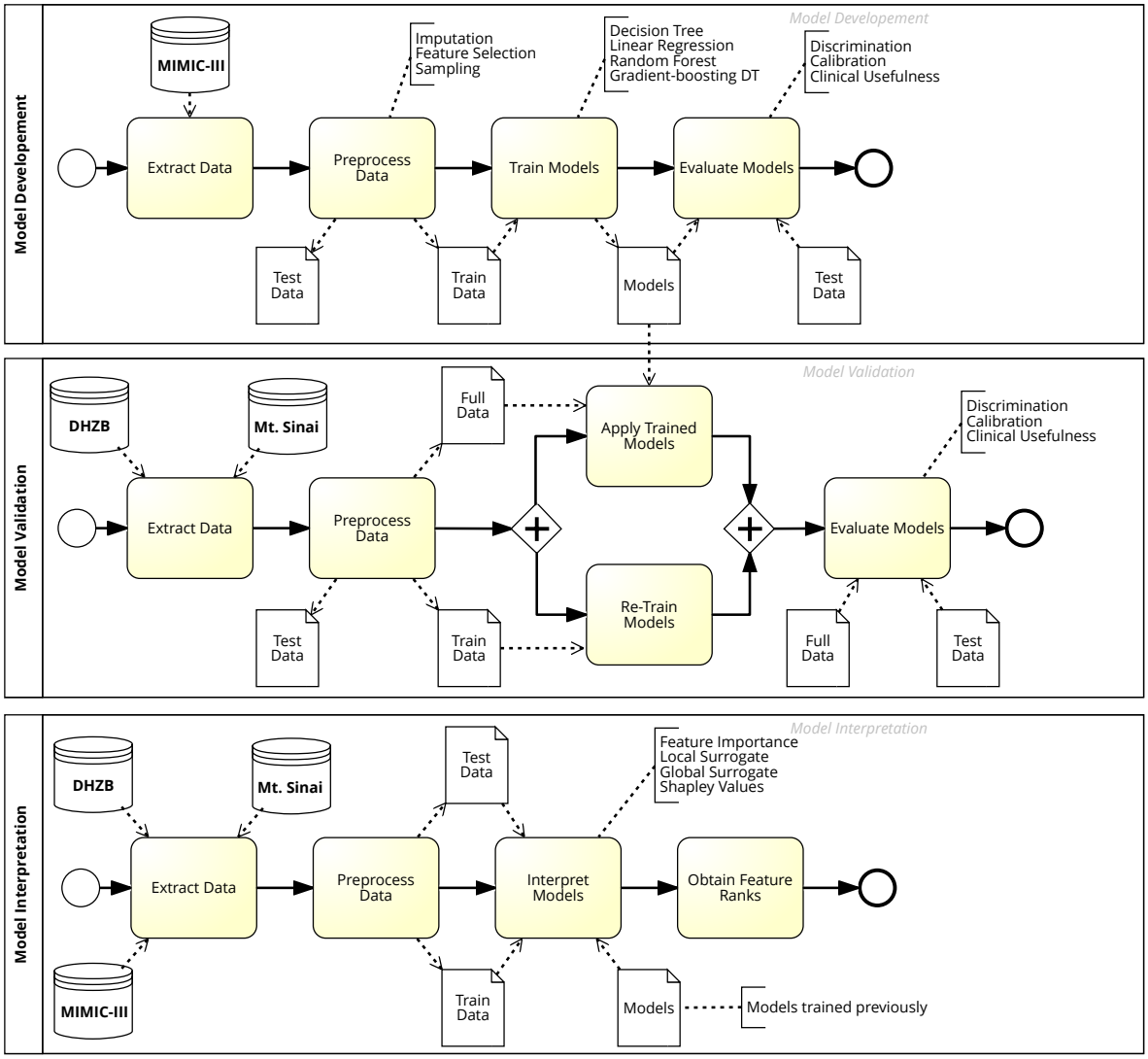


Figure 4.1.: Graphical abstract depicting the set-up of the experiments conducted in this chapter as a Business Process Modeling (BPMN) diagram. First models are developed on MIMIC-III database (derivation cohort) and evaluated on German Heart Center (DHZB) and Mount Sinai (validation cohorts). Model validation follows two set-ups: 1) applying trained models and 2) re-training models. Model interpretation is then performed on all cohorts, obtaining feature ranks.

Table 4.1.: Overview of CPMs for cardiac surgery-associated AKI. Abbreviations: CPM=Clinical Prediction Model; N=number of patients; AUC=Area Under the Curve, LR=Logistic Regression GAM=Generalized Additive Model, SL=Super Learner, GBDT=Gradient-Boosting Decision Tree, EL=Ensemble of Learners, RF=Random Forest, RNN=Recurrent Neural Network, NS=Not Specified

Type	CPM	*N*	Model	AUROC
Regression Models	Cleveland Score [130]	33,217	LR	0.81
	STS [132]	86,009		0.83
	AKICS [131]	603		0.84
	SRI Score [141]	2,566		0.78
	Ng et al. [125]	28,422		0.77
	Jiang et al. [142]	7,233		0.74
ML Models	Thottakkara et al. [134]	50,318	GAM	0.85
	Legrand et al. [135]	212	SL	0.76
	Eyck et al. [136]	810	NS	0.83
	Kate et al. [137]	25,521	EL	0.74
	Flechet et al. [138]	2,123	RF	0.84
	Lee et al. [139]	2,010	XGBoost	0.78
	Tomašev et al. [140]	703,782	RNN	0.83
	Our approach	**6,782**	**GBDT**	**0.90**

4.2.3. Model Interpretation

Despite the fact that black-box models may achieve better discriminative performance, clinicians tend to favor interpretable prediction models over opaque, ML models [144]. As such, in addition to applying new algorithms on the problem, I employ interpretability methods to lend intelligibility to the algorithms' predictions. In addition, the need for model interpretability is driven in part by laws and regulations applicable to automated algorithms which call for more transparency [145].

Doshi-Velez and Been define interpretability as "the ability to explain or to present in understandable terms to a human", which can be assessed globally, i.e., for the model as whole, or locally, i.e., for specific instances [55]. Taken together, these strategies rely on deriving surrogate models based on the original model to be explained. They are therefore termed global and local surrogates, respectively. Additionally, if one is dealing with a tree-based model, is it possible to derive rank of feature importance, which also provides some degree of transparency as to how the model works at the global level. Given that multiple interpretability methods exist, I follow Hall and Gill's recommendation, and combine both global and local interpretability methods, along with method-based feature importance and Shapley values [15]. To date, this is the first work comparing four different interpretability methods side-by-side applied to the validation of a CPM.

4.3. Methods

This section addresses the methods employed to address this use case. The overall approach is depicted in Figure 4.1.

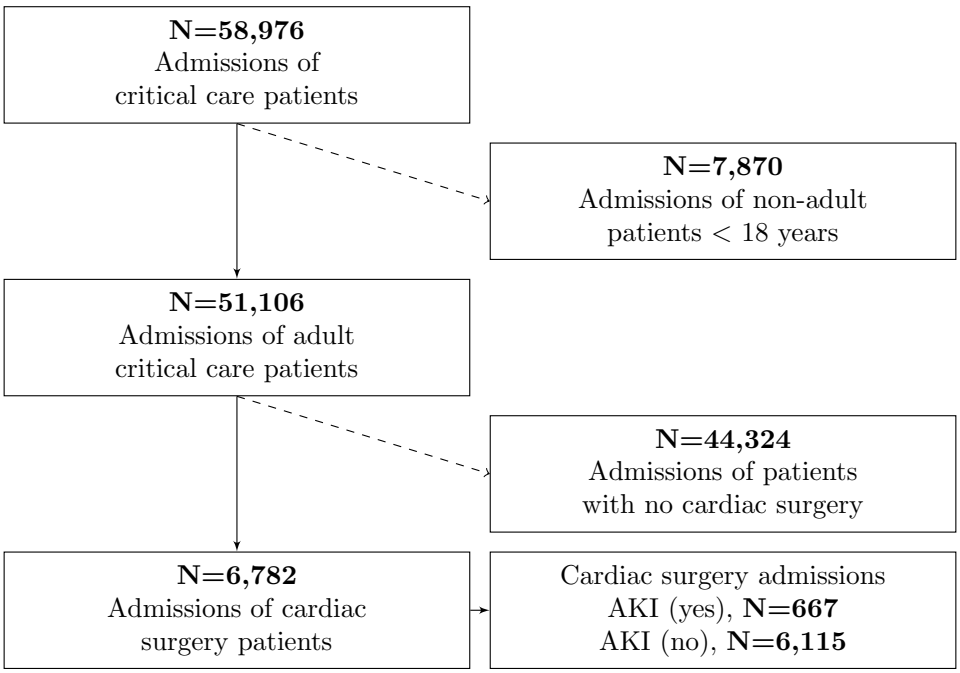


Figure 4.2.: Target AKI cohort obtained the MIMIC-III database. The data comprises 6,782 patient admissions, thus forming the basis for model development. Credit: extraction and diagram by F. Schneider[3].

4.3.1. Model Development

In the following, I describe the experimental setup for the development of the CPM addressed in this work. I provide implementation details, such as data used for training, preprocessing and prediction models used.

Experimental Set-up

I utilized a cohort of intensive care patients from MIMIC-III [127]. From this cohort, an initial feature set extracted was extracted based on expert consultation and analysis of literature. Following a number of preprocessing steps and data splitting to obtain training and validation datasets following the 80:20 ratio, I proceeded to train tree models, DT, GBDT, having LR as baseline. Subsequently, hyperparameter optimization was performed with gridsearch using 10-fold cross-validation as score. The models thus trained were then validated on a held-out dataset comprised of 20% of the original dataset. The 10-fold cross-validation discrimination and calibration metrics for each of the algorithms are compared side-by-side.

Derivation Cohort

The derivation cohort was extracted from the MIMIC-III database. Holding 53,423 hospital admissions, the database covers more than 10 years patient records of the Beth Israel Hospital in Boston [127].

The overall model training and validation data was comprised of 6,782 admissions in total. Inclusion criteria was the occurrence of a coronary artery bypass graft surgery or aortic valve repair/ replacement. The procedures were mapped to the respective Current Procedural Terminology (CPT) codes. Exclusion criterion was a minimum of 18 years of age, i.e., adult patients). Cohort building is illustrated in Figure 4.2.

With respect to the target outcome for model, AKI definition followed the Acute Kidney Injury Network (AKIN) classification occurring after the surgery. AKIN establishes three stages for AKI with increasing levels of renal failure. Determination of renal function for AKIN relies on the concentration of serum creatinine [146]. In the extracted MIMIC-III cohort, the incidence of AKI was approximately 10%, agreeing with general clinical observations [18].

It must be noted that there is a significant imbalance in the derivation cohort with respect to the number of AKI cases. The cohort has significantly more patients not being affected by AKI as opposed to those that do present it. As such, model development must account for this class imbalance in positive and negative cases (cf. Figure 4.2).

Model Features

The initial model features from MIMIC-III were derived from consultations with medical experts and analysis of extant literature as laid out in Section 4.2. Model features encompass demographics, i.e., sex and age, comorbidities, and laboratory markers obtained from blood samples.

The available comorbidities were used to calculate a composite comorbidity index, which provides a summary of the patient's overall health status. Specifically, this work relies on the Elixhauser comorbidity score, which correlates with disease severity [147]. The score is made up of 30 comorbidities in total, including, e.g., hypertension, obesity, weight loss, among others.

To capture the effects of longitudinal health variables, the model also includes laboratory tests for a range of biomarkers. Some of these biomarkers are known to be associated with AKI and renal function, such as creatinine. The laboratory tests were then averaged on three consectutive days, respectively, before the surgery.

Since each laboraty test generates three features (one for each day, up to three), plus comorbidities and demographics, the model relied on overall 103 features, which were then submitted to preprocessing and feature selection.

Feature Preprocessing

Prior to modeling, the features extracted need to be adequately preprocessed. In this work, three main processing steps apply. First, missing values were imputed, since ML algorithms such as GBDT often require complete datasets.

To handle missing values, I applied the k-Nearest Neighbors (kNN) imputation method with k=3, which operates under the assumption that missing values can be approximated by samples that are most similar to it. The framework implemented in MORPHER is the `fancyimpute` Python library [88].

It must be noted that the missingness is more prevalent for laboratory values two and three days before surgery. At the same time, missingness is lowest for routinely collected demographic data. An overview of each of the features used in the model, long with the prevalence of missing data is provided in Table A.1.

Feature Selection

In this processing step, I performed tests using different percentiles of top features, using the full set of features (103), top 50% and top 25%, reporting the results for all the algorithms tested. I chose feature selection based on the mutual information approach, since it can capture non-linear dependencies among variables, unlike an F-test, which can capture only linear correlations [148]. The effects of feature selection on model performance is handled at greater length in Chapter 5.

Modeling Algorithms

In the following, I outline the modeling algorithms applied, alongside the respective hyperparameters' configuration and optimization strategy.

Logistic Regression. Usual parameters to adjust for LR include regularization strength and the type of penalty, L1 or L2. Regularization can improve model performance for unseen data by penalizing large coefficients in an effort to reduce overfitting or learning training data 'peculiarities'. Higher values for λ, or regularization strength, can lead to more sparse models. The library utilized exposes the parameter C defined as the inverse of regularization strength. The regularization parameter was set to 1.0 and iterative convergence parameter was set to 1000.

Decision Tree. I applied the Gini impurity measure to calculate optimal splits. Furthermore, to correct for the class imbalance observed, class weights were calculated according the prevalence of each class (negative or positive) in the dataset, leading to balanced class weights. By means of hyperparameter grid search with five-fold cross-validation, a maximum tree-depth of five was determined.

Gradient-boosted Decision Trees.

This modeling algorithms makes use of an ensemble of small decision trees, i.e., three with a small depth [32]. By means of five-fold cross validation, an optimal tree depth of three was determined. As such, in addition to those pertaining to decision trees, hyperparameters related to the ensemble itself can be tuned, such as the number of composite trees (ensemble size or number of estimators). The number of estimators applied was 150. Finally, the contribution of each individual estimator can be regularized by means of a learning rate. The optimal value for this parameter was set to 0.1 after grid search.

Random Forest. Similarly to GBDT this algorithms belongs to the class of ensemble learners. It builds different decision trees using random subsets of features. Since prediction is decided by means of a voting procedure, RF tend to overfit less in comparison to single decision trees. Hyperparameter optimization led to a number of 300 individual estimators, i.e., trees, each containing up to 16 in tree depth with balanced class weight.

Model Performance

In order to evaluate performance of the models, I relied on discrimination, calibration and clinical usefulness.

Discrimination. The first aspect to be assessed in a given classifier is the extent to which it can correctly discriminate between the occurance or absence of the target variable, i.e., its discriminative performance. To measure discrimination, I utilized both AUROC and Diagnostic Odds Ratio (DOR), a metric commonly applied in medicine [38], alongside precision and recall.

Calibration. While discrimination seeks to assess a classifier's ability to distinguish between presence or absence of patient outcome (in the binary case), calibration defines to that extent the model overestimates or underestimates patient risk in different risk deciles. In other words, a calibration analysis helps to assess how well the predicted probability actually can be observed in real life. Since the real probability is not known, which is the reason to conduct predictive modeling in the first place, one can define fraction of positives within deciles and compare that to the mean predicted probability of the model for each decile. A thus generated plot, i.e., a calibration plot, enables a graphical assessment of calibration [42].

Clinical Usefulness. Whereas discrimination and calibration shed light on a classifier's performance when it comes to discriminating between classes or correctly predicting risk. However, none is concerned with whether a practitioner should actually use such a model in clinical practice to guide treatment decisions, especially given that there is often a trade-off between benefits and harms. A

threshold is usually defined beyond which action would be advisable and based upon which sensitivity and specificity can be calculated. This relies on the assumption that the relative benefits of a true positive are comparable to the harms of a false positive, a naive assumption that is implausible in practice [4].

4.3.2. Model Validation

After model development, I sought to validate the CPM by investigating its performance when used in two different cohorts, German Heart Center Berlin (DHZB) and Mount Sinai Hospital. In the following, we elucidate the experimental set-up employed followed by a description of the validation cohorts.

Experimental Set-up

I utilized two experiment set-ups. The first consisted in running the original model without any modification, in order to ascertain its generalizability. The second set-up consisted in updating the original model. In this work, model update consisted in re-training the original classifiers exclusively on the validation dataset – therefore not including the derivation cohort – while also optimizing the respective algorithms' hyperparameters using gridsearch with 5-fold cross-validation. The train/test split chosen was 80:20. As such, the metrics reported refer to the performance on the held-out test set.

Validation Cohorts

The two cohorts, DHZB and Mount Sinai Hospital, will be described in further detail below.

German Heart Center Berlin (DHZB) Data for external validation was drawn from 54,958 admissions in the period of 2013-2018 at the German Heart Center Berlin (DHZB), a hospital specialized in the care of cardiac patients. Exclusion criteria entailed admissions of non-adult patients (5,853) and those that had no surgery or only minor surgery (31,635), with a final cohort of N=14,191 admissions. In this cohort, AKI incidence was approximately 38.4% (5,449 out of 14,191) and therefore higher than usually reported in the literature [149], also differing from the AKI incidence in the derivation cohort (9.83%).

Mount Sinai Hospital Additionally, I extracted a cohort from an American medical institution, the Mount Sinai Hospital in New York. The Mount Sinai Health System is a large and diverse healthcare provider, located in New York, NY, which produces a high volume of structured, semi-structured and unstructured in-patient, out-patient and emergency visits data. In this cohort, patients were included in the cohort who underwent bypass or valve replacement surgery, while excluding non-adult subjects. It comprised a total of 25,799 admissions presenting an AKI incidence of 4,51%, commensurate with rates reported in the literature [149], while being lower than AKI incidence in the derivation cohort (9.83%).

A complete listing of the features used in the models, comparing the derivation cohort and both validation cohorts side-by-side can be examined in the Appendix (cf. Table A.1).

4.3.3. Model Interpretation

In addition to development and validating the CPM, I apply four interpretability methods, global and local surrogate, along with method-based feature importance and Shapley values, to help shed light on how the ML model works and possibly inform future model updating.

Global Surrogate

To provide more transparency on the ML models employed, I utilized the approach mimic learning [58]. Based on the approach by Che et al., an interpretable model, i.e., the mimic model, was trained on the predicted probabilities of the ensemble models, i.e., RF and GBDT [58]. In this approach, the mimic model is trained on the same features used for the more complex model. In this work, I utilized Bayesian Ridge Regression (BRR) as mimic model. Similarly to linear regression, which is widely employed in the medical field, BRR yields coefficients for the respective input features, being therefore more amenable to interpretation. More importantly, it includes regularization parameters learned directly from the data as opposed to set manually, which is necessary, e.g., in the case of λ for l_2 regularization in Ridge regression. As such, this model has the additional benefit of automatically shrinking the regression coefficients [150].

Local Surrogate

As local surrogate, I use the interpretability method LIME to shed light on the prediction results of the GBDT model. LIME uses more comprehensible models, e.g., linear regression, to approximate the behavior of a given model in the vicinity of the instance/prediction being explained. The algorithm generates a number of perturbed instances close to the instance of interest, weighing this perturbed input according to a distance measure. After applying the original model on these perturbed instances, a linear function is applied to approximate the thus resulting outputs [52]. The coefficients of this linear function represent the degree of influence of a given feature for the original prediction we intended to explain. The higher the number of these perturbed instances or samples, the higher the fidelity of the approximate model, but the higher the algorithm run-time. In this work, I used a sample size of 25% of the total cohort size.

Method-based Feature Importance

Besides the aforementioned global and local surrogates, I provided feature importance metrics for selected algorithms. In tree-based methods such as RF, one can estimate the relative feature importance by computing the decrease in node impurity when using the given feature as split criterion. This decrease is averaged across all constituent trees and weighted proportionally to the number of samples it splits, i.e., nodes closer to the root of the tree will be deemed more important [59].

Shapley Values

This interpretability method uses concepts from game theory to calculate feature rankings. Succinctly, this method considers different 'coalitions' of features and averages the marginal contribution of the given feature when it is added to the respective coalitions of features. Specifically, in this work, I utilized SHapley Additive exPlanations (SHAP), a computationally efficient alternative to estimating Shapley values, particularly in the case of tree-based methods [62].

Interpretation Heatmap

As we shall exemplify, the results of applying different interpretability methods differ from one another, even though some features seem to repeat often across methods. This is a known issue with such methods: since the underlying strategies for computing the feature importances differ, it is hard to rely on any single method as representative of the model's underpinnings. In other words, robustness or consistency across methods cannot be guaranteed. As such, researchers recommend making use of different interpretability methods when seeking to interpret a given model [62].

In order to provide a summary view on different interpretability methods, I devised a heatmap-like representation of feature importances, which offers a ready-to-inspect perspective on feature importance across methods. First, the actual feature importance values differ in their numeric meaning. For example, in method-based feature importance the numeric values represent mean decrease in Gini impurity. In mimic learning, the feature importance values reflect the regression coefficients of the surrogate logistic regression model. As such, to allow comparison across methods, the feature importances must be normalized. I chose to use min-max normalization between 0.1-1 to avoid any given feature being assigned zero importance (a very low feature importance is still more informative than zero).

For plotting, the mean feature importance across all methods for a given feature is computed. Then, I retrieve the K top feature with the highest mean feature importance. When compared side by side, the outputs of the different interpretability methods provide some insight into the relevance of model features. The heatmap generated by coloring features and methods according to the intensity of the given feature's importance after normalization, thereby allowing comparison between the different methods.

4.4. Results

This section outlines the results obtained on discrimination, calibration and clinical usefulness, while interpretation is targeted at feature importance for each of the cohorts.

4.4.1. Discrimination

In this section, I present the performance results achieved using the proposed ML methods. Performance analysis entails AUROC, precision recall and DOR, a metric often utilized in the clinical context [38].

Derivation Cohort

Table 4.2 reports the selected metrics across all feature selection configurations and models tested, considering respectively all features, top 50% and top 25% percentiles. DT performed worse than LR and GBDT for most metrics, regardless of feature selection, except for recall, where it presented a substantial advantage against the other two approaches, e.g. recall of 0.66 as opposed to GBDT's 0.48 for the top 50% features.

The different configurations chosen for feature selection demonstrated that the models achieve a similar performance even when only subsets of the available features are used. Particularly when it comes to the ensemble models, GBDT and RF, the DOR was substantially improved by removing features, e.g., from 90.74 to 149.92 for GBDT. As more features are removed, though, performance begins to deteriorate perceptibly, e.g., a drop of approximately 3% in AUROC when only 25% of the features are used in the GBDT. However, RF remained constant at around 0.90 AUROC across all feature selection thresholds.

In summary, the GBDT and RF ensemble classifiers achieved better discriminative performance in comparison to decision trees or logistic regression. In the case of GBDT, this is most notable when it comes to precision (40% increase over LR) and AUROC (6% increase over LR), even though it performs poorly when it comes to recall. Furthermore, GBDT and RF presented substantially better results with respect to DOR with a 7-fold increase when compared to LR with 50% of features (GBDT).

Table 4.2.: Precision, recall, diagnostic odds ratio (DOR), and area under receiver operating curve (AUROC) for AKI=yes achieved with the proposed approach employing logistic regression (LR), decision tree classification (DT) and gradient-boosted decision trees (GBDT) respectively for different feature selection configurations (all features, top 50% and 25%). The results were obtained by applying the trained models on a hold-out validation dataset made up of 20% of the original dataset.

Metrics	MIMIC-III											
	Precision			Recall			DOR			AUROC		
	All	50%	25%	All	50%	25%	All	50%	25%	All	50%	25%
LR	0.63	0.63	0.59	0.28	0.25	0.25	19.55	19.14	16.67	0.84	0.84	0.82
DT	0.33	0.35	0.29	**0.67**	**0.66**	**0.70**	10.86	11.22	10.16	0.80	0.80	0.78
GBDT	0.86	0.90	0.62	0.43	0.48	0.32	90.74	149.92	**115.50**	0.89	**0.90**	0.87
RF	**0.90**	**0.92**	**0.86**	0.33	0.41	0.42	**120.8**	**169.58**	87.54	**0.90**	**0.90**	**0.90**

Table 4.3.: Precision, Recall, Diagnostic Odds Ratio (DOR), and Area Under Receiver Operating Curve (AUROC) for AKI=yes achieved in the validation cohort, i.e., German Heart Center, with model 'as is' (applying derivation model) and after re-training the model, employing Logistic Regression (LR), Decision Tree (DT), Gradient-Boosted Decision Trees (GBDT), and Random Forest (RF).

Metrics	German Heart Center							
	Applying Derivation Model				Re-Training Model			
	Precision	Recall	DOR	AUROC	Precision	Recall	DOR	AUROC
LR	0.00	0.00	n/a	0.56	0.68	0.33	4.27	0.69
DT	0.68	0.16	3.78	0.52	**0.75**	0.30	6.08	0.71
GBDT	0.58	**0.22**	2.54	0.62	0.72	**0.42**	5.94	0.75
RF	**0.90**	0.02	**14.44**	**0.70**	**0.75**	**0.42**	**6.90**	**0.76**

Validation Cohorts

With regards to performance in the validation cohorts I analyzed two variants, one considering the derivation model "as-is" and another by training the algorithms with the local dataset, i.e., model updating. As expected, a sharp deterioration in most metrics can be observed when applying the models without any changes on the validation cohorts.

German Heart Center. Validation results are provided in Table 4.3. The AUROC of the ensemble methods was reduced in 30%, e.g., with respect to GBDT in comparison to the performance of the derivation model. Exception to this is the precision from the RF classifier, which remained similar to that of the original model (≈0.9). Performance was deteriorated to such a sharp degree that some of the metrics were set to zero, because not mathematically definable, i.e., division by zero. After model update, including hyperparameter tuning with gridsearch, a modest increase in performance could be achieved with AUROC=0.76 for RF. Nevertheless, the updated model still performed significantly worse than the original model (derivation cohort). Most strikingly, a difference of about 12% in AUROC was accompanied by 20 times lower DOR, i.e., 169.58 vs. 6.90 for RF.

Mount Sinai. An overview of the validation results can be found in Table 4.4. For this cohort, in order to analyze the range of results, I also included the confidence interval in the results reported. Similarly to the DHZB cohort, the AUROC of RF and GBDT were reduced in 17% and approximately 50%, respectively. Re-training the model, lead to an increase in performance, achieving AUROC=0.84

with RF. Furthermore, as in the DZHB cohort, a slight difference of about 4% in AUROC was accompanied by an almost 50% lower DOR, i.e., 22.199 vs. 11.883 for RF. This points towards the necessity of possibly adjusting the discrimination thresholds in the validation cohorts.

Overall, the ensemble algorithms used, GBDT and RF performed best in derivation and validation alike, with the exception of LR placed as a close second when it comes to recall. Notwithstanding these results, performance on validation was significantly worse, even more pronouncedly for DZHB, with marked differences observed specially with regards to DOR. From this point on, for the sake of brevity, I turn the focus to the ensemble models, which have performed best, for the analysis of calibration, clinical usefulness and interpretability.

Table 4.4.: Precision, Recall, Diagnostic Odds Ratio (DOR), and Area Under Receiver Operating Curve (AUROC) for AKI=yes achieved in the validation cohort, i.e., Mount Sinai, with model 'as is' and after full model update, employing Logistic Regression (LR), Decision Tree (DT), Gradient-Boosted Decision Trees (GBDT), and Random Forest (RF). Metrics are reported alongside 95% confidence interval calculated using 5,000 bootstrapped samples, i.e., with replacement (applying derivation model) and via 10-fold cross validation (re-training model).

Metrics	**Mount Sinai**							
	Applying Model				Re-Training Model			
	Precision	Recall	DOR	AUROC	Precision	Recall	DOR	AUROC
LR	0.07 ± 0.00	0.64 ± 0.02	2.46 ± 0.21	0.64 ± 0.01	0.11 ± 0.01	**0.73 ± 0.06**	7.45 ± 1.86	0.80 ± 0.02
DT	0.14 ± 0.02	0.11 ± 0.01	3.86 ± 0.61	0.64 ± 0.01	0.18 ± 0.04	0.49 ± 0.10	6.04 ± 3.09	0.73 ± 0.09
GBDT	0.04 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.43 ± 0.01	0.32 ± 0.16	0.20 ± 0.22	**15.16 ± 13.76**	0.82 ± 0.03
RF	**0.25 ± 0.09**	0.01 ± 0.00	**7.75 ± 4.20**	**0.73 ± 0.01**	0.30 ± 0.041	0.23 ± 0.01	11.88 ± 2.44	**0.84 ± 0.01**

4.4.2. Calibration

In this section, I turn to the calibration metrics of the proposed models, in order to investigate the degree to which the predicted probabilities agree with the observed fraction of positives within the cohorts. The basis for these results is depicted in the calibration plots in Figure 4.3.

Derivation Cohort

In Figure 4.3, one can observe that for patients with low to moderate observed risk (up to 0.7) both RF and GBDT systematically overestimate risk, though to a smaller extent for the latter, as evidenced by the lower Brier score. For patients deemed to be at higher risk range, e.g., > 0.8, the RF classifier provides a better risk assessment, even in comparison to GBDT. In both cases, applying Platt's method (sigmoid calibration) did not substantially improve calibration.

Validation Cohorts

A similar set of observations can be made for the Mount Sinai cohort with respect to RF, with the model overpredicting risk for low-risk patients. With respect to RF, however, the model presented better calibration for risk level below 0.7 (fraction of positives). After using Platt's method, model calibration could be somewhat improved for RF in both the derivation and Mount Sinai cohort, achieving lower Brier scores. An exception to both the derivation cohort and the Mount Sinai cohort is the DHZB cohort. The model predictions are mostly well calibrated, with substantially smaller Brier score. Yet, minor miscalibrations can be observed for RF as observed risk increases, i.e., from 0.5 fraction of positives on, with the minor underpredicting risk. A similar trend (underprediction) risk can be ascertained for GBDT in the ranges of 0.5-0.7.

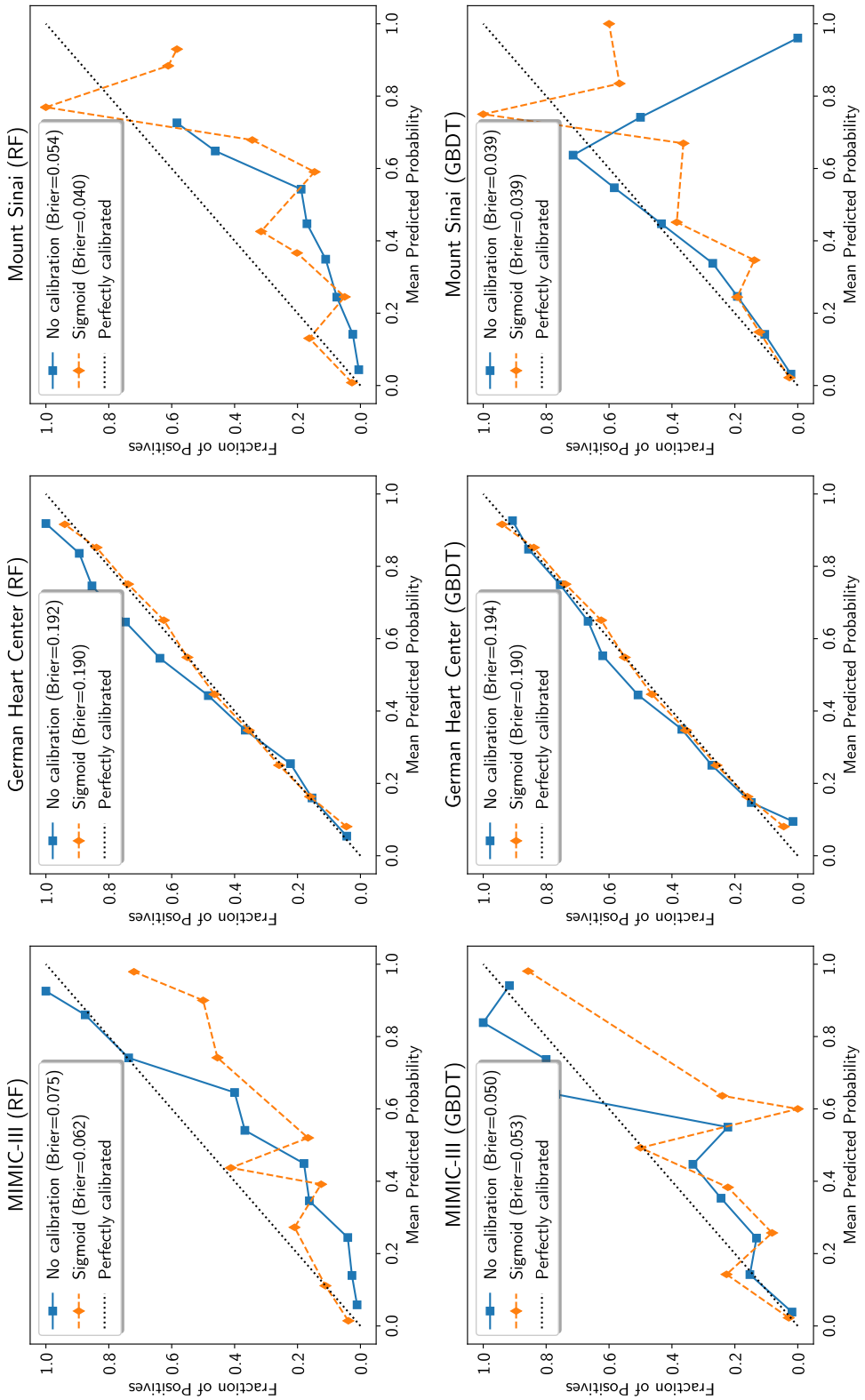


Figure 4.3.: Calibration plots for both the gradient boosting classifier (top row) and random forest classifier (lower row) for the derivation (column one) and validation cohorts (columns two and three). Graphs show model output without calibration and after applying Platt's method along with Brier scores.

4.4.3. Clinical Usefulness

We now turn our gaze towards clinical usefulness. By means of the net benefit curve, I seek to establish a trade-off between risk thresholds where use of the model predictions would be advisable in comparison to default standards of care (treat all or treat none). The decision curves for all the cohort and both RF and GBDT algorithms can be examined in Figure 4.4.

Derivation Cohort

Figure 4.4 displays the net benefit curve for all the algorithms under analysis for all three cohorts, using the re-trained models. The decision curve helps to visualize the trade-off between false positives and false negatives over different thresholds. Given a predicted probability by a model, the thresholds represent the cut-off point for determining a positive or negative outcome.

If the model curve lies above the curves of the default strategies (treat all or treat none), one can consider that the model offers a *net benefit* over these strategies in the threshold intervals considered. As such, Figure 4.4 suggests that GBDT provides a net benefit over all risk thresholds, except in the high risk ranges (> 0.8). In contrast, using the RF model exhibits low or zero net benefit in the low-to-moderate risk ranges. However, it seems to be better than the default alternative of not treating all the patients for a threshold > 0.8.

Validation Cohorts

In the case of the DHZB cohort, the distinction between the different models cannot be readily distinguished, with a roughly equivalent Area under the Net Benefit Curve (ANBC). Furthermore, using any of the models seems to provide a net benefit over the default strategies for all risk thresholds, except > 0.8, where the benefits zero out for both models. With respect to the Mount Sinai cohort, not only there is a marked difference between the two models, but the RF model presents negative benefit – or rather harm – for most thresholds (roughly between 0.12 and 0.8). Even the GBDT only provides moderate benefit in lower thresholds (< 0.25), with ANBC=0.

Overall, it can be ascertained that for certain threshold ranges the models provide a net benefit in both the derivation and DHZB cohort, but the net benefit is uncertain or questionable for the Mount Sinai cohort. However, these benefits can only be observed in specific ranges of thresholds, also depending on the algorithm under analysis.

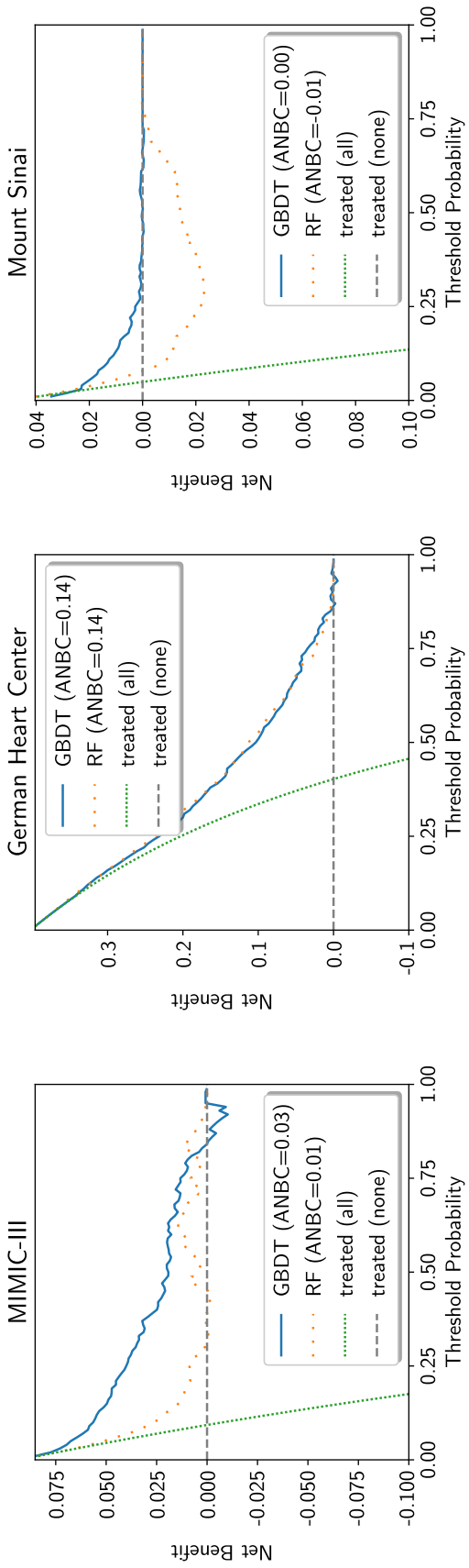


Figure 4.4.: Decision curves for both the derivation and validation cohorts and the different algorithms. ANBC=Area Under the Net Benefit Curve.

4.4.4. Model Interpretability

Using the interpretability approaches discussed, I seek to shed light into the 'black-box' by providing the feature importance results for the derivation cohort and both validation cohorts.

Derivation Cohort

For the interpretation of the models in the derivation cohort, I chose one of the ensemble models (GBDT) because of its clinical usefulness performance and examine it with the selected interpretation methods. First at the global level, with mimic learning and method-based feature importance, followed by LIME and SHAP. Method-based feature importance is derived directly by the model, e.g., using a metric such mean decrease in Gini impurity [59]. For the sake of brevity, in the derivation cohort I provide a detailed output of all the methods, while opting for a summarized perspective in the validation cohorts using the interpretability heatmap.

Mimic learning and method-based. Regarding both mimic learning and method-based interpretability approaches, Figure 4.5 shows the respective feature importances. Blood urea nitrogen (BUN) in the days prior to the surgical procedure featured turned out to be an important feature. In fact, this is an important biomarker associated with kidney function, along with creatinine levels. However, creatinine did not figure in the mimic learning approach, but only in the method-based approach. Notably, both methods also included Elixhauser score, an overall indicator of co-morbidity, as an important feature. These interpretability approaches offer an overall perspective on the model, but little information on the direction of that correlation, e.g., are low or high values of serum creatinine associated with a positive outcome? This question is tackled by methods such as LIME and SHAP.

Local Surrogate. The LIME method expects a given prediction sample – or a patient – along with the trained model an inputs. Therefore, an expert can inquire the model as to 'why' a given decision was made by the algorithm. However, to obtain an understanding of the model as whole, one would have to explain many instances. Since, this task might be too consuming depending on dataset size, LIME provides a strategy called submodular pick that retrieves the instances that are the most representative of the overall model's behavior [52]. The results provided by LIME represent the coefficients of the surrogate regression model applied locally. I report the top ten features that explain the onset of AKI using submodular pick of six prediction samples in the test set, as displayed in Figure 4.6. Some of features tend to be important for the model's output across different prediction instances, e.g., Elixhauser score, platelet count, serum creatinine, and BUN lab tests. Strikingly, blood-related biomarkers, such as high hemoglobin, feature in the local explanations as *protective* factors.

Shapley Values. Finally, the SHAP summary plot conveys both a global perspective, as well as local perspective using a game-theoretical approach [62]. To a certain extent, it merges the approaches discussed previously. Indeed, we observe some of features mentioned in the other methods once again highlighted by SHAP: BUN, serum creatinine and hematologic values. Indeed, the trend observed for these blood-related values via LIME reappear in SHAP, i.e., high-values of hemoglobin, hematocrit levels, seem to lead to lower likelihood of AKI.

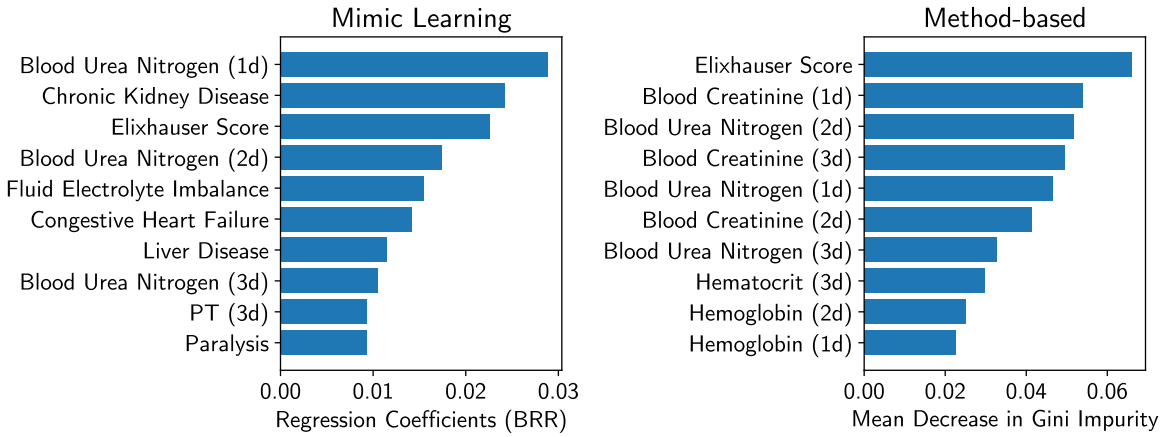


Figure 4.5.: Global explanations provided by mimic learning and method-based feature importance using mean Gini impurity for the 10 most important features. Note that some of the features are shared among both methods, such as blood urea nitrogen. Abbreviation: PT=Prothrombin Time.

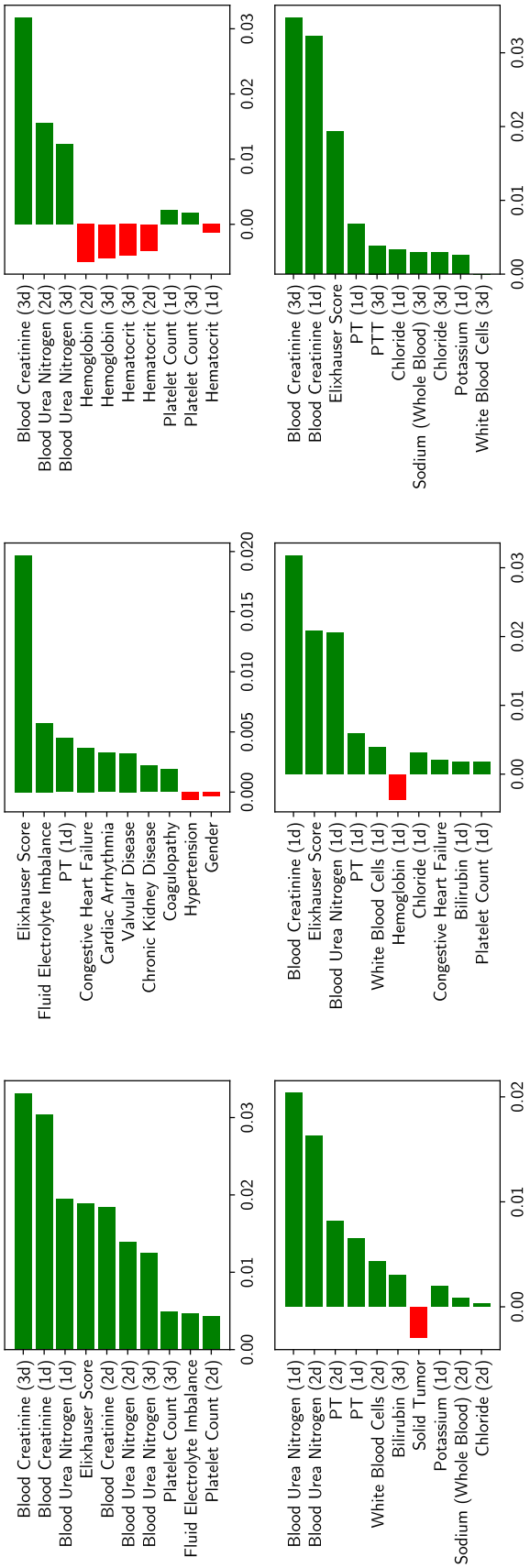


Figure 4.6.: Local explanations provided by LIME. Using submodular pick, LIME chooses the most significant examples for explanation, i.e, the individual subplots. Each shows the top 10 features chosen by the method as the most meaningful for the local predictions. Note that positive coefficients are correlated with increased likelihood of the outcome (Acute Kidney Injury). Abbreviations: PT=Prothrombin Time, PTT=partial Thromboplastin Time.

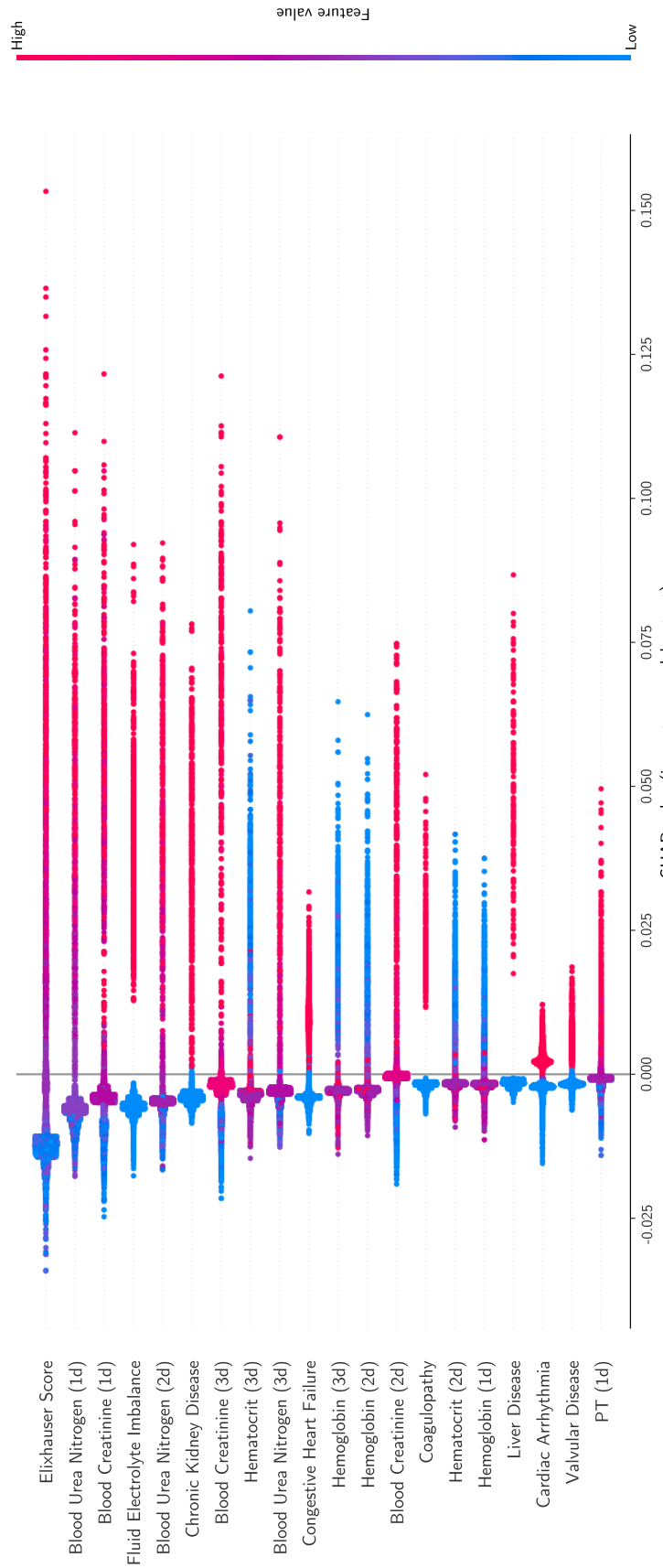


Figure 4.7.: Summary plot provided by SHAP [62]. Features are sorted by importance. Dots represent individual data points and how different features values (high and low) influence model outcome. Colors refer to the model output, red for positive and blue for negative.

Validation Cohorts

The heatmap in Figure 4.8 shows that blood urea and serum creatinine shortly before the procedure, both clinically recognized biomarkers of kidney function, as well as Elixhauser score, a measure of combined comorbidity, were considered important for all three methods in both cohorts. In contrast, pre-existing chronic kidney disease featured prominently in the Mount Sinai cohort, while not the in DHZB cohort.

It is important to notice that the magnitude of the importance of a given feature varies across methods. For example, gender is assigned a relatively low feature importance across cohorts and methods. However, it seemed comparatively more important for the local surrogate in the Mount Sinai cohort. A similar pattern emerges with respect to peripheral vascular disease: important for the global surrogate in the DHZB and for the local surrogate in the Mount Sinai cohort, but was assigned a substantially lower importance otherwise.

Also noteworthy is the fact that markers associated with cardiac disease, such as valvular disease, peripheral vascular disease, congestive heart failure and cardiac arrhythmia seemed to play an important role for the models predictions across the cohorts. Exception here is cardiac arrhythmia: featured importantly only on the Mount Sinai cohort.

4.5. Discussion

In the following, I analyze the results with respect to model development, validation and interpretation and their significance in the context of previous work.

4.5.1. Model Development

Being able to more accurately predict AKI cases after surgery can empower doctors to adopt targeted kidney-protective measures ahead of time.

In the extant literature, ensemble models have achieved excellent results for a wide variety of prediction tasks [42]. The high discriminative performance achieved in this work by GBDT and RF reflects the expectations of better performance with respect to linear models.

Furthermore, as exemplified by Table 4.1, the model developed in this work outperforms the Cleveland score by a considerable margin. However, the Cleveland score's authors utilize a substantially larger and more diverse cohort. The same observation applies to the STS score. The work of Tomašev et al. (Google DeepMind) takes it a step further with a massive cohort of more than 700,000 patients [140]. One could reasonably argue that those scores potentially present higher generalizability, but this is not always the case, as exemplified by applying the Cleveland score on a cohort of Chinese patients [142].

Even though the model developed performed well across most metrics, it showed significant drawbacks with regards to recall. While this issue can possibly be mitigated by adjusting classification thresholds, it might have critical implications for clinical practice. Since a lower recall means that patients who will develop AKI might incorrectly be deemed as not under risk, calibration must be conducted, at the expense of possibly harming patients. This fact speaks for the necessity of a holistic evaluation of discrimination metrics.

Finally, the models developed included laboratory values prior to the surgery. Arguably, these are not always available for surgery patients, particularly when it comes to emergency surgeries. This fact has led to a high degree of missing values in the cohorts. Even though imputation has been performed, it is not possible to guarantee that the model has not been biased in some way or another.

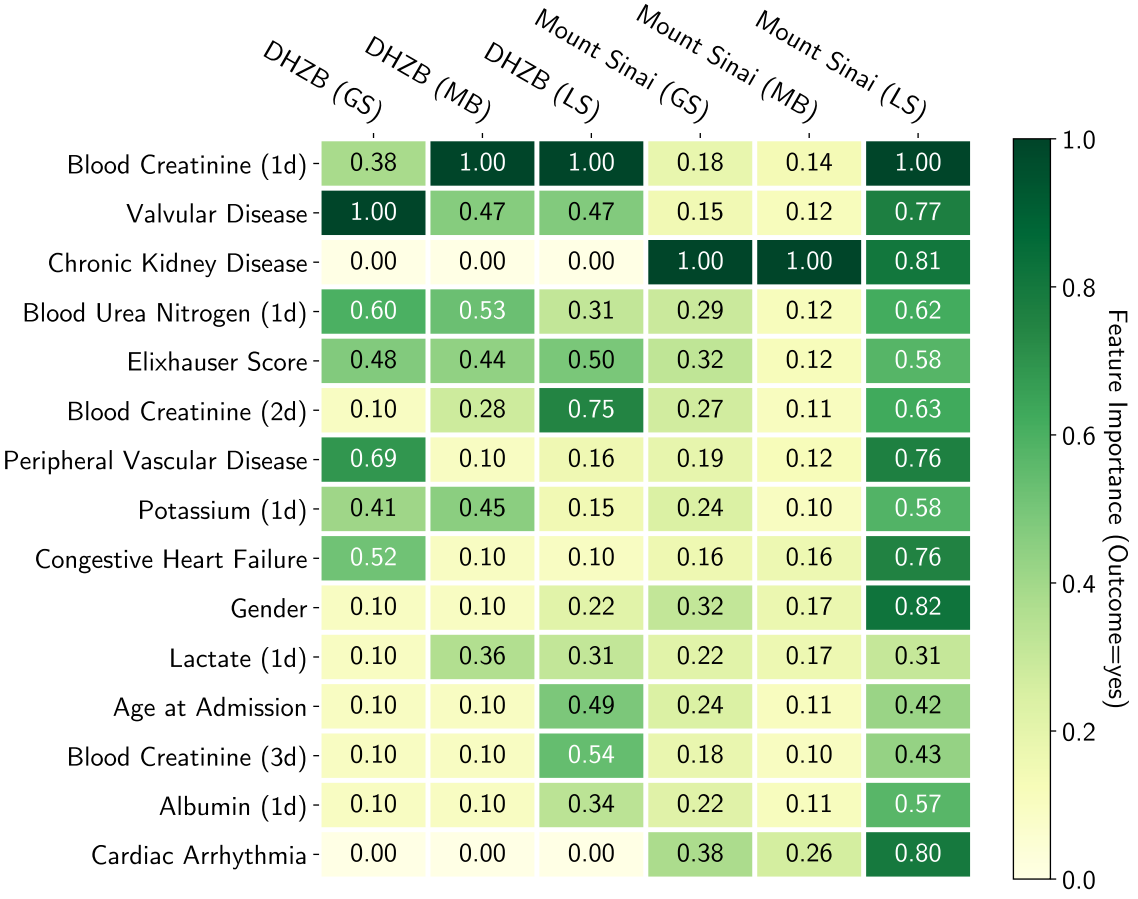


Figure 4.8.: Heatmap displaying normalized feature contributions from mimic learning, GBDT feature importance and LIME. Features are sorted according to the average contribution of all methods taken together. Abbreviations: GS=Global Surrogate, LS=Local Surrogate, MB=Method-based feature importance.

4.5.2. Model Validation

In the derivation cohort, while a high AUROC could be achieved, an imbalance with respect to precision and recall could be observed, which was also present in the validation cohorts. In this work's context, higher precision than recall means that the model is particularly sure that patients it classifies as under risk of AKI do in fact develop it, though it was very selective when doing so. As a result, it means that the model will likely 'miss out' on patients under risk (false negatives). This tendency of the model was further exacerbated in the validation cohorts as illustrated by the large difference in terms of DOR. While precision and recall differed upon validation, they did so to a considerable smaller extent when compared to DOR for both cohorts.

This difference in DOR can be explained by a much larger ratio of false negatives to true negatives in the validation cohort (cf. Equation 2.8). For AKI management, the costs of misclassifying a high-risk patient as not under risk are higher than enabling protective measures for patients who are not under risk. Therefore, thorough model calibration would be necessary before clinical deployment with other techniques beyond Platt's method. The differences in performance upon validation, particularly in the set-up *without* model update, can be traced back in part to a difference in the prevalence of the outcome of interest, i.e., 9.83% (derivation), 38.4% (DHZB) and 4.51% (Mount Sinai).

Overall, the deterioration in the discriminative performance of the derivation model (MIMIC-III) was more pronounced in the DHZB than the Mount Sinai cohort. This can be partially explained by the fact that MIMIC-III is based on data from Beth Israel Deaconess Hospital, also an American institution, with potentially similar standards of care with respect to Mount Sinai hospital. This difference highlights the necessity of examining the geographical stability of prediction models [36].

Finally, clinical usefulness analysis suggests that the model as developed provides benefits only within given threshold ranges: this fact must necessarily be taken into account in a future model deployment, i.e., a blanket deployment might lead to deleterious effects in spite of satisfactory discriminative performance. These results underscores the need for comprehensive model reporting, including discrimination, calibration and clinical usefulness, particularly when it comes to machine learning models, at the risk of potentially compromising patients' health [36].

4.5.3. Model Interpretation

Using the interpretability approaches presented, I seek to shed light into the black-box models in order to 1) help to uncover potentially new clinically relevant results, 2) identify potential model biases and 3) obtain information on how the models could be updated in the validation cohorts. In the following, I provide the feature importance results for the derivation cohort, and both validation cohorts.

Potentially Clinically Relevant Insights

Model interpretability is critically relevant for medical practitioners to foster acceptance and increase trust in decision support for clinical applications [53].

Upon examination of the instances chosen by the submodular pick by LIME, we can observe that a high Elixhauser score tends to increase the risk of post-surgery AKI. Note that positive coefficients are positively correlated with the outcome and vice-versa. This observation agrees with the medical interpretation of this comordibity score: higher values are in general associated with poorer patient outcomes in general [151]. With regards to blood/serum creatinine, it is an important marker of kidney function, being present in the definition of AKI itself, with higher values indicating deterioration of kidney function [146].

Furthermore, a striking result observed in the derivation cohort is the influence of blood-related biomarkers, such as hemoglobin, which were captured in by the different methods. In particular, LIME indicate these biomarkers as being protective, i.e., lowering the likelihood of AKI. The direction of this

association was made more clear by SHAP: low values of hemoglobin and hematocrit increased the likelihood of outcome (AKI), while high values seemed to have a protective effect. A known association exists between anemia (low red cell count) and CKD, since the kidneys are responsible for producing erythropoietin, the hormone that mediates red cell production. In fact, a study by Lebensburger et al. hinted at a possible protective effect of higher hemoglobin for kidney disease in children with sickle anemia [152]. This warrants further investigation, particularly because hemoglobin and hematocrit levels are modifiable factors which can be regulated via adequate medication.

Uncovering Potential Model Biases

The use of interpretability approaches make it straight forward to inspect potential model biases. For example, the fact that Elixhauser score features prominently across the cohorts might only be a reflection of the fact that the patients under analysis are in a critical condition. This realization does not necessarily add knowledge or increase the understanding of the underlying factors of the disease. A similar argument can be made for biomarkers which are usual proxies for kidney function, such as creatinine. Further, one can discern a relative preponderance of cardiovascular diseases across the cohorts. Indeed, cardiovascular risk factors and valvular disease have been linked to acute renal failure in the elderly [153]. However, this might be solely a reflection of the underlying cohort (heart surgery patients), which necessarily is likely to contain patients presenting those comorbidities. Another issue to consider is the role that the type of procedure itself plays in mediating AKI risk: it is arguable that valvular disease only affects outcomes for valve surgery patients. This prompts the necessity of a stratified analysis of the respective cohorts.

Updating Models for Better Validation Performance

Differences in model performance in the validation cohorts can be attributed to a number of factors. One factor is concerned with differences in incidence rate of the outcome. Another factor refers to the distribution of the model features: some of the model features might even be completely missing in the validation cohort. Since extracting additional features is potentially a resource-intensive activity, which could include for example performing laboratory tests, a feature should only be considered for extraction if its presence would be likely to improve model performance.

In this work's particular use case, one can ascertain that feature importance analysis suggests that laboratory values leading up to the surgery, one, two or up to three days before were important markers for AKI. This effect was particularly pronounced in the derivation cohort (MIMIC-III), therefore the model relied to a significant extent on these longitudinal values. Indeed, examining the distribution of these longitudinal laboratory values in the validation cohort (cf. Table A.1), one can observe that they are more often missing in the DHZB cohort as opposed to both the MIMIC-III and Mount Sinai cohorts. Case in point is BUN three days before surgery: missing for 85% of the DHZB patients (compare to 73.5% and 60.8%, for MIMIC-III and Mount Sinai, respectively). This actually reflects another issue: discussion with the medical partner at DHZB revealed that in this German hospital surgical patients are usually admitted only shortly before the procedure takes place. As such, laboratory values days before surgery are not available as a rule. This observation suggests that in the DHZB cohort patients were admitted to the hospital before any monitoring could take place. The different levels of feature 'missingness' possibly suggest a different patient case mix. This difference in patient case mix is also reflected in the Elixhauser score. Upon closer inspection of the distribution of this variable, one can observe that its mean value in the DHZB cohort is substantially lower than in the other cohorts (Beth Israel and Mount Sinai). This is indicative of the fact that in these hospitals patients tend to treat patients presenting a high spectrum of diseases (higher Elixhauser score), contrasting with the DHZB, an institution focusing solely on heart patients.

Building on these insights, possibilities to update the original model for better validation performance could be: 1) e.g., exclusion of Elixhauser score in favor of its constituent comorbidities, 2) limiting model features to lab values available upon or shortly before admission, which are more likely to be available, or 3) though more difficult, increasing monitoring of patients at risk so that a longer history of data points is available. It is important to notice that a mere analysis of the distribution of different variables while informative, would have offered little towards understanding underlying causes for performance differences from the perspective of the model itself. This strengthens the case for the use of interpretability approaches in clinical predictive modeling.

4.6. Limitations

A number of limitations apply to the work described, particularly with respect to feature selection bias, summarization of longitudinal data, robustness of the interpretability methods in general and finally the lack of an impact analysis of the clinical model in practice.

Selection Bias

I conducted the initial variable selection via hand-crafting the features: after literature exploration and expert consultation, a set of features were identified which guided model development. As such the features selected only cover a fraction of those which are potentially available in an EHR. Indeed, ML models thrive when provided with copious amounts of data. However, increasing the number of features might lead to issues with overfitting, potentially limiting even more the reproducibility of the results [154].

Summarizing Longitudinal Data

Another issue to be taken into account is the approach I followed to process the laboratory data used: for each of the periods, the mean value of the feature was used, i.e., a feature summary. By processing these longitudinal features in this way, relevant information is possibly lost, which could be informative for the model. Better approaches exist to handle this problem, such as alternative time-series representations, e.g., using Symbolic Aggregate approXimation (SAX) [155]. Also, deep learning approaches using Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks could be employed that require little feature pre-processing [156].

Robustness of Interpretability Methods

The interpretability algorithms themselves are also not entirely without downsides and pitfalls: explanations tend to lack robustness, i.e., different methods might provide different explanations for the same model [157]. For one, while global surrogates such as mimic learning are flexible, the conclusions drawn concern the model, not the data, since the surrogate model does not have access to the actual data. As such, explanations tend to be only as good as the original model. Furthermore, since the choice of surrogate impacts how well it can mimic the original model, i.e., its R^2 values, the surrogate itself will bring along a host of potential issues. Second, local surrogates such as LIME tend to exhibit a considerable degree of instability for their explanations. This happens because of the random neighborhood sampling inherent to this method. In other words, if the sampling process is repeated, one might obtain different explanations for the same instance, calling into question its robustness. I sought to mitigate this effect by applying submodular pick and averaging out the contributions across many different explanations.

For these reasons, an overall feature rank could be developed which considers a range of different interpretability methods, taking into account how often features are mentioned across methods and with what importance [15]. In other words, if a feature shows up in multiple interpretability methods as important, its contribution towards the outcome of interest is more stable. Therefore one could surmise that the particular feature is either a) reflective of properties inherent to the use case or b) inherent to the data (learned biases). If the feature is inherent to the use case, a desirable property, one can hypothesize that using those particular features for updating a model in a validation study would be beneficial, i.e., lead to simpler, easier-to-understand models. Ultimately, such hypotheses drawn from such interpretability algorithms must be validated by experiment for claims to be considered valid. In the next chapter of this thesis, I address this very issue: how can interpretability methods provide a contribution towards updating clinical predictions models in validation studies.

Impact Study

Validation studies such as the one I conducted in this work are only the first line of validation. If a model performs well in an external cohort, it is an *indication* that it can perform well in practice. For a model to be deployed in the clinic, a comprehensive evaluation in the clinic should be conducted. This could take two forms in the clinical setting. The first one is to compare risk assessments made by the model with those made by the doctors themselves and compare the outcomes: the model does not need to be perfect, only better than the doctors. This approach, however, only provides information on the model quality, but not how its use could impact patient outcomes or the quality of care itself. The second option is more resource-demanding: conducting a fully-fledged RCT, in which two cohorts of patients are treated based on the prediction probabilities generated by the model, and another follows the standard care pathway.

4.7. Conclusion

In this chapter, I presented the development, validation and interpretation of a CPM that makes use of ML algorithms on EHR data to help predict the risk of AKI using parameters collected prior the actual procedure on the heart. Prospectively, the use of such a predictive model can help physicians to identify high-risk patients so that protective and/or preventive measures can be adopted early in the treatment. Examples for such measures are readying renal replacement resources in advance and avoiding nephrotoxic agents. A range of modeling algorithms was put to analysis. The best model outperformed established clinical scores for post-surgical AKI onset by a significant margin upon derivation (0.90 vs. 0.83 AUROC for the best-performing algorithm). In the current literature, validation of CPM is performed only rarely. In this work, I analyzed in-depth how the model performed when applied in two external cohorts, German Heart Center and Mount Sinai. The excellent results achieved could not be reproduced to the same extent in the validation cohorts, with performance deterioration most pronounced in the German hospital cohort (0.90 vs. 0.76 AUROC) even after re-training the model.

While a performance difference was expected to some degree, the issue is compounded by the use of 'black-box' algorithms. As the first work in the literature for clinical predictive models, I applied local and global interpretability methods side-by-side in order to illuminate possible reasons that could account for performance differences, including potential model biases inherent to the derivation cohort (MIMIC-III). Indeed, the methods employed highlighted particular characteristics of the CPM developed, which, as it turned out, relied considerably upon longitudinal lab values, and on an aggregated marker, the Elixhauser score. Even though the insights obtained can potentially inform model update for external validation, hypotheses drawn from these methods must be validated by

experiment, i.e., via iteratively refining the model. Potentially, this might lead to more generalizable models that are easier to understand for practitioners. Open questions remain, however, regarding the robustness of interpretability methods, an issue that warrants further investigation. Finally, a major weakness is the lack of an analysis on the impact of the model on patient outcomes, which requires setting up a RCT.

Credits

Next to my own contributions in developing, validating and interpreting the models, designing and conducting the experiments and analyzing the results, the following credits are due with respect to the work presented in this chapter. Boris Pfahringer performed data extraction and ran experiments for the German Heart Center validation cohort. Tom Martensen helped with data extraction for the Mount Sinai validation cohort. Frederic Schneider helped with data extraction from the MIMIC-III cohort and generated first models for the AKI use case with his `akilearner` framework. Siegfried Horschig helped with exploring and implementing the mimic learning approach utilized. Alexander Meyer served as medical advisor. Matthieu-P. Schapranow served as technical advisor and reviewed the manuscripts of the case study papers [2, 3].

5. Explanation-Driven Recursive Feature Elimination

"We are not going in circles, we are going upwards. The path is a spiral; we have already climbed many steps."

—Hermann Hesse, *Siddhartha (1922)*

How to select the best features for modeling? In this chapter, I demonstrate how applying feature selection approaches can lead to simpler, more interpretable models. Different works exist evaluating such methods in the omics domain, but remains scarcely explored in predictive modeling, particularly in external validation studies.

5.1. Introduction

In Chapter 4, we observed that model performance upon validation deteriorated with respect the validation cohorts. When validation performance is worse than that of model derivation, a procedure called 'model updating' can be applied. In this context, it consists in adapting model parameters or the model itself to the characteristics of the external validation cohort [158]. This is a straight-forward process when it comes to models such as logistic regression, in which it suffices to update regression weights or adjust decision thresholds. It is a less trivial task with respect to "black box" approaches such as GBDT or RF, since "the reasoning behind the function is not understandable by humans and the outcome returned does not provide any clue for its choice" [159]. This issue is compounded by the fact that ML often rely on a large number of predictors, allowing for non-linear relationships to be learned, but at a greater risk of overfitting to the original dataset.

Therefore, if a researcher wishes to update the original predictive model in such as way as to increase its generalizability on a validation cohort, she should investigate what model features are most likely to reflect intrinsic characteristics of the phenomenon under analysis. As such, a fundamental question presents itself: what features should be included in the model that will provide acceptable performance for both the derivation *and* the validation cohort? Furthermore, often, not all of the features used on derivation can be found in validation. Take, for the sake of example, a predictive model that partially relies on hard-to-acquire serum biomarkers: since this involves obtaining a blood sample and laboratory analysis, this feature should only be required in a validation scenario if the model's performance depends to a large extent on its availability. Therefore, a similar question arises: what features *not* present in the validation cohort should be extracted, which will lead to most significant gains in performance? Identifying the right model features to keep can lead to simpler models, with shorter training times, and enhanced generalization.

In fact, a host of feature selection approaches exist, which are aimed at selecting the most informative features in a dataset. In the bioformatics domain, there has been extensive research evaluating such methods. However, comparatively fewer studies have been conducted specifically in the clinical predictive modeling domain with a particular focus on performance on an external validation cohort. Furthermore, these methods rely to a large extent on the training data itself, therefore with a high potential for overfitting. For example, a feature selection method applied on one cohort is likely to

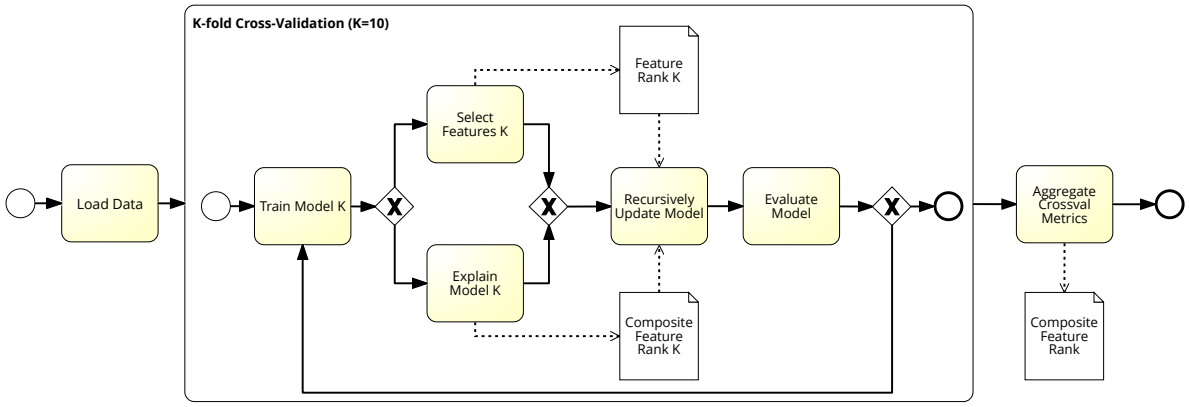


Figure 5.1.: Graphical abstract of the update procedure using Business Process Modeling Notation. The composite feature rank is obtained by averaging the metrics obtained within the cross-validation loop.

be of little value in a validation cohort, among others, because the feature distributions might be fundamentally different. We examined previously how interpretability methods can help shed light on a model's intrinsic behavior, providing key insights into the model's decisions. I hypothesize that combining the feature rankings provided by each of the interpretability methods into a single, composite feature ranking, can be used to develop models that perform better than traditional feature selection methods in an external validation cohort.

To test this hypothesis, I apply four interpretability methods, global and local surrogates, along with method-based feature importance and Shapley values on the use case AKI discussed previously to derive a composite feature rank, which will be compared against state-of-the-art feature selection methods. Figure 5.1 outlines the overall procedure followed in this work for obtaining the composite score. This approach will be evaluated on the validation cohort (Mount Sinai) and on 25 other synthetic cohorts based on it, which were generated using Generative Adversarial Networks (GANs).

The rest of this chapter is structured as follows. Section 5.2 relates a brief overview of the feature selection methods I will employ and the model update procedure. Section 5.3 expounds on the Explanation-Driven Recursive Feature Elimination (ED-RFE) procedure itself and the experiments designed to evaluate it, while Section 5.4 presents results obtained. Finally, Section 5.5 outlines their the significance.

5.2. Related Work

This section provides a brief overview of related work with respect to model update, feature selection methods and the recursive feature elimination procedure.

5.2.1. Model Update

Su et al. outlines three possible courses of action: 1) coefficient updating, 2) combining multiple previous CPMs in a meta-model and 3) dynamic updating of models [160]. The first strategy is widely employed when it comes to models based on LR and therefore not necessarily applicable to ML-based models. The second approach requires the availability of those multiple CPMs based on a range of different cohorts. Since obtaining validation cohorts is a challenging task, combining different models might be less practical. Finally, the last approach takes into account how patient populations change overtime. In this thesis, I consider an updated model as one that was trained from scratch on the

validation setting, as opposed to incremental learning or transfer learning [161]. Therefore, the *unit of knowledge* that is transfered across cohorts is the respective feature rank obtained from the derivation cohort.

5.2.2. Feature Selection Methods

Feature selection methods help practitioners to identify redundant or non-essential features in predictive models. This is of particularly relevance for ML models, which tend to learn specific characteristic of the data at hand. Applying such methods can lead to simpler models, and therefore better interpretability and shorter training times, and prospectively better generalization, owing to reduced overfitting [162]. Roughly, current techniques can be categorized as filter, embedded or wrapper methods, which apply different approaches to selecting the best features.

Filter Methods. This approach operates by establishing a proxy of feature importance for each feature, or relevance index, independently of the classifier used [163]. These methods are typically computationally inexpensive and because they are classifier-independent, they tend to expose relationships between features, since they do not rely on assumptions made by the underlying prediction model. Examples of such methods are correlation-based approaches, such as t-test, F-test and mutual information [164] and ReliefF algorithms [165].

Embedded Methods. Another broad category of feature selection methods are the one in which the selection procure is part of the model building itself, i.e., embedded into it. In the Least Absolute Shrinkage and Selection Operator (LASSO) technique, this is achieved by introducing a regularization coefficient, which shrinks non-relevant regression coefficients of a model towards zero with a L1 penalty [162]. Another prominent example of this category of approach is the Elastic Net approach, which linearly combines the L1 penalty of LASSO with the L2 penalty of ridge regression [166].

Wrapper Methods. Finally, this category of methods rely on a given predictive model to evaluate a different candidate feature subsets, developed on a training set and evaluated on a held-out test set [167]. Each feature subset is evaluated on the basis of a given metric, for example, AUROC, thus establishing that subset's score. Because a new model needs to be trained for each subset, these methods tend be computationally costly. On the upside, given their reliance on a specific predictive model, they tend to provide the best results in comparison to the previous two approaches [162]. A widely employed method in the biomedical domain, Boruta, is a wrapper method built with random forest, which iteratively compare the importances of actual features against that of synthetic features [168].

The Recursive Feature Elimination (RFE) technique can be interpreted as an instance of a wrapper method, in which different feature subsets are continuously evaluated, aiming towards progressive model enhancement. Technically, RFE is a backward selection of predictors, i.e., it starts out with the full feature set, features are removed until an optimal number of features can be achieved according to a scoring function. In essence, it initially builds a model on the full set of predictors and ranks those predictors using a metric of feature importance. Using this ordered sequence of predictors, at each step of the procedure, k variables are removed, and the model is re-fit and re-evaluated on the retained features. Finally, the top features are used to fit the final model [169]. Given the method's procedure, the choice of feature importance metric has an important impact the performance of the approach. The original paper by Guyon et al. employed Support Vector Machines (SVM) feature importances, but other alternatives such as mean decrease in Gini impurity from RF could be used [169]. Being a wrapper method, while this procedure is more computationally intensive, it tends to provide better results in comparison to filter or embedded methods.

5.2.3. Knowledge Gaps

Knowledge gaps can be identified with respect to two main aspects. The first refers to the relative lack of comparisons of feature selection methods particularly in the clinical modeling domain. In the bioinformatics domain, particularly when it comes to gene expression, a number of studies compare the performance of myriad of feature selection methods seeking to ascertain the 'best' method in high-dimensional datasets [17, 170, 163, 171]. Gene expression datasets, by their very nature, tend to provide similar sets of features, making such studies possible. However, little has been examined in the domain of clinical prediction models, with two works by Sanchez-Pinto et al. [17] and Bagherzadeh et al. [172]. Given the inherently empirical nature of such studies, it is challenging to establish an undisputed 'winner'. Therefore, more empirical examinations in the clinical modeling domain are needed.

Second, the few studies which exist do not deal with the issues of evaluating the results obtained in external validation cohorts. Critically, no work so far in the clinical modeling domain has particularly analyzed the issue of *generalizability* of the results obtained by the different feature selection algorithms, for example, in the context of a validation study. Results obtained in the literature suggest that feature selection in clinical predictive models can indeed improve model performance with higher simplicity. However, it is not known whether these promising results can be transferred to external validation cohorts. In other words, do the feature selected upon derivation also perform well in an external validation cohort? If the model has learned underlying feature relationships this should be verified in practice. This issue merits further investigation because one of the quintessential promises of feature selection is to reduce model bias.

Finally, we saw in Chapter 4 how the use of interpretability methods can aid in understanding the underpinnings of the models themselves. To a certain extent, these interpretability methods could also be used as feature selection methods, since they provide a feature ranking. We also observed how the outputs of those methods differ from one another (cf. Figure 4.8). While this lack of robustness can be a detrimental factor, if one combines those outputs into a single composite ranking, this can potentially lead to better generalization in an external validation cohort, because one could then achieve a consensus on the importance of different features. In effect, this consensus-based approach has been shown to perform better than individual methods in the biomedical domain [17]. Therefore, I adapted the RFE procedure using a consensus-based or composite feature ranking, i.e., ED-RFE, building on the individual feature rankings of different explanation methods.

5.3. Methods

This section outlines in the detail the experimental set-up pursued in this chapter, along with a brief description of the data generation procedure. Finally, I explain how the ED-RFE procedure was carried out.

5.3.1. Experimental Set-up

The experimental set-up is divided in three experiments, which will be evaluated by means of AUROC as evaluation metric on the AKI use case. Overall, the experiments seek to ascertain how adequate are the different methods to the task at hand.

Experiments. The first analyzes the performance of the ED-RFE approach on the derivation cohort (MIMIC-III) and validation cohort (Mount Sinai). This experiment seeks to answer the question: how generally adequate is the ED-RFE procedure, in other words, is it possible to obtain simpler models using this procedure as is? In the second experiment, I compare the ED-RFE approach to

the other feature selection methods, i.e., I use RFE in which the feature rank is determined by the feature selection algorithms themselves. This experiment answers the question: how does the ED-RFE approach compare to other methods in terms of discriminative performance? In the final experiment, using 25 synthetic cohorts based on the validation data, I apply the RFE procedure once again, aiming to obtain a generalization for the results obtained in Experiment 2. As such, this experiment helps to answer: to what extent are the results obtained statistically valid?

Evaluation. For the evaluation, I utilize the use case AKI, as discussed in Chapter 4, using the AUROC as primary evaluation metric. I correct for bias by applying 10-fold cross-validation throughout the experiments. Finally, note that throughout the rest of this work, I focused on the two best-performing algorithms according to the evaluation in Chapter 4, i.e., Gradient Boosting Decision Trees (GBDT) and in particular Random Forest (RF).

5.3.2. Data Generation

A thorough evaluation of the methods requires the availability of appropriate datasets. Since obtaining validation datasets for clinical predictive modeling can be a daunting task, I resorted to synthetic data generation. Many approaches exist for this task, for example using Bayesian networks [173]. Recently, the use of GAN-based approaches has been demonstrated to outperform conventional statistical methods in capturing inter-column correlation, scalability and data privacy protection [174, 175]. I utilized the method Tabular GAN (TGAN) for this task [175].

The basic GAN architecture relies on a generator which tries to 'fool' a discriminator with synthetic examples. In TGAN, the generator is a Long-Short-Term Memory (LSTM) network and the discriminator is modeled by a Multilayer Perceptron (MLP). This method has the additional benefit of modeling each data column separately, including both continuous and categorical features.

5.3.3. Statistical Evaluation

Following the recommendations from the literature, in addition to the individual metrics for each of the 25 datasets, I also provide a statistical evaluation of the results achieved across the datasets for each feature selection method [73]. Following Demšar's recommendations [73], first, I applied the Friedman test to ascertain whether statistically significant differences exist between the different selection methods for $\alpha < 0.05$. In the second step of the evaluation, I conducted a pair-wise Nemenyi post-hoc test to ascertain differences between the methods themselves.

5.3.4. Feature Selection Methods

For the performance comparison, I selected methods which are representative for each of the categories discussed in Section 5.2. As an example for filter methods, I utilized F-Test, Mutual Information and ReliefF. Since I am using RF as the underlying model, for the wrapper methods, I chose mean decrease in Gini impurity as feature importance and also Boruta using RF. Finally, as an instance of embedded methods, I selected Elastic Net, since it combines L1 and L2 regularization. Table 5.1 summarizes the methods used in the evaluation.

These are implemented via the job `Select` in MORPHER Toolkit. Using this resource, the task of imputing a dataset with KNN, performing feature selection with Boruta, training a default RF and evaluating on a held-out test set can be implemented as demonstrated in Listing 5.1.

Listing 5.1: Exemplary depiction of how to train a RF model with feature selection using MORPHER Toolkit jobs. Note that feature selection should take place *after* spliting, otherwise leakage my occur.

```
import morpher
from morpher.config import algorithms, imputers, selectors
from morpher.jobs import *
from morpher.metrics import *

''' define the input file and target variable '''
filename="cohort.csv"
target = "target"

''' First we load, impute and split the dataset in train and test '''
data = Load().execute(filename=filename)

''' data and target variables have been defined previously '''
data = Impute().execute(data, imputation_method=imputers.KNN)
train, test = Split().execute(data, test_size=0.2)

''' select relevant features with Boruta '''
train, selected_features = Select().execute(
        data=train,
        selection_method=selectors.BORUTA
)

''' then we train the given algorithms on the training set '''
models = Train().execute(
        data=train,
        target=target,
        algorithms=[algorithms.RF]
)

''' and evaluate them on the test set, slice by selected features '''
results = Evaluate().execute(
        test[selected_features],
        target=target,
        models=models
)
```

Table 5.1.: Feature selection methods employed and the corresponding implementations used. All these methods are available via the MORPHER Toolkit API.

Category	Method	Implementation
Filter	F-Test [164]	scikit-learn [45]
	Mutual Information [164]	scikit-learn [45]
	ReliefF [176]	scikit-rebate [177]
Wrapper	Gini Impurity [59]	scikit-learn [45]
	Boruta Selector [168]	boruta_py [178]
Embedded	Elastic Net [134]	scikit-learn [45]

5.3.5. Explanation-Driven Recursive Feature Elimination

Drawing on a joint feature ranking taking into account all interpretability methods, I apply recursive feature elimination on the validation cohort using knowledge gleaned from the *derivation cohort*. Using a measure of feature importance, it is possible to develop a feature rank for a given classifier, e.g., the ordered weights of a logistic regression classifier. By recursively removing features of the model based on this rank, it becomes possible to produce simpler models with similar performance, effectively an instance of feature selection [179]. This approach is called RFE and traditionally relies on a single measure of feature importance, such as the mean decrease in Gini impurity outlined previously. I extended this approach to utilize a composite feature rank, which relies on different interpretability methods taken together.

Composite Feature Rank

Since the use of the different explanation methods provides a convenient feature ranking, we can use a combination of those multiple ranks to obtain a more balanced view on feature importance across all methods. As such, this approach was extended as follows. First, I computed the feature rankings for each explanation method for a given model trained on the derivation dataset. Second, I normalized the calculated feature contributions to lie in the range of 0.1 - 1.0 (to avoid attributing zero to lowest-ranked features). Third, I obtained the specific feature's normalized average contribution across all methods, i.e., the feature's *mean*. Finally, I calculated a weighted average of the mean feature importances, using as weights how often the feature was mentioned across the different methods, i.e., its *support*. This was done in order to avoid having a feature dominating the rank which was mentioned only once. This approach is detailed algorithmically in Algorithm 3.

In this work, I considered the feature's mean importance and the support to be equally relevant. However, one could optionally, e.g., confer more importance to *how often* a feature was mentioned by the methods as opposed to mean importance.

Recursive Feature Elimination

The composite feature rank thus created can be construed as 'distilled knowledge' about underlying factors mediating the target variable in the derivation cohort. I sought to assess to what extent this knowledge is reflected back upon validation. If similar features turn out to be relevant in both cohorts, one could develop models that potentially perform well on both cohorts that rely on fewer features. Once the composite rank is calculated on the derivation cohort, I recursively train a model with different feature subsets with cross-validation to control for bias, as outlined in Algorithm 4.

Algorithm 3: Obtaining composite feature rank

Input: Trained Model, Features, Test Data, Explanation Methods
Result: Composite Feature Rank
// R is vector of vectors, dimension $|features| \times |methods|$;
Initialize rank matrix R across all $features$ and *methods*;
Initialize rank summary vector S across all $features$;
foreach *method in {explanation methods}* **do**
  //use *test data* if method requires
  Apply *method* on *model*;
  Obtain feature importance rank vector r_{method};
  Normalize r_{method} to $[0.1, .., 1]$ (min-max);
  foreach $feature$ *in* r_{method} **do**
    Assign importance of $feature$ to $R_{feature,method}$;
  end
end
foreach *feature in vector R* **do**
  Compute *mean* of $feature$ importance $mean_{feature}$ in $R_{feature}$;
  Compute *support* of $feature$ citations $support_{feature}$ in $R_{feature}$;
  // i.e., length of vector $R_{feature}$ (how often $feature$ was cited);
  Normalize $support_{feature}$ to $[0.1, .., 1]$ (min-max);
  $S_{feature}$ = Average of $mean_{feature}$ and $support_{feature}$;
  [Optional] use weight factor for mean and support;
end
Reverse-sort S;
Return Reverse-Sorted Composite Feature Rank S;

Algorithm 4: Model update with feature elimination

Input: Target Cohort, Composite Feature Rank S
Result: Cross-Validated Performance Metrics
Initialize K cross-validated train, test $folds \in$ *target cohort* ;
Initialize *max features*, i.e., total maximum number of features;
foreach *k fold in K folds* **do**
  Initialize feature subset s with all features $\in S$;
  while $|s| >=$ *max features* **do**
    Train model with feature subset s and evaluate it on *test*;
    Store model metrics for fold k and feature subset s;
    Remove lowest-ranked feature from s (reverse-sorted);
  end
end
for *Each k fold and feature subset s* **do**
  Compute cross-validated metrics (mean and standard deviation);
end
Return Cross-validated metrics for each feature subset on S;

5.4. Results

In this section, I present the experimental results in the following order. First, I utilize the composite feature rank to update the derivation cohort and validation cohort, i.e., the ED-RFE approach. Third, I compare the ED-RFE approach against the other feature selection methods on the derivation cohort (Mount Sinai). Finally, I apply the different methods, including ED-RFE, on the 25 synthetic cohorts and evaluate the statistical significance of the results.

5.4.1. Experiment 1: Applying the Composite Feature Rank

How adequate is the ED-RFE procedure? Figure 5.2 displays the results of this process for both RF and GBDT. First, the overall AUROC was higher for RF. Second, GBDT presents a substantially wider standard deviation in 10-fold cross-validation than RF, which potentially casts doubt on the robustness of the results. Third, as more features are added to the model, performance ceases to improve substantially, basically flattening out. For RF, the top 40 features seem to provide for *both cohorts* the best results. In the case of GBDT, this is less obvious because of the substantial degree of variance observed, but performance seems to be best at around 35 features.

Focusing on the RF model, Figure 5.3 shows the top 15 features which were selected by the procedure along with the scores obtained via 10-fold cross-validation from the derivation cohort (MIMIC-III). The results obtained by the top features Elixhauser score, blood urea nitrogen, fluid electrolyte imbalance and creatinine show a considerably higher range of values across the cross validation folds as compared to the other features. However, the values tended to be more uniformly distributed, as evidenced by the the probability density of the scores displayed in the violin plot. Note that the features included in the composite rank reflect to a certain extent the features analyzed in Chapter 4.

5.4.2. Experiment 2: Comparing Feature Selection Methods

How do the different selection methods compare to one another? In a first step, taking the RF model, I applied the RFE procedure, exchanging the feature rank computation (cf. Algorithm 4) for the feature importance rank obtained by applying the respective feature selection method. Then, using each of the respective feature ranks, I recursively applied it to the derivation cohort (MIMIC-III). The results are displayed on Figure 5.4. On the one hand, the RFE procedure using the composite feature rank from the explanations (ED-RFE) performed only marginally better than the feature rank provided by mean decrease in Gini impurity, i.e., feature importance from RF. With respect to F-Test, a somewhat more pronounced advantage could be ascertained, albeit only for the first 20 features. On the other, a marked benefit could be ascertained with respect to mutual information, ReliefF, Boruta and Elastic Net. Note that the last two methods do not provide a feature rank itself, e.g., Boruta only categorizes features as relevant or not, without a ranking among the relevant features. Similarly, with Elastic Net, only the non-zero coefficients are included in the method's feature rank. This is the reason why the not all feature subsets can be seen in Figure 5.4.

Does the knowledge obtained from the derivation cohort transfers to the validation cohort? To answer this, in a second step, I sought to assess the extent to the which the different feature ranks obtained in the *derivation cohort* perform in the *validation cohort*. For this evaluation, I decided to set the maximum number of feature subsets at 20, since this seemed to the number where differences between the methods was most apparent. The results obtained are depicted in Figure 5.5. Here again, the general trend observed in the derivation cohort is to a certain extent confirmed on the validation cohort. A slight advantage of the ED-RFE over Gini impurity, F-Test and mutual information ($\Delta \sim 0.01$ against Gini impurity). However, this advantage disappears as more features are added. Furthermore, the

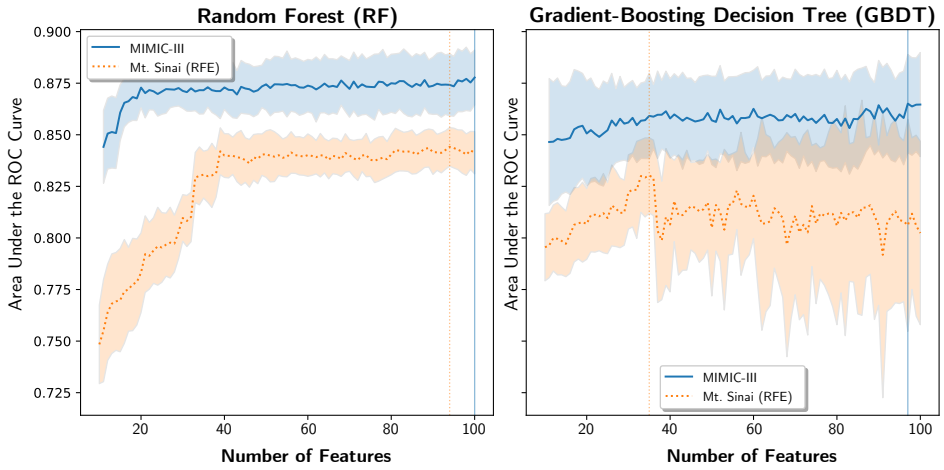


Figure 5.2.: Recursive feature elimination using the composite feature rank obtained from the derivation cohort. Note that iterations using RF showed lower standard deviation as compared to GBDT.

advantage of the ED-RFE is most pronounced for ReliefF, Boruta and Elastic Net, with a performance difference of ∼0.2 or 34% against Boruta @ 20 features.

Overall, the results obtained suggest that ED-RFE performs better than the methods evaluated with respect to AUROC. Only to a small extent against Gini impurity, F-Test and mutual information, but to a somewhat larger degree in comparison to ReliefF, Boruta and Elastic Net.

5.4.3. Experiment 3: Testing Statistical Significance

To what extent are these results statistically valid? To answer this question, I generated 25 synthetic cohorts using GANs, which allows us to try to generalize the results obtained. Figure 5.6 shows the metrics obtained in the different cohorts for RF @5, 10, and 15 features along the different methods. As such, each data point in the plot is the AUROC of training the RF on the synthetic validation cohort using the respective feature rank of the derivation cohort.

Using Friedmann's test, statistically valid differences between the groups could only be observed when utilizing the top 5 features, but not for the other combinations tested ($\alpha < 0.05$). Indeed, after carrying out pairwise Nemenyi significance test, one can ascertain that differences in performance by F-Test in comparison to all but the mutual information method are significant ($\alpha < 0.05$). Therefore, here, a different picture emerges as compared to the previous experiment: the slight advantage of the ED-RFE method over the others gives way to a better overall performance by the F-Test method (as measured by the distribution's median), namely, 0.66 vs. 0.71. Besides, ED-RFE showed no statistically significant differences with respect to all but F-Test and Boruta, performing better than the latter, but worse than the former ($\alpha < 0.05$). The features selected by the methods and their respective rank are displayed in Figure 5.8.

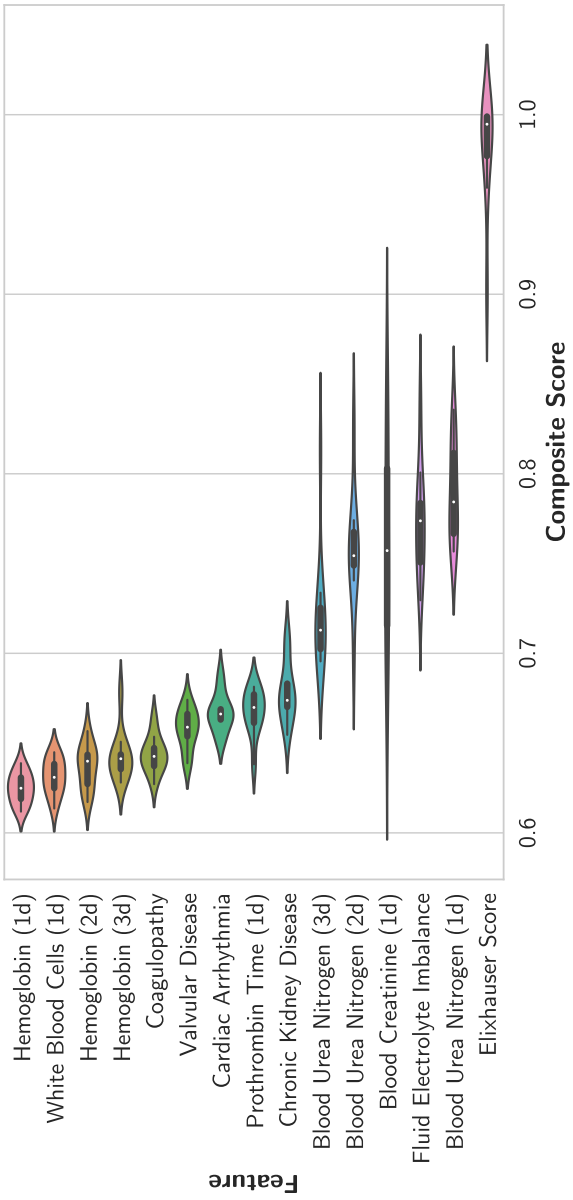


Figure 5.3.: Composite score of the top 15 features sorted according to median feature importance. Violin plots depict the different values obtained within 10-fold cross-validation. Some of the features are already known to the reader from Chapter 4.

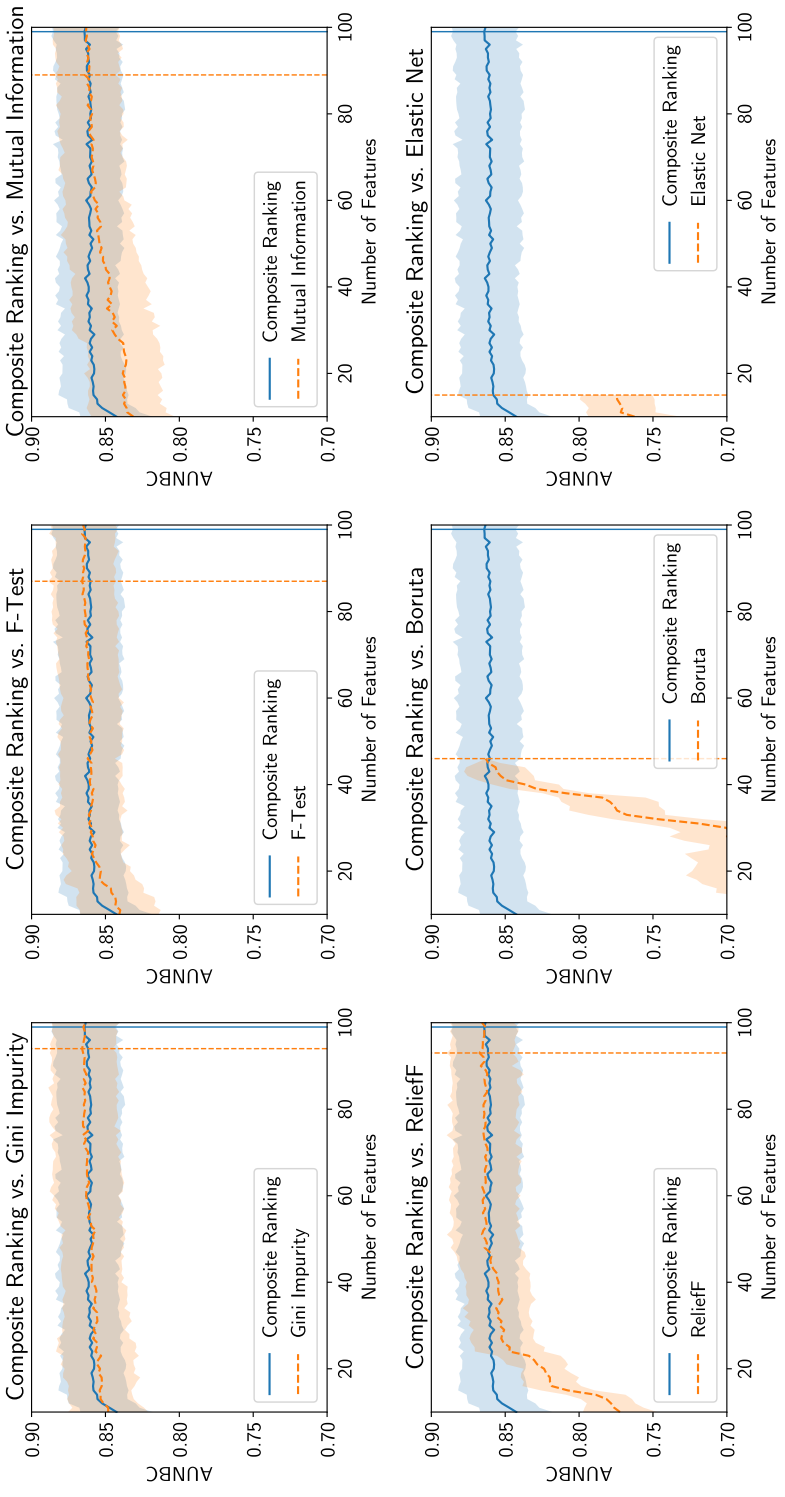


Figure 5.4: Area under the Receiver Operating Curve (AUROC) after applying Recursive Feature Elimination (RFE) using the feature rankings of the different feature selection methods obtained from the derivation cohort (MIMIC-III), compared to the composite feature ranking approach. Vertical lines highlight when model performance was highest. Shaded region depicts standard deviation of 10-fold cross-validation.

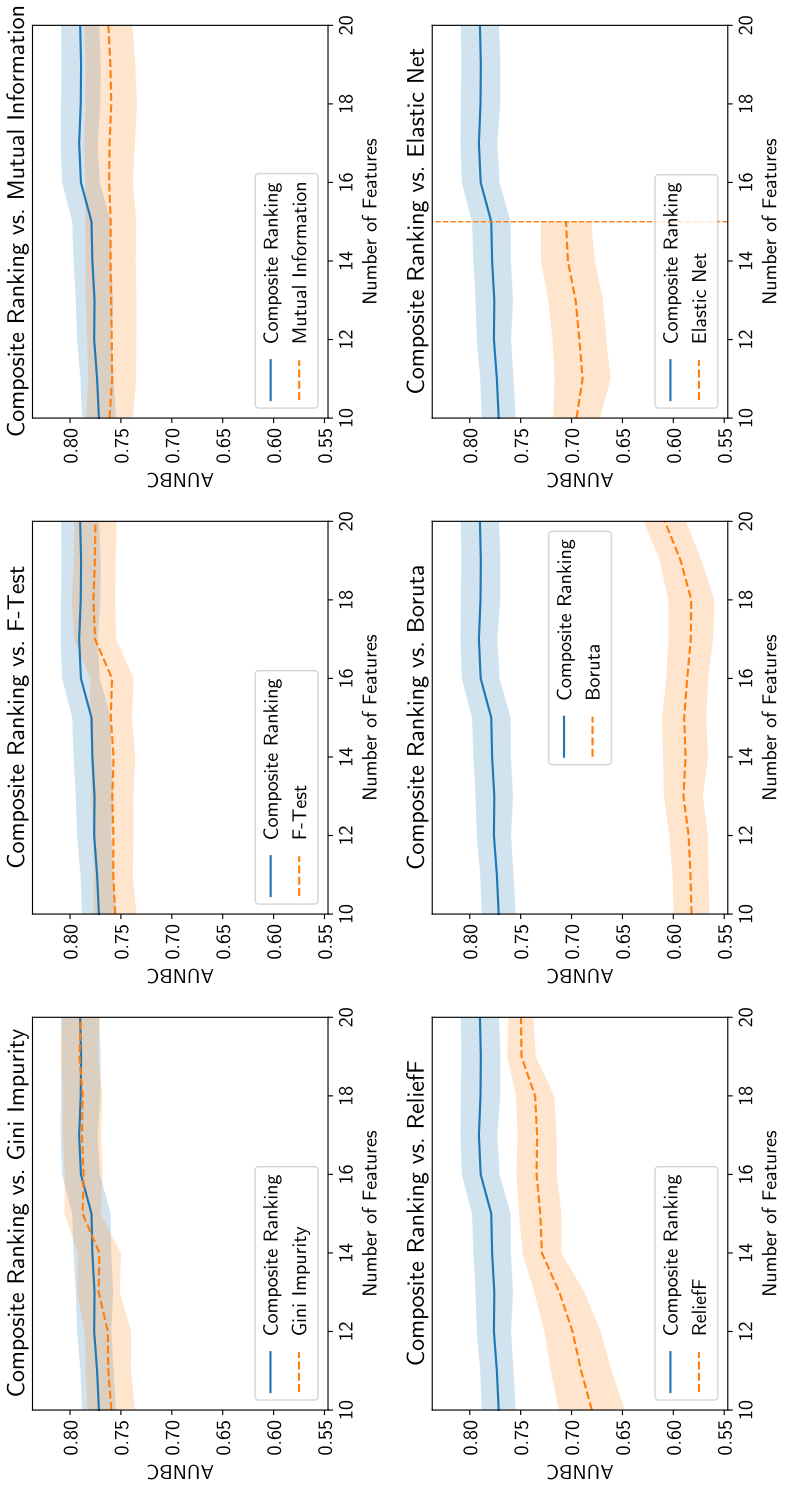


Figure 5.5.: Area under the Receiver Operating Curve (AUROC) after applying Recursive Feature Elimination (RFE) using the feature rankings of the different feature selection methods obtained from derivation cohort (MIMIC-III) applied on the validation cohort (Mount Sinai), compared to the composite feature ranking approach. Vertical lines highlight when model performance was highest. Shaded region depicts standard deviation of 10-fold cross-validation. Note that for the sake of enhanced clarity, only the top 20 features are displayed.

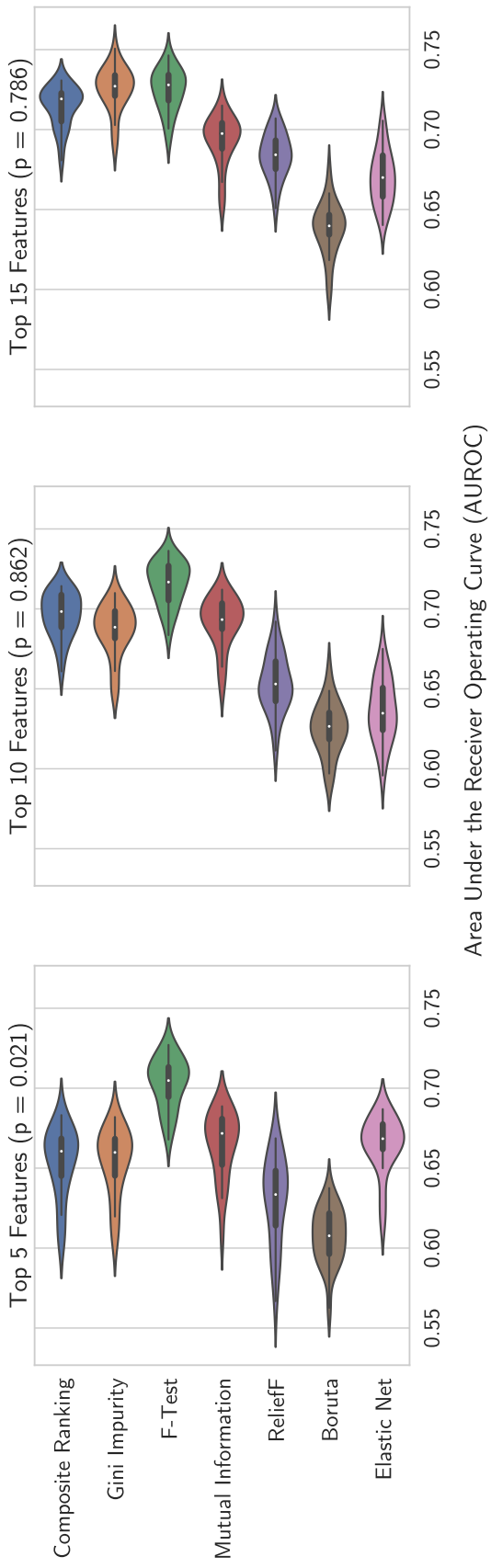


Figure 5.6.: Area Under the Receiver Operating Curve (AUROC) after applying Recursive Feature Elimination (RFE) using composite feature ranking of the derivation cohort onto the synthetic validation cohorts for the top 5, 10, and 15 features.

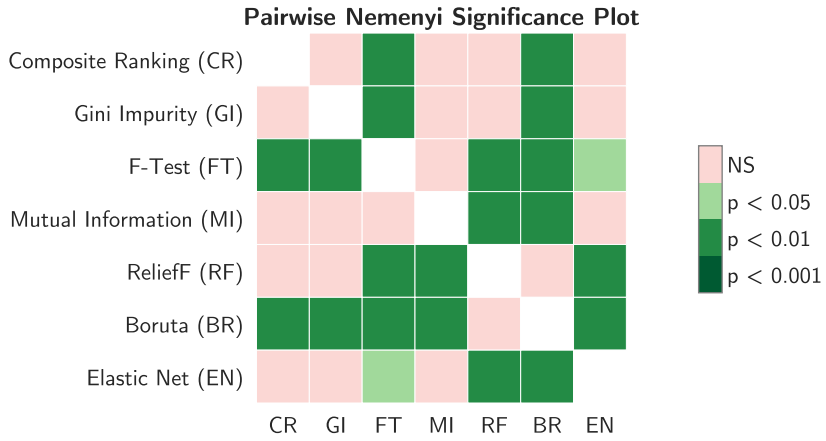


Figure 5.7.: Pairwise significance test of the different methods using Nemenyi test for models with the top five features. This significance plots makes it easy to visualize the significance of differences across the different methods.

5.5. Discussion

The following discussion will focus on the aspects concerned with the recursive feature elimination procedure outline above as well as pinpoint limitations inherent to the ED-RFE procedure and this evaluation.

5.5.1. Evaluation of the Feature Selection Methods

As demonstrated by the experiments, the choice of feature selection method has a substantial effect on model performance, both for the derivation and validation cohorts. Strikingly, the results obtained in the experiments performed, particularly Experiment 3, seem to suggest that the filter method F-Test performs better than more complex wrapper methods such as mean decrease in Gini impurity and our proposed method ED-RFE. Even though this result still needs further empirical evidence, since statistically significant differences could only be ascertained for the top five features, it runs counter to the usual notion that wrapper methods tend to perform better than filter methods. Indeed, previous works have indicated that simple methods such as Student's t-test provide the best predictive performance with respect to molecular signatures [170]. A similar observation has been made by Hua et al. on high-dimensional biological datasets, in which study classifier-independent filter methods such as the t-test performed better than ReliefF [171]. Yet another study by Abusamra et al. dealing with gene expression data reported similar results [17]. This trend had not been demonstrated in a *validation* study focused on clinical predictive modeling.

5.5.2. Explanation-Driven Recursive Feature Elimination

In this work, I departed from the assumption that using knowledge gained upon derivation via interpretability methods could be useful in updating models in a validation study, giving rise to simpler, more robust models, i.e., possessing fewer features. The aim of interpretability methods is illuminating a model's inner workings. Therefore, one would expect the model to reflect the underlying properties of the prediction problem. Since there is an inherent lack of robustness across the methods, combining the different methods could lead to a more comprehensive perspective on feature importance than any single methods [15].

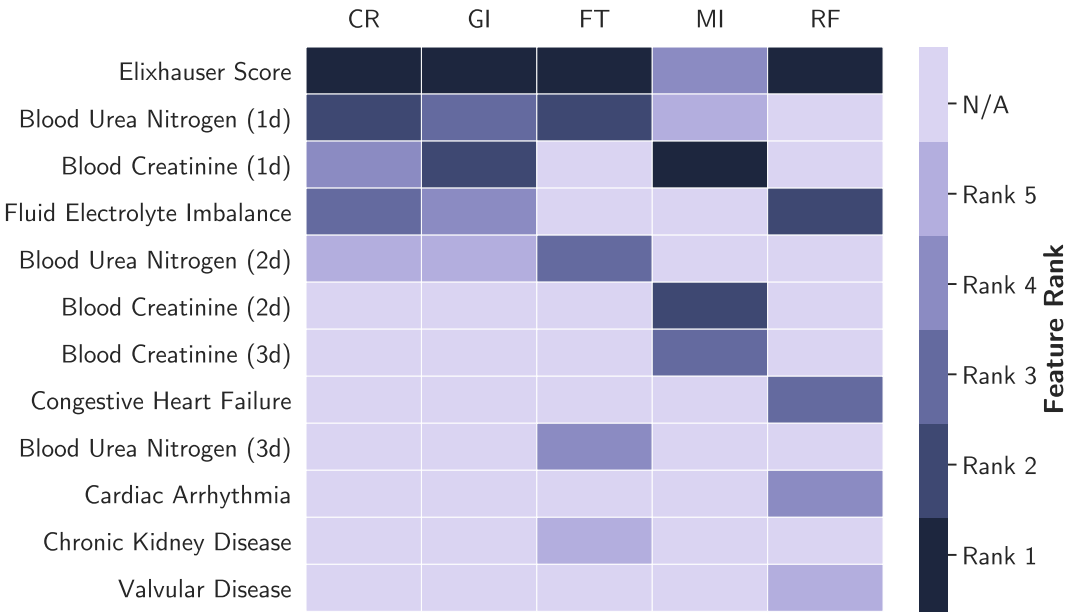


Figure 5.8.: Feature ranks of the top 5 features selected by each method. Note that Elastic Net and Boruta have been purposefully excluded from this graph, because they do not allow a direct feature rank. Abbreviations: CR=Composite Rank, GI=Gini Impurity, FT=F-Test, MI=Mutual Information and RF=ReliefF.

Indeed, as the results suggest, using the composite feature rank together with recursive feature elimination did lead to simpler models, i.e., from approx. 100 to overall 20 most predictive features performing similarly. It presented moderate advantages over the other methods tried, i.e., better AUROC for the same number of features, in the validation cohort (Mount Sinai). However, it is unclear whether the ED-RFE method should be preferred over a simpler feature selection model. First, the interpretability methods require substantially more computational power than simple statistical tests, because they often rely on repeated sampling of the underlying data, such as the local surrogate LIME. Another case in point is the computation of Shapley values, e.g., requires setting up multiple candidate predictor sets, with and without the given feature, a resource-intensive task. This issue is further compounded if cross validation procedures are used. Second, exactly because of the inherent data sampling in interpretability methods and the accompanying lack of robustness, the results obtained are subject to significant degree of variation, depending on the underlying model (cf. Figure 5.2). One possibility to mitigate this effect is to quantify and account for the robustness of the interpretability methods into the update procedure [157]. Despite the ED-RFE approach not performing substantially better than the filter method F-Test or the wrapper method based on Gini impurity in the experiments performed, it does not perform substantially worse than those methods either. Rather, it provides *further* insight into overall feature importance, potentially helping modelers to identify biases and develop better predictive models.

Limitations

A number of limitations apply to this work. First, I analyzed the performance of the candidate models using the different feature subsets only in terms of discrimination, a more thorough analysis should include the other dimensions of analysis mentioned, i.e., calibration and clinical usefulness discussed previously in Chapter 4. Such an analysis might reveal that while no benefits could be ascertained for the ED-RFE procedure for AUROC, the benefits might lie in other model metrics. Second, in a similar

vein, I did not consider in the evaluation other optimality criteria, which could help to establish a trade-off between accuracy and complexity, such as Bayes or Akaike information criterion (AIC), which penalizes model complexity. Third, I did not explore the effects of including or excluding correlated features as part of the analysis pipeline. Particularly in the case of random forests, previous studies reported that the effects of correlated predictors on the performance of feature selection methods are not unequivocally positive or negative [180]. Fourth, the statistical analysis was carried out on synthetic validation cohorts in a single use case (AKI), therefore the results presented are still subject to further evaluation.

Finally, our model update procedure relied on re-training the models from scratch. I tried to retain the knowledge from the derivation cohort by means of the feature ranks. However, given that the models need to be re-trained might compromise the final model metrics. While out of the scope of this evaluation, there are approaches which seek to keep the knowledge gained upon derivation for future cohorts. Most prominent among those is the field of transfer learning with neural networks, which relies on using knowledge from a previous task to learn a new, closely-related task [181]. In practice, this is achieved by re-training selected layers of the neural network. Another related approach is incremental on-line learning, typically applied in the context of learning from streaming data [182]. On-line Random Forest (ORF) is an example of an approach where trees and grown and discarded as new data is made available [183], which remains to be explored in the field of clinical predictive modeling.

5.6. Conclusion

Applying predictive models on external validation cohorts often leads to decrease in performance. Therefore, the original models need to be updated, so as to produce better results upon external validation. Since data extraction from EHR can produce datasets with a substantial amount of features, modelers are left with the task of selecting what features to keep and what features to discard. The problem is compounded in a validation setting: depending on the case, extracting the needed features from the EHR might be associated with substantial effort. In this context, feature selection approaches are often employed. However, since these methods rely on statistical properties of the training or derivation dataset, it is unclear to what extent those properties are transferable to external validation cohorts in clinical predictive modeling.

In this chapter, I presented an extensive evaluation of selected feature selection methods applied to a an external validation cohort. The evaluation also included our method termed Explanation-Driven Recursive Feature Elimination (ED-RFE), an attempt at distilling knowledge from the derivation cohort using interpretability methods. Since I do not make use of deep learning in this work, the goal was to attempt to trickle down the knowledge gleaned via interpretability methods from the derivation cohort onto the validation cohort, i.e., an assisted model update. For this purpose, I built a series of candidate update models for the validation cohort using the composite feature ranking from the *derivation cohort*. This was achieved by iteratively removing the least important features from the feature ranking and computing model performance for each feature subset. In essence, this approach utilizes a composite feature rank instead of solely using feature importance, e.g., mean decrease in Gini impurity, unlike the usual case in recursive feature elimination.

Remarkably, the results obtained in the different experiments suggest that filter methods such as F-Test perform at least as well as – or better than – more complex feature selection methods, including our Explanation-Driven Recursive Feature Elimination (ED-RFE), in this specific context of external model validation. The results agree with observations made in the extant literature pointing towards simple statistical testing as opposed to complex methods. Still, even though the ED-RFE procedure is computationally more demanding than simple statistical testing, it can highlight features that are not

captured by the other methods, further enhancing model understanding and informing model update. Therefore, I argue that the use of interpretability methods should become a routine aspect of clinical predictive modeling, including when performing feature selection for validation studies.

Future Work

This work was based on a single learning task (AKI prediction). A more thorough evaluation is needed which includes other tasks, in order to assess generalizability and statistical validity of the results. Furthermore, our evaluation was restricted to one single performance metric, and other metrics should be also be considered. Besides, it is also promising to assess how incremental learning or transfer learning techniques for model update compare against retraining the models from scratch in validation scenarios. Finally, I have not assessed every feature individually: introducing further evaluation dimensions such as ease of retrieval, objectivity or modifiability, i.e., the extent to which it is possible to *change* the feature to influence outcome, could provide a the foundation for better predictive models.

6. Conclusion

"Works of natural philosophy invariably include thesis and antithesis, the strict pro and con of a theory. A book which does not include its opposite, or 'counter-book', is considered incomplete."

—Jorge Luis Borges, *Tlön, Uqbar, Orbis Tertius. In: Ficciones (1940)*

What have we learned? In this final chapter, I revisit the research questions posed in the beginning, and analyze critically the contributions derived, while also outlining limitations and directions for future research.

6.1. Revisiting Research Questions

Have the research questions been answered? To recap, the questions posed in Chapter 1 refer to the standardization of clinical predictive modeling, the potential use of interpretability methods to uncover biases and their application for model update in validation studies.

To what extent can the development and validation of clinical models be standardized with software support across institutional barriers?

Standardization in any field almost necessarily comes at the expense of flexibility. As we have seen, the sheer number of modeling decisions that must be made by practitioners adds substantial complexity to the process: from the choice of imputation method all the way to how discrimination will be evaluated, these decisions impact model performance and ultimately influence whether those models will be deployed in practice. As such, any tool that aims to 'automate' this complex process will be met with challenges. Indeed, as the number of tools available grows, it becomes hard to ascertain what methods have been used, what parameters optimized and how the models were evaluated. How can we then guarantee reproducibility? With MORPHER, I was able to map and implement a substantial subset of those parameters, spanning the different process modeling steps (cf. Figure 2.1) via the Toolkit and user interface (MORPHER Web).

Furthermore, in the case study chapter (cf. Chapter 4), I focused exclusively on AKI, but the approach can be easily extended to any other disease or medical use case which can be modeled as a binary classification task and for which adequate data is available. Therefore, using either the Toolkit or Web frontend I developed, modelers already have a myriad of modeling options available, particularly with respect to model evaluation. As such, the modeling work can be shifted towards extracting the necessary data and identifying biases within these data by collaborating with medical experts. Besides, an added benefit of standardization is the possibility to reduce complexity and make it possible for non-technical people to learn the most important concepts quickly, so that they can enter a constructive dialog with data scientists. Indeed, the interviews revealed that there is a lack of knowledge on predictive modeling by medical researchers. The same point could be argued in reverse: there is lack of clinical knowledge by data scientists. As such, once again, collaboration is essential and MORPHER represents a first step towards this direction.

To what extent can the use of interpretability methods on black-box prediction models help illuminate model biases?

Chapter 4 provided an extensive treatment of the topic of predictive modeling in a concrete clinical task – onset of AKI following a heart surgery. Unlike what usually happens in the literature, I validated the model developed in two external cohorts, German Heart Center and Mount Sinai. Model validation revealed that promising results obtained in derivation cannot always be transferred to external cohorts. In this work's specific case this was most evident in the German cohort as opposed to the American cohort. These sobering results underscore the need for validation studies of existing models, instead of funneling resources towards developing new and more complex models relying on millions of data points.

Indeed, large EHR datasets can hold a multitude of spurious correlations, which ML models are particularly apt at capturing. Therefore, modelers need to 'restrain' their enthusiasm and question assumptions. To this end, we demonstrated that the use of interpretability approaches is a useful tool to help illuminate potential biases. In our concrete case study, it became evident that compound predictors such as Elixhauser score and others related to the underlying disease itself turned out as important predictors. To a certain extent, while this confirms that the model is sound, i.e., it learned an expected pattern, the utility of this information is limited, because this only confirms known risk factors: rather, more important are factors which contain surprising or unexpected content. In information theory, one would speak of a high *information entropy* [30]. Most interestingly in this work, high levels of blood-related factors such hemoglobin featured as *protective* factors in the analysis carried out. Though in different contexts, current work points towards an association to this effect [152], but this remains to be validated clinically.

To what extent can interpretability methods be useful in updating models in validation studies?

Departing from the hypothesis that interpretability methods could provide an advantage in selecting features, that perform well for both derivation and validation, i.e., helping to guide model update, I set out to evaluate the ED-RFE procedure against state-of-the-art feature selection methods. This novel method relies on the outputs of different interpretability methods combined. We have seen in Chapter 5 that indeed feature selection methods can be used to derive simpler models, i.e., models relying on a fewer number of predictors. The results of the experiments conducted suggest that there are minor gains in performance to be had using the ED-RFE procedure. However, even these minor gains did not withstand statistical evaluation using synthetic data sets based on the validation cohort (Mount Sinai).

While not final, as per the experiments, simpler methods, such as F-test, proved to perform better than more complex methods as measured by AUROC, particularly with respect to my ED-RFE approach. Reasons for this behavior cannot be pinpointed exactly, but because these algorithms rely on assumptions that can be partially verified in medical data. Furthermore, with respect to ED-RFE, a substantial degree of variation could be observed in the results, likely a result of the variability inherent to interpretability methods because of the sampling procedures involved in their algorithms.

6.2. Revisiting the Contributions

To what extent do this work contributions represent an advancement with respect to existing work? In the following the three contributions are scrutinized.

6.2.1. Software Platform (MORPHER)

Building upon the standardized modeling building blocks provided in the Toolkit (cf. Table 3.2), the software platform MORPHER makes it possible to rapidly prototype predictive models, either by explicitly defining the modeling parameters or by iterating through the available options in a automated mode. A similar functionality is provided in different analytics tools, such as RapidMinerGo. In the user study conducted, in which a clinical modeling task was posed, statistically significant differences could be ascertained only with respect to user interface elements: the RapidMinerGo required substantially less user input, while my tool required users to choose parameters explicitly. While a disadvantage at the surface, this means that users had to be more *aware* of the choices they were making. At any rate, this result prompted me to develop an optional automated mode. Apart from the user-facing software, the Toolkit, or the Python framework developed is a further contribution in itself: by providing out-of-the-box functionalities for modeling and evaluation. The framework can be continuously updated, so that recurring modeling tasks can be parameterized.

More importantly, however, current modeling tools do not cover the whole spectrum of the predictive modeling process. In particular with regards to the validation process, it is not trivial to validate models across institutions, which is provided via MORPHER. We did not address this aspect in the work, but this lays out the foundation for more sophisticated approaches, e.g., using federated learning and privacy-preserving data mining: since the underlying technical infrastructure is the same among the nodes, it would suffice to write the appropriate `MorpherJob` to handle this scenario (cf. Section 3.4.3).

Still, a critical issue that remains is the availability of data. The possibility to exchange models instead of data, i.e., to bring the algorithms to the researchers, provided by MORPHER is a promising way to foster reproducibility and increase the number of validation studies being published. However, admittedly, this is still only one piece of the puzzle: if the underlying data is not ready to be processed by the respective algorithms, model validation cannot take place. Obviously, this issue is probably as old as data itself and initiatives such as OMOP are gaining more and more acceptance. The way MORPHER was designed, it can be extended with ease to standardized data sources by means of available libraries such as inspectOMOP [121].

6.2.2. Case Study: Acute Kidney Injury (AKI)

In tackling the AKI use case, making use of different model metrics and interpretability approaches, I highlighted the importance of pursing a more comprehensive approach to modeling: it does not suffice to focus solely on discriminative performance, otherwise the results can be biased. With respect to the model itself, as one of the few studies in the field, I demonstrated the validation results of the model in two different hospitals leads to a more cautious evaluation of the performance obtained in derivation. The largest study on AKI by Google Deep Mind achieved excellent results on a vast cohort of patients, however their study is yet to be validated externally [140].

Furthermore, biases exist not only in terms of model features, but also lie in the choice of performance metric: excellent discrimination results as measured by AUROC do not necessarily translate into excellent calibration or clinical usefulness results. In particular, I demonstrated that a given model provides clinical benefit, meaning the trade-off between true and false positives only in selected decision threshold ranges, rather over all possible thresholds. These results highlight the need for more comprehensive evaluation in clinical modeling, a need recognized in the TRIPOD statement. Concretely, I made a contribution to this problem in that the recommended metrics by the TRIPOD statement are implemented out of the box in MORPHER, i.e., readily available for modelers.

Finally, a fundamental limitation is concerned with model impact on clinical care, which was not an object of this dissertation but is nevertheless critical. If a predictive solution is to be deployed at all in

practice, it should either provide tangible benefits for the patients, i.e., better outcomes, or substantially ease the burden of care personnel, which translates into better health care delivery. Such an evaluation requires setting up an appropriate RCT. This was not touched upon during this thesis, but MORPHER Web provides a set of APIs that could be extended to support, e.g., the FHIR `RiskAssessment` profile, which, in turn, could be directly integrated into a Hospital Information System (HIS) to facilitate clinical trials [184].

6.2.3. Model Evaluation

Feature selection is widely applied in bioinformatics for high-dimensional datasets, with several works evaluating strategies side-by-side. Given this genomics focus, however, few works to date had dealt specifically with clinical predictive modeling including external validation performance. As such, it had been unclear whether features selected from a derivation cohort would be equally important in a validation cohort.

Performing this task upon derivation can indeed be advantageous, as demonstrated in Chapter 5. As expected not all methods perform equally. As evidenced by the results, my ED-RFE did not substantially outperform simpler statistical methods (F-test). Indeed, similar results had been reported in the literature, with statistical methods performing better than more complex wrapper methods [17, 170, 171]. It is striking to also verify that for this specific scenario of feature selection across cohorts for clinical predictive modeling. To a certain extent, these results cast some shadow on the "algorithmic arms race" with respect to artificial intelligence applications in a medical context: more complex does not necessarily mean better. Rather, a return towards simpler methods, with fewer assumptions, could prove more beneficial than convoluted methods. Yet, the ED-RFE procedure represents a further tool in the modeler's 'utility belt' that can provide additional insight, possibly highlighting features which might have gone unnoticed by other methods.

The evaluation conducted had several limitations, most importantly the datasets utilized were synthetic and, as such, could have suffered from inherent statistical biases. Also, we restricted the evaluation to two ensemble algorithms, RF and GBDT. A different picture might emerge for other algorithms.

6.3. Directions for Future Research

What were lessons learned that can inform future research efforts? Here, I provide general directions for future research based on the insights gleaned throughout this work. These recommendations touch on aspects regarding 1) collaboration between data scientists and clinicians, 2) need for embedded educational support 3) the need for a formal experiment description language for clinical modeling, 4) data engineering tools for data preparation/harmonization, and the need of a feature taxonomy.

More support for collaboration between clinicians and data scientists. In the interviews conducted, one aspect that was often mentioned was the possibility to use MORPHER as hub for clinicians and data scientists to collaborate. This was not the original goal of the platform, so there are limitations. Upcoming tool support for predictive modeling should be designed with that very goal in mind: given that predictive modeling is by definition an interdisciplinary endeavor, solutions to streamline this process are welcome. For example, models could go through a 'life cycle', from inception till maturity (multiple validations), in which each step has input from both clinicians and data scientists iteratively.

Need for embedded educational support. Clinical modeling is a complex activity by definition. Without the proper background knowlegde, any modeling tool can be misused and produce questionable results. Building the required knowlegde takes time, but this can be somewhat facilitated

by appropriate educational resources embedded in the tools themselves. This could be achieved, e.g., using integrated quizes, linking litature references and other strategies. This is of critical importance if collaboration is to take place between clinicians and data scientists.

Necessity of a formal experiment description language. We have seen that a multitude of parameters can be defined when modeling, from imputation method, to sampling to evaluation metrics. While complex, those parameters *c*an be mapped, as we have sought to demonstrate with this work. However, the diversity of tools available will not go away, since every researcher is familiar with their own 'tried and true' approach. If a formal description language were available, in which most modeling decisions are documented in a structured, machine-readable format, available tools could easily parse this resource and replicate the experiment. I achieved this to a certain extent with MORPHER and BPMN 2.0, but more research into this topic is necessary.

Data engineering tools for data preparation. Modeling requires the availability of high-quality datasets that are structurally and semantically compatible if validation studies are to take place at all. While iniatives such as OMOP and openEHR are poised to solve this problem, their penetration remains limited. While the community waits for existing resources to be migrated to those standards, better tools are necessary which make it possible to match and harmonize datasets with less effort. Better yet, these tools should enable data quality experts and modelers to work together. Indeed, this was one of the frequently reported items during interviews.

Need for a taxomy of features in EHR data. A large part of this work was dedicated to analyzing the impact of specific features on model performance. However, different features have different inhrent properties. They can be either objective (such as laboratory value), or subjective (result from an anamnesis). They can be modifiable via a given intervention, such as blood pressure, and thereby have high clinical potential, or fixed, such as age, against which not much can be done, from a clinical perspective. They can be easy to acquire (body temperature) or be associate with risks for the patient (lumbar punction). Particularly coupled with interpretability approaches, a 'feature taxonomy' could be an important tool in further assessing the clinical utility of prediction models, particularly with respect to how they can be actually deployed in practice.

* * *

Ultimately, this work has demonstrated that more raw computational power is not necessarily the answer when it comes to clinical predictive modeling. Rather, clinicians and data scientists should embrace the existing body of knowledge and devise ways to make the art of modeling more reproducible, transparent and accessible.

A. Appendix

A.1. Case Study: Acute Kidney Injury

Table A.1.: Feature distributions of the three cohorts, including target variable Acute Kidney Injury (AKI). 1d, 2d, 3d refer to lab values *n* days prior to surgery. Numerical features are presented with mean values ± standard deviation and binary variables with the number and percentage of instances where feature is present, e.g. AKI=yes or AIDS=yes. MD (%) stands for missing data in percent. *p* refers to the p value of applying Ordinary Least Squares regression the independent variable AKI.

Feature	MIMIC-III (*N*=6,782)			DHZB (*N*=14,191)			SINAI (*N*=25,799)		
	Summary	MD (%)	*p*	Summary	MD (%)	*p*	Summary	MD (%)	*p*
Acute Kidney Injury	667 (9.83%)	0	0.000	5,449 (38.4%)	0.0	0.000	1,163 (4.51%)	0	0.000
Age at Admission	69.44 ± 28.46	0	0.041	67.71 ± 14.26	0.0	0.003	66.66 ± 15.95	0	0.506
Elixhauser Score	4.48 ± 6.32	0.5	0.020	0.9 ± 2.92	7.6	0.000	11.26 ± 10.91	0	0.000
Sex / Gender	4,657 (68.67%)	0	0.009	4,551 (32.08%)	0.0	0.000	15,331 (59.42%)	0	0.757
AIDS	11 (0.16%)	0.5	0.044	0 (0.0%)	7.6	0.982	0	100	n/a
Alcohol Abuse	173 (2.56%)	0.5	0.057	27 (0.21%)	7.6	0.386	253 (0.98%)	0	0.232
Blood Loss Anemia	53 (0.79%)	0.5	0.021	6 (0.05%)	7.6	0.924	525 (2.03%)	0	0.014
Cardiac Arrhythmia	3,516 (52.08%)	0.5	0.008	1967 (14.99%)	7.6	0.000	6,291 (24.38%)	0	0.002
Chronic Kidney Disease	669 (9.91%)	0.5	0.019	207 (1.58%)	7.6	0.299	8,467 (32.82%)	0	0.372
Chronic Pulmonary Disease	1,359 (20.13%)	0.5	0.021	291 (2.22%)	7.6	0.858	3,755 (14.55%)	0	0.127
Coagulopathy	546 (8.09%)	0.5	0.019	3 (0.02%)	7.6	0.117	12,427 (48.17%)	0	0.314
Congestive Heart Failure	1,938 (28.71%)	0.5	0.020	1,232 (9.39%)	7.6	0.000	1,050 (4.07%)	0	0.005
Deficiency Anemias	102 (1.51%)	0.5	0.021	6 (0.05%)	7.6	0.639	2,047 (7.93%)	0	0.653
Depression	372 (5.51%)	0.5	0.020	4 (0.03%)	7.6	0.916	60 (0.23%)	0	0.001
Diabetes w/ Complications	435 (6.44%)	0.5	0.744	7 (0.05%)	7.6	0.748	3,124 (12.11%)	0	0.128
Diabetes w/o Complications	1,750 (25.8%)	0	0.636	207 (1.58%)	7.6	0.722	327 (1.27%)	0	0.953

Feature	MIMIC-III (*N*=6,782)			DHZB (*N*=14,191)			SINAI (*N*=25,799)		
	Summary	MD (%)	*p*	Summary	MD (%)	*p*	Summary	MD (%)	*p*
Drug Abuse	108 (1.6%)	0.5	0.020	6 (0.05%)	7.6	0.450	10,637 (41.23%)	0	0.032
Fluid Electrolyte Imbalance	820 (12.15%)	0.5	0.019	1 (0.01%)	7.6	0.339	17,097 (66.27%)	0	0.006
Hypertension	4,732 (69.77%)	0	0.233	304 (2.32%)	7.6	0.000	2,789 (10.81%)	0	0.772
Hypothyroidism	577 (8.55%)	0.5	0.703	9 (0.07%)	7.6	0.282	812 (3.15%)	0	0.032
Liver Disease	223 (3.3%)	0.5	0.019	66 (0.5%)	7.6	0.437	353 (1.37%)	0	0.002
Lymphoma	45 (0.67%)	0.5	0.020	7 (0.05%)	7.6	0.909	908 (3.52%)	0	0.483
Metastatic Cancer	19 (0.28%)	0.5	0.020	8 (0.06%)	7.6	0.340	1,155 (4.48%)	0	0.021
Obesity	544 (8.06%)	0.5	0.021	340 (2.59%)	7.6	0.581	820 (3.18%)	0	0.630
Other Neurological Issues	228 (3.38%)	0.5	0.020	15 (0.11%)	7.6	0.047	254 (0.98%)	0	0.134
Paralysis	35 (0.52%)	0.5	0.019	18 (0.14%)	7.6	0.225	126 (0.49%)	0	0.312
Peptic Ulcer	38 (0.56%)	0.5	0.076	3 (0.02%)	7.6	0.839	5,403 (20.94%)	0	0.689
Peripheral Vascular Disease	966 (14.31%)	0.5	0.020	2985 (22.75%)	7.6	0.000	238 (0.92%)	0	0.064
Psychoses	36 (0.53%)	0.5	0.285	2 (0.02%)	7.6	0.153	6,784 (26.3%)	0	0.362
Pulmonary Circulation Diseases	549 (8.13%)	0.5	0.020	244 (1.86%)	7.6	0.002	5,143 (19.93%)	0	0.000
Rheumatoid Arthritis	169 (2.5%)	0.5	0.341	18 (0.14%)	7.6	0.071	768 (2.98%)	0	0.716
Solid Tumor	72 (1.06%)	0	0.020	35 (0.27%)	7.6	0.556	1,524 (5.91%)	0	0.087
Valvular Disease	1,570 (23.26%)	0.5	0.021	5503 (41.95%)	7.6	0.000	1,8606 (72.12%)	0	0.460
Weight Loss	53 (0.79%)	0.5	0.020	10 (0.08%)	7.6	0.875	2,928 (11.35%)	0	0.137
Albumin (1d)	3.85 ± 0.48	80.9	0.946	3.32 ± 0.8	67.9	0.000	2.81 ± 0.6	66.4	0.024
Albumin (2d)	3.80 ± 0.47	90.4	0.239	3.36 ± 0.81	83.5	0.066	3.25 ± 0.7	87.1	0.028
Albumin (3d)	3.75 ± 0.49	92.2	0.762	3.28 ± 0.82	88.7	0.703	3.07 ± 0.64	85.2	0.244
Anion Gap (1d)	13.88 ± 2.65	54	0.762	0	0.0	0.885	10.62 ± 3.17	100	0.030
Anion Gap (2d)	13.90 ± 2.56	68.6	0.304	0	0.0	0.053	10.77 ± 5.42	100	0.449
Anion Gap (3d)	13.87 ± 2.63	75.1	0.331	0	0.0	0.001	9.5 ± 3.71	100	0.080
Bilirubin (1d)	0.71 ± 0.89	77.5	0.155	1.17 ± 1.71	77.7	0.792	1.04 ± 2.28	66.7	0.105
Bilirubin (2d)	0.73 ± 0.92	89.1	0.367	1.19 ± 1.75	89.6	0.883	1.29 ± 3.42	87	0.023
Bilirubin (3d)	0.81 ± 1.1	90.9	0.002	1.23 ± 1.82	92.8	0.850	1.18 ± 2.88	85.1	0.828
Blood Creatinine (1d)	1.32 ± 1.2	50.5	0.019	1.32 ± 1.0	61.5	0.000	1.2 ± 1.03	33.7	0.000
Blood Creatinine (2d)	1.37 ± 1.27	66.1	0.002	1.32 ± 0.98	78.3	0.007	1.28 ± 1.15	56.7	0.550
Blood Creatinine (3d)	1.43 ± 1.42	73.4	0.631	1.38 ± 1.1	84.7	0.407	1.27 ± 1.12	60.9	0.249
Blood Urea Nitrogen (1d)	23.91 ± 13.88	50.8	0.000	23.93 ± 14.92	61.6	0.000	21.32 ± 13.84	33.7	0.000
Blood Urea Nitrogen (2d)	24.76 ± 14.79	66.4	0.000	24.27 ± 15.01	78.6	0.101	23.79 ± 15.35	56.6	0.568
Blood Urea Nitrogen (3d)	25.95 ± 16.09	73.5	0.000	24.98 ± 16.41	85.1	0.110	23.8 ± 15.59	60.8	0.801

Feature	MIMIC-III (*N*=6,782)			DHZB (*N*=14,191)			SINAI (*N*=25,799)		
	Summary	MD (%)	*p*	Summary	MD (%)	*p*	Summary	MD (%)	*p*
Chloride (1d)	103.15 ± 3.89	52.7	0.108	0	100	n/a	102.15 ± 4.96	33.7	0.181
Chloride (2d)	102.92 ± 3.99	68.1	0.347	0	100	n/a	100.49 ± 4.61	56.7	0.049
Chloride (3d)	102.85 ± 4.24	74.5	0.002	0	100	n/a	101.04 ± 4.84	60.9	0.027
Chloride Whole Blood (1d)	105.79 ± 6.68	98.3	0.508	0	100	n/a	0	100	n/a
Chloride Whole Blood (2d)	105.89 ± 4.47	99.7	0.939	0	100	n/a	0	100	n/a
Chloride Whole Blood (3d)	105.74 ± 4.96	99.7	0.095	0	100	n/a	0	100	n/a
Glucose (1d)	133.41 ± 60.14	54.1	0.653	125.41 ± 50.06	79.4	0.087	123.69 ± 39.1	33.7	0
Glucose (2d)	129.14 ± 51.26	68.7	0.852	127.1 ± 52.67	85.4	0.329	118.89 ± 45.6	56.6	0.551
Glucose (3d)	134.04 ± 61.05	75.1	0.655	129.85 ± 54.05	90.2	0.079	121.36 ± 40.6	60.8	0.860
Glucose Blood Gas (1d)	190.88 ± 81.25	95.7	0.868	120.74 ± 45.92	21.7	0.000	0	100	n/a
Glucose Blood Gas (2d)	156.67 ± 44.24	98.8	0.000	171.7 ± 63.93	96.6	0.617	0	100	n/a
Glucose Blood Gas (3d)	172.00 ± 63.76	98.7	0.062	175.4 ± 61.73	97.6	0.972	0	100	n/a
Hematocrit (1d)	35.94 ± 4.92	50.9	0.092	37.32 ± 6.53	60.1	0.372	31.71 ± 4.73	32.1	0.133
Hematocrit (2d)	35.64 ± 5.01	67	0.918	37.17 ± 6.66	78.3	0.796	30.94 ± 5.22	55.8	0.851
Hematocrit (3d)	35.27 ± 5.05	73.8	0.562	36.84 ± 6.67	84.8	0.005	30.7 ± 5.01	60.6	0.477
Hematocrit Calculated (1d)	36.39 ± 5.68	94.8	0.000	0	100	n/a	0	100	n/a
Hematocrit Calculated (2d)	35.02 ± 5.76	98.2	0.001	0	100	n/a	0	100	n/a
Hematocrit Calculated (3d)	35.58 ± 6.08	98.3	0.082	0	100	n/a	0	100	n/a
Hemoglobin (1d)	12.24 ± 1.83	53.4	0.058	12.44 ± 2.37	60.1	0.754	10.69 ± 1.62	32.1	0.172
Hemoglobin (2d)	12.10 ± 1.87	68.7	0.219	12.32 ± 2.4	78.3	0.866	10.43 ± 1.76	55.8	0.991
Hemoglobin (3d)	11.95 ± 1.89	75	0.476	12.18 ± 2.42	84.8	0.001	10.35 ± 1.69	60.6	0.586
Hemoglobin Blood Gas (1d)	12.11 ± 1.89	94.8	0.000	11.93 ± 2.03	21.1	0.000	0	100	n/a
Hemoglobin Blood Gas (2d)	11.66 ± 1.93	98.2	0.001	11.3 ± 2.18	96.5	0.263	0	100	n/a
Hemoglobin Blood Gas (3d)	11.85 ± 2.03	98.3	0.087	11.07 ± 2.03	97.5	0.048	0	100	n/a
International Normalized Ratio (PT) (1d)	1.26 ± 0.44	57.5	0.077	1.39 ± 0.66	64.7	0.045	1.40 ± 0.52	52.9	0.165
International Normalized Ratio (PT) (2d)	1.28 ± 0.46	74.8	0.744	1.36 ± 0.57	80.8	0.937	1.39 ± 0.56	70.8	0.052
International Normalized Ratio (PT) (3d)	1.30 ± 0.42	79.7	0.056	1.41 ± 0.6	86.2	0.767	1.40 ± 0.58	74.3	0.612
Lactate (1d)	3.08 ± 2.48	96.9	0.002	1.1 ± 1.95	34.0	0.068	2.09 ± 1.56	73	0.696
Lactate (2d)	2.05 ± 1.82	99.1	0.186	2.49 ± 2.88	97.1	0.653	1.75 ± 1.97	97.3	0.153
Lactate (3d)	2.04 ± 1.16	99	0.929	2.39 ± 2.35	98	0.609	1.26 ± 1.22	94	0.081

Feature	MIMIC-III (*N*=6,782)			DHZB (*N*=14,191)			SINAI (*N*=25,799)		
	Summary	MD (%)	*p*	Summary	MD (%)	*p*	Summary	MD (%)	*p*
Platelet Count (1d)	235.26 ± 82.83	52.3	0.240	222.76 ± 89.82	60.1	0.222	198.79 ± 97.42	32.3	0.458
Platelet Count (2d)	238.49 ± 88.31	68	0.549	229.2 ± 95.21	78.3	0.924	196.12 ± 101.15	56.1	0.000
Platelet Count (3d)	239.48 ± 89.38	74.5	0.166	228.56 ± 97.16	84.8	0.920	182.42 ± 101.0	60.9	0.005
Potassium (1d)	4.23 ± 0.49	50.8	0.645	3.96 ± 0.56	21.1	0.001	4.21 ± 0.43	33.7	0.961
Potassium (2d)	4.24 ± 0.5	66	0.029	4.61 ± 0.57	96.6	0.010	4.15 ± 0.5	56.6	0.923
Potassium (3d)	4.21 ± 0.46	73.4	0.011	4.66 ± 0.55	97.6	0.224	4.21 ± 0.5	60.8	0.689
Potassium Whole Blood (1d)	4.23 ± 0.49	50.8	0.645	0	100	n/a	0	100	n/a
Potassium Whole Blood (2d)	4.24 ± 0.5	66	0.029	0	100	n/a	0	100	n/a
Potassium Whole Blood (3d)	4.49 ± 0.88	98.6	0.286	0	100	n/a	0	100	n/a
Prothrombin Time (1d)	13.94 ± 2.96	57.6	0.002	0	100	n/a	17.07 ± 4.28	52.9	0.534
Prothrombin Time (2d)	14.11 ± 3.16	74.9	0.621	0	100	n/a	16.84 ± 4.55	70.9	0.021
Prothrombin Time (3d)	14.33 ± 3.87	79.7	0.365	0	100	n/a	16.93 ± 4.47	74.3	0.632
Partial Thromboplastin Time (1d)	53.94 ± 30.92	55.3	0.000	43.89 ± 19.43	57.6	0.307	37.18 ± 10.77	55.7	0.000
Partial Thromboplastin Time (2d)	56.90 ± 32.35	71.8	0.881	43.78 ± 17.55	75.9	0.546	37.68 ± 12.07	73.4	0.093
Partial Thromboplastin Time (3d)	55.18 ± 31.86	77.9	0.154	44.91 ± 16.77	83	0.540	37.58 ± 11.29	76.3	0.059
Sodium (1d)	138.98 ± 3.05	53.1	0.490	140.12 ± 3.58	21.3	0.130	138.41 ± 3.41	33.7	0.000
Sodium (2d)	138.91 ± 3.17	68.1	0.103	140.7 ± 6.64	96.6	0.168	137.39 ± 3.61	56.6	0.001
Sodium (3d)	138.74 ± 3.29	74.6	0.411	140.68 ± 6.83	97.6	0.573	137.29 ± 3.67	60.8	0.001
Sodium Whole Blood (1d)	137.34 ± 4.33	96.9	0.101	0	100	n/a	0	100	n/a
Sodium Whole Blood (2d)	138.25 ± 3.54	99.6	0.250	0	100	n/a	0	100	n/a
Sodium Whole Blood (3d)	138.00 ± 4.16	99.5	0.075	0	100	n/a	0	100	n/a
White Blood Cells (1d)	8.85 ± 3.99	53.2	0.363	0	100	n/a	10.61 ± 5.11	32.2	0.000
White Blood Cells (2d)	8.84 ± 3.73	68.5	0.952	0	100	n/a	9.8 ± 4.49	55.8	0.014
White Blood Cells (3d)	8.91 ± 3.68	74.8	0.195	0	100	n/a	10.49 ± 4.6	60.6	0.187

A.2. Software Platform for Clinical Predictive Modeling

A.2.1. Constituent Items for Performance and Effort Expectancy

A.2.2. Modeling Task: How-To

This document was provided to study participants to inform them what to expect.

A.2.3. Modeling Task: Description

This document describes in detail what the required modeling task was.

A.2.4. Modeling Task: Questionnaire

This document contains the questions users were required to answer after using one of the selected tools.

A.2.5. Modeling Task: User Questionnaire

This document contains the information that was required from participants.

The documents are provided below.

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

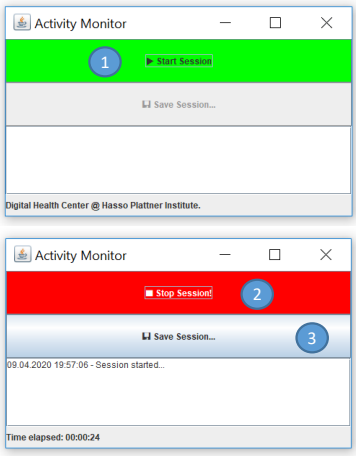
Modeling of Outcome and Risk Prediction for Health Research

Dear study participant, thanks for taking the time to participate in this research. This will help us to gain insights on how prospective researchers interact and use tools for clinical predictive modeling. You are expressly invited to ask any questions.

1. **Download the necessary files and access the assigned software.**
   To do so, follow the link provided to you via e-mail. It will contain the files needed to carry out the study, including the software link (please check `README.txt`), the modeling task description, the introductory video and the Activity Monitor. Unzip the contents to your prefered location.

2. **Watch the introductory video and try to replicate the results displayed.**
   The introductory video `Tutorial.mp4` will help you to get acquainted with the tool to carry out the modeling task. As you watch the video, try to replicate the steps presented to familiarize yourself.

3. **Complete the proposed clinical modelling task.**
   Read through the file `Task Description.pdf`. Once you are done reading, please launch the app `ActivityMonitor.jar` and click "Start Session" (1). Your mouse clicks and movements will be logged **locally**. You can now use the tool to complete the fulfill the task description.



4. **Save your activity information.**
   Once you are done using the tool, please press "Stop Session" (2) and then press "Save Session" (3). Please note that the saved file does not contain any person-identifying information (you can inspect it yourself).

5. **Now fill out the proposed task questionnaire.**
   You will have 15min to fill out the questionnaire. You do not have to answer everything – just do as much as you can within the allotted time. Do not forget to add your activity file contents to the form.

Questionnaire Link

https://forms.gle/gFCpo5SEC2iD6X9w7.

Thanks again for your participation!

Table A.2.: Constituent items for performance and effort expectancy according to Venkatesh et al.'s Unified Theory of Acceptance and Use of Technology (UTAUT) framework.

Construct	Metric	Using the System will...
Performance Expectancy	Perceived Usefulness	... enable me to perform clinical predictive modeling more quickly. ... improve my performance at clinical predictive modeling. ... increase my productivity in at clinical predictive modeling. ... enhance my effectiveness at clinical predictive modeling. ... make it easier to perform clinical predictive modeling. ... be useful in performing clinical predictive modeling.
	Job Fit	... have no effect on my performance at clinical predictive modeling. ... can decrease the time needed for my important job responsibilities. ... can significantly increase the quality of output in my clinical predicitive modeling tasks. ... increase the effectiveness of performing clinical predictive modeling tasks. ... increase the quantity of output for the same amount of effort. ... can in general assist in clinical predictive modeling considering all tasks.
	Relative Advantages	... enables me to accomplish clinical predictive modeling tasks more quickly. ... improves the quality of the clinical predictive modeling work I do. ... makes it easier to perform clinical predictive modeling. ... enhances my effectiveness on clinical predictive modeling. ... increases my productivity.
Effort Expectancy	Perceived Ease of Use	... learning to operate the system would be easy for me. ... I would find it easy to get the system to do what I want it to do. ... my interaction with the system would be clear and understandable. ... I would find the system flexible to interact with. ... it would be easier for me to become skillful at using the system. ... I would find the system easy to use.
	Complexity	... using the system takes too much time from my normal duties. ... working with the system is so complicated, it is difficult to understand what is going on. ... using the system involves too much time doing mechanical operations (e.g. data input). ... it takes too long to learn how to use the system to make it worth the effort.
	Ease of Use	... my interaction with the system is clear and understandable. ... I believe that it is easy to get the system to do what I want it to do. ... overall, I believe the system is easy to use. ... learning to operate the system is easy for me.

Clinical Modeling Task

Here you will perform a pre-defined clinical modeling task. This task entails developing a clinical predictive model to ascertain the risk of chronic kidney disease based on an open source data set. Please use the tool provided to answer the items below.

Preparation

- **Use case description**: Chronic Kidney Disease (CKD) is defined as abnormalities of the kidney structure or function, present for at least 3 months, with clearly implications for the patients' health. These alterations are identified via imaging or biopsy studies, or inferred from laboratory tests and other clinical data.
- **Objective**: Train a model to predict the presence of CKD on the subjects given the available clinical information.
- **Target**: predict occurrence of CKD.

Dataset Selection

- For your cohort, please use the dataset provided `chronic_kidney_disease.csv`.
- Upload the available data in the modeling tool as per the tool's instructions

Predictor Handling

- Remove from the model the Patient ID (it should not be part of the model).

Model Generation

- Cohort: Chronic Kidney Disease.
- Use mean imputation for missing data.
- Retain 20% of the dataset for test (Test-split validation).
- Target: CKD.
- Use the algorithms: logistic regression, decision tree, random forest, gradient-boosting decision tree.

Model Evaluation

For evaluating the model, please obtain the following metrics (if your tool supports the given metrics, if not leave blank):

- Discrimination: area under the curve (AUC), precision, recall, F1-score
- Calibration: alpha calibration (intercept), beta calibration (slope)
- Clinical usefulness: treated, all, ADAPT

Model Interpretation

For evaluating the model, please obtain feature importance using the following methods:

- Global surrogate
- Local surrogate
- Model-based Feature Importance

MORPHER Clinical Modeling Task

This part is about using the MORPHER app while performing a pre-defined clinical modeling task. This task entails developing a clinical predictive model to ascertain the risk of chronic kidney disease based on an open source data set. Please use the tool provided to answer the items below.

Data Selection

1. What is the incidence rate of the outcome, i.e., percentage number of CKD cases?

   ☐ 0.5%
   ☐ 25%
   ☐ 50%
   ☐ Cannot answer

2. What is the number of predictors included in the analysis?

   ☐ 25
   ☐ 26
   ☐ 400
   ☐ Cannot answer

3. What is the percentage of cells with missing data?

   ☐ Less than 0.5%
   ☐ Less than 10%
   ☐ More than 25%
   ☐ More than 50%
   ☐ Cannot answer

4. What is the average age of the study participants?

   ☐ 10
   ☐ 20
   ☐ 100
   ☐ Cannot answer

5. Choose 3 features that you consider the most important a priori.

   _____
   _____
   _____

6. Indicate the 3 features with the most NA's

   _____
   _____
   _____

Dataset Handling

7. How was any missing data handled in the modeling process?

   ☐ No imputation
   ☐ Mean imputation
   ☐ K Nearest Neighbors
   ☐ Cannot answer

8. Why "Patient ID" needs to be deleted from the model?

   ☐ It is a cofounder feature.
   ☐ It is too relevant for the model. It will overperform.
   ☐ It is related with the age of the patient.
   ☐ Cannot answer

Model Generation

9. What were the modeling algorithms utilized?

☐ Logistic Regression
☐ Random Forest
☐ Decision Tree
☐ Gradient-Boosting Decision Tree
☐ Cannot answer

10. What outcome are the different models predicting?

☐ The appearance of AKD
☐ The appearance of CKD
☐ The relation between AKD and CKD
☐ Cannot answer

11. How was model performance evaluated?

☐ Internal validation
☐ External validation
☐ Cannot answer

12. What types of metrics were utilized?

☐ Discrimination
☐ Calibration
☐ Clinical Usefulness

Model Evaluation

Please fill out the table below according to the results you obtained for the above algorithms. Insert a hyphen "—" if you are not sure of the value or have not developed a model the given algorithm.

Algorithm	Metric	Value
Decision Tree	Precision	
	Recall	
	Area Under the Curve (AUC)	
Logistic Regression	Precision	
	Recall	
	Area Under the Curve (AUC)	
Random Forest	Precision	
	Recall	
	Area Under the Curve (AUC)	
Gradient-boosting Decision Tree	Precision	
	Recall	
	Area Under the Curve (AUC)	
Multilayer Perceptron	Precision	
	Recall	
	Area Under the Curve (AUC)	

13. Which model performs the best?

☐ Logistic Regression
☐ Random Forest
☐ Decision Tree
☐ Gradient-Boosting Decision Tree
☐ Cannot answer

14. Which model performs the worst?

☐ Logistic Regression
☐ Random Forest
☐ Decision Tree
☐ Gradient-Boosting Decision Tree
☐ Cannot answer

Model Interpretation

Please report the three most important features obtained by any of the models developed previously using the method "Model-based Feature Importance".

Algorithm	Top Feature (Top 1, 2, 3)	Importance
Decision Tree		
Logistic Regression		
Random Forest		
Gradient-boosting Decision Tree		
Multilayer Perceptron		

15. Are the features chosen in question # 5 among the top 3 after the interpretation has been made?

☐ Yes.
☐ No
Which:
_____
_____
_____

16. According to the different models, what are the 2 most important factors in the appearance of CKD?

_____
_____

User Activity

Please paste the content of your user activity during usage of the tool (Activity Monitor).

Additional questions

Please answer the following questions. They are based entirely in your opinion.

1. Which limitations did you find using this dataset?

   _____
   _____
   _____

2. Which limitations did you find using this tool?

   _____
   _____
   _____

3. Which limitations did you find with the information given in general?

   _____
   _____
   _____

4. BONUS: If you were creating a research paper, how would you call it?

   _____
   _____
   _____

Participant Information

Gender: ☐ female ☐ male ☐ other | Age: _____ ☐ prefer not to say

1. What is your profession?

   ☐ Research clinician / medical expert
   ☐ Data scientist / ML researcher
   ☐ Other: _____

2. Did your university or college studies cover the topic of clinical predictive modeling and/or epidemiology?

   ☐ Yes
   ☐ No

3. Have you ever performed clinical predictive modeling yourself or were (co-)author of a research article that included results of a clinical predictive model?

   ☐ Yes
   ☐ No

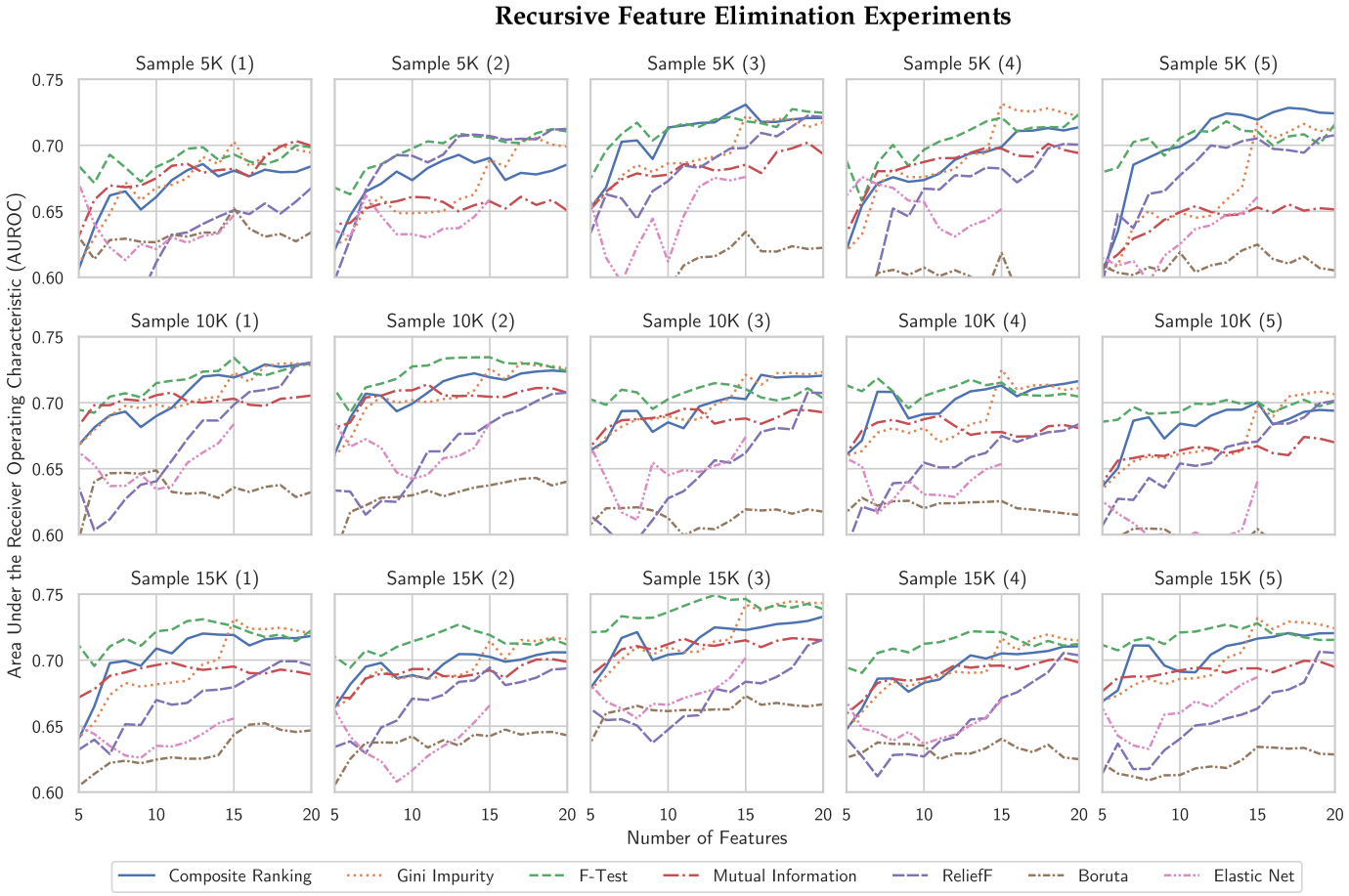
4. Have you ever interpreted the results of a clinical predictive model, including research publications?

   ☐ Yes
   ☐ No

5. Regarding your previous knowledge and exposure to programming, statistical packages, and clinical predictive modeling in general?

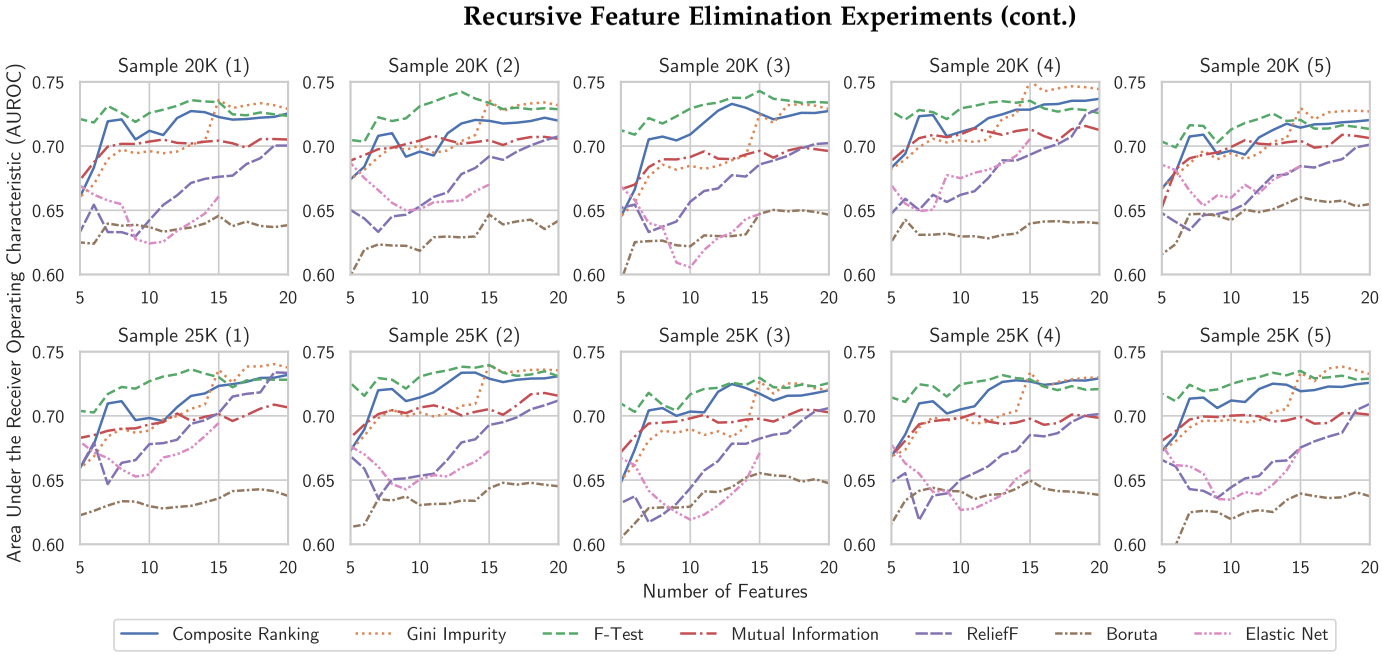
	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
I am experienced conducting and evaluating clinical predictive modeling studies.	☐	☐	☐	☐	☐
I am experienced using Python and/or R languages for machine learning (coding).	☐	☐	☐	☐	☐
I am experienced using packages such as SAS, SPSS or tools such as Weka and Orange	☐	☐	☐	☐	☐

A.3. Explanation-Driven Recursive Feature Elimination



(a) Experiments for generated samples with 5, 10, and 15 thousand patients.

Figure A.1.: Recursive feature elimination experiments conducted on the synthetic cohorts. Note that each sample size, an addition of 5 other cohorts was generated to reduce variance.



(b) Experiments for generated samples with 15 and 20 thousand patients.

Figure A.1.: Recursive feature elimination experiments conducted on the synthetic cohorts. Note that each sample size, an addition of 5 other cohorts was generated to reduce variance (cont.).

Bibliography

[1] HF da Cruz, B Bergner, O. Konak, F. Schneider, P. Bode, C. Lempert, and MP Schapranow. "MORPHER–A Platform to Support Modeling of Outcome and Risk Prediction in Health Research". In: *Proc. of the 19th IEEE Intl. Conf. on Bioinformatics and Bioengineering*. IEEE. 2019, pages 462–469.

[2] HF da Cruz, B Pfahringer, F Schneider, A Meyer, and MP Schapranow. "External Validation of a "Black-Box" Clinical Predictive Model in Nephrology: Can Interpretability Methods Help Illuminate Performance Differences?" In: *Proc. of the 17th Conf. on Artificial Intelligence in Medicine*. Springer. 2019, pages 191–201.

[3] HF da Cruz, F. Schneider, and MP Schapranow. "Prediction of Acute Kidney Injury in Cardiac Surgery Patients: Interpretation using Local Interpretable Model-agnostic Explanations". In: *Proc. of the 12th Intl. Conf. on Biomedical Engineering Systems and Technologies*. Volume 5. Prague, Czech Republic, 2019, pages 380–387. ISBN: 978-989-758-353-7.

[4] Ewout W. Steyerberg and Yvonne Vergouwe. "Towards Better Clinical Prediction Models: Seven Steps for Development and an ABCD for Validation". In: *European Heart Journal* 35.29 (2014), pages 1925–1931. ISSN: 15229645. arXiv: arXiv:1011.1669v3.

[5] J. C Wyatt and D. G Altman. "Commentary: Prognostic models: Clinically Useful or Quickly Forgotten?" In: *BMJ* 311.7019 (Dec. 1995), pages 1539–1541. ISSN: 0959-8138.

[6] Yong-Ho Lee, Heejung Bang, and Dae Jung Kim. "How to Establish Clinical Prediction Models". In: *Endocrinology and Metabolism* 31.1 (2016), page 38. ISSN: 2093-596X.

[7] Gary S. Collins et al. "Transparent Reporting of a Multivariable Prediction Model for Individual Prognosis or Diagnosis (TRIPOD): The TRIPOD Statement". In: *European Urology* 67.6 (2015), pages 1142–1151. ISSN: 18737560.

[8] William A Knaus et al. "APACHE II: A Severity of Disease Classification System." In: *Critical care medicine* 13.10 (1985), pages 818–829.

[9] Friedo W. Dekker, Chava L. Ramspek, and Merel van Diepen. "Con: Most Clinical Risk Scores are Useless". In: *Nephrology Dialysis Transplantation* 32.5 (May 2017), pages 752–755. ISSN: 0931-0509.

[10] Luke Eliot Hodgson et al. "Systematic Review of Prognostic Prediction Models for Acute Kidney Injury (AKI) in General Hospital Populations". In: *BMJ Open* 7.9 (2017), pages 1–10. ISSN: 20446055.

[11] Pablo Perel et al. "Systematic Review of Prognostic Models in Traumatic Brain Injury". In: *BMC Medical Informatics and Decision Making* 6 (2006), pages 1–10. ISSN: 14726947.

[12] Matthew M Churpek et al. "Multicenter Comparison of Machine Learning Methods and Conventional Regression for Predicting Clinical Deterioration on the Wards". In: *Crit Care Med* 44.2 (2016), pages 298–305.

[13] Zachary C. Lipton. "The Mythos of Model Interpretability". In: Whi (2016), pages 96–100. arXiv: 1606.03490.

[14] Alvin Rajkomar et al. "Scalable and Accurate Deep Learning for Electronic Health Records". In: *npj Digital Medicine* January (2018), pages 1–10. ISSN: 2398-6352. arXiv: 1801.07860.

[15] Patrick Hall and Navdeep Gill. *An Introduction to Machine Learning Interpretability: An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI*. O'Reilly, 2018. ISBN: 9781492033141. arXiv: 1702.00832.

[16] Jerome H Friedman. "On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality". In: *Data mining and knowledge discovery* 1.1 (1997), pages 55–77.

[17] Heba Abusamra. "A Comparative Study of Feature Selection and Classification Methods for Gene Expression Data of Glioma". In: *Procedia Computer Science* 23 (2013), pages 5–14.

[18] O'Neal, Jason and others. "Acute Kidney Injury Following Cardiac Surgery: Current Understanding and Future Directions." In: *Critical Care* 20.1 (2016), page 187.

[19] Brian H Nathanson and Thomas L Higgins. "An Introduction to Statistical Methods Used in Binary Outcome Modeling". In: *Seminars in cardiothoracic and vascular anesthesia* 12.3 (Sept. 2008), pages 153–66. ISSN: 1089-2532.

[20] Thomas Allweyer. *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. Books on Demand, 2016.

[21] Alison Callahan and Nigam H. Shah. "Machine Learning in Healthcare". In: *Key Advances in Clinical Informatics*. Elsevier, 2017, pages 279–291. ISBN: 9780128095232.

[22] K.Z. Mao. "Orthogonal Forward Selection and Backward Elimination Algorithms for Feature Subset Selection". In: *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 34.1 (Feb. 2004), pages 629–634. ISSN: 1083-4419.

[23] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13 (2012), pages 1–25.

[24] Henry de-Graft Acquah. "Comparison of Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) in Selection of an Asymmetric Price Relationships". In: *Journal of Development and Agricultural Economics* 21.1 (2010), pages 001–006. ISSN: 2006-9774.

[25] Jenna Burrell. "How the Machine Thinks: Understanding Opacity in Machine Learning Algorithms". In: *Big Data & Society* 3.1 (Jan. 2016). ISSN: 2053-9517.

[26] Gilmer Valdes et al. "MediBoost: A Patient Stratification Tool for Interpretable Decision Making in the Era of Precision Medicine". In: *Scientific Reports* 6.November (2016), pages 1–8. ISSN: 20452322.

[27] Greg Ridgeway. "The Pitfalls of Prediction". In: *National Insitute of Justice Journal* 271 (2013).

[28] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied Logistic Regression*. Volume 398. John Wiley & Sons, 2013.

[29] Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, 2012.

[30] Claude E Shannon. "A Mathematical Theory of Communication". In: *The Bell system technical journal* 27.3 (1948), pages 379–423.

[31] Wei-Yin Loh. "Classification and Regression Trees". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011), pages 14–23. ISSN: 19424787.

[32] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. 1st. O'Reilly Media, Inc., 2017. ISBN: 1491962291, 9781491962299.

[33] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of Statistics* (2001), pages 1189–1232.

[34] Si Si et al. "Gradient Boosted Decision Trees for High Dimensional Sparse Output". In: *International conference on machine learning*. 2017.

[35] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Volume 1. Springer, 2009, pages 337–387. ISBN: 9780387848570.

[36] Ewout W Steyerberg et al. "Assessing the Performance of Prediction Models : A Framework for Some Traditional and Novel Measures". In: *Epidemiology* 21.1 (2010), pages 128–138. ISSN: 1531-5487.

[37] Abdul Ghaaliq Lalkhen and Anthony McCluskey. "Clinical Tests: Sensitivity and Specificity". In: *Continuing Education in Anaesthesia Critical Care & Pain* 8.6 (Dec. 2008), pages 221–223. ISSN: 1743-1816.

[38] Afina S. Glas et al. "The Diagnostic Odds Ratio: A Single Indicator of Test Performance". In: *Journal of Clinical Epidemiology* 56.11 (2003), pages 1129–1135. ISSN: 8954356.

[39] Ana-Maria Šimundić. "Measures of Diagnostic Accuracy: Basic Definitions". In: *Ejifcc* 19.4 (2009), page 203.

[40] Rajul Parikh et al. "Understanding and Using Sensitivity, Specificity and Predictive Values". In: *Indian journal of Ophthalmology* 56.1 (2008), page 45.

[41] Steven Tenny and Mary R Hoffman. "Prevalence". In: *StatPearls [Internet]*. StatPearls Publishing, 2019.

[42] Robi Polikar. "Ensemble Learning". In: *Ensemble Machine Learning*. Springer, 2012, pages 1–34.

[43] Kaspar Rufibach. "Use of Brier Score to Assess Binary Predictions". In: *Journal of Clinical Epidemiology* 63.8 (2010), pages 938–939.

[44] John Platt et al. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers* 10.3 (1999), pages 61–74.

[45] F. Pedregosa and others. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pages 2825–2830.

[46] Ben Van Calster et al. "Reporting and Interpreting Decision Curve Analysis: A Guide for Investigators". In: *European Urology* 74.6 (2018), pages 796–804. ISSN: 18737560.

[47] John Harty. "Prevention and Management of Acute Kidney Injury". In: *The Ulster medical journal* 83.3 (2014), page 149.

[48] Vicker Andrew J and Elkin Elena B. "Decision Curve Analysis: A Novel Method for Evaluating Prediction Models". In: *Med Decis Making* 26.6 (2008), pages 565–574. ISSN: 0272989X. arXiv: 1305.0820.

[49] Shi-Tao Yeh et al. "Using Trapezoidal Rule for the Area Under a Curve Calculation". In: *Proceedings of the 27th Annual SAS User Group International* (2002).

[50] Zachary C Lipton. "The Mythos of Model Interpretability". In: *Queue* 16.3 (2018), pages 31–57.

[51] David Baehrens et al. "How to Explain Individual Classification Decisions". In: *Mach Learning Res* 11 (2010), pages 1803–1831.

[52] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proc. 22nd ACM SIGKDD*. San Francisco, California, USA, 2016, pages 1135–1144. ISBN: 978-1-4503-4232-2.

[53] Gajendra Jung Katuwal and Robert Chen. "Machine Learning Model Interpretability for Precision Medicine". In: *ArXiv e-prints* (Oct. 2016). arXiv: 1610.09045.

[54] Dieter Hayn et al. "Plausibility of Individual Decisions from Random Forests in Clinical Predictive Modelling Applications". In: *Studies in Health Technoly and Informatics* (2017), pages 328–335.

[55] Finale Doshi-Velez and Been Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: *arXiv e-prints*, arXiv:1702.08608 (Feb. 2017), arXiv:1702.08608. arXiv: 1702.08608 [stat.ML].

[56] Donghee Shin and Yong Jin Park. "Role of Fairness, Accountability, and Transparency in Algorithmic Affordance". In: *Computers in Human Behavior* 98 (2019), pages 277–284.

[57] W. James Murdoch et al. "Interpretable Machine Learning: Definitions, Methods, and Applications". In: *arXiv e-prints*, arXiv:1901.04592 (Jan. 2019), arXiv:1901.04592. arXiv: 1901.04592 [stat.ML].

[58] Zhengping Che et al. "Interpretable Deep Models for ICU Outcome Prediction." In: *AMIA Symposium* 2016 (2016), pages 371–380. ISSN: 1942-597X.

[59] G Louppe et al. "Understanding Variable Importances in Forests of Randomized Trees". In: *Neural Information Processing Systems* (2013), pages 1–9. ISSN: 1098-6596.

[60] Lloyd S Shapley. "A Value for n-Person Games". In: *Contributions to the Theory of Games* 2.28 (1953), pages 307–317.

[61] Erik Štrumbelj and Igor Kononenko. "Explaining Prediction Models and Individual Predictions with Feature Contributions". In: *Knowledge and Information Systems* 41.3 (2014), pages 647–665.

[62] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. 2017, pages 4765–4774.

[63] Ravindra Khattree and Dayanand N Naik. *Applied Multivariate Statistics with SAS Software*. SAS Institute Inc., 2018.

[64] Norman H Nie, Dale H Bent, and C Hadlai Hull. *SPSS: Statistical Package for the Social Sciences*. Volume 227. McGraw-Hill New York, 1975.

[65] Lee C Adkins et al. *Using Stata for Principles of Econometrics*. Volume 5. Wiley New York, NY, 2008.

[66] Wendy L Martinez and Angel R Martinez. *Computational Statistics Handbook with MATLAB*. Chapman and Hall/CRC, 2007.

[67] John W Eaton, David Bateman, and Soren Hauberg. *GNU Octave Manual*. Network Theory Ltd. Bristol, UK, 2002.

[68] Darren Cook. *Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI*. "O'Reilly Media, Inc.", 2016.

[69] Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

[70] Nikhil Ketkar. "Introduction to Pytorch". In: *Deep Learning with Python*. Springer, 2017, pages 195–208.

[71] Christopher Gandrud. *Reproducible Research with R and RStudio*. Chapman and Hall/CRC, 2016.

[72] Mark Hall et al. "The WEKA Data Mining Software: an Update". In: *ACM SIGKDD Explorations* 11.1 (2009), pages 10–18.

[73] Janez Demvsar et al. "Orange: from Experimental Machine Learning to Interactive Data Mining". In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 2004, pages 537–539.

[74] Markus Hofmann and Ralf Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. CRC Press, 2013. ISBN: 9781482205497.

[75] Thomas Kluyver et al. "Jupyter Notebooks-a Publishing Format for Reproducible Computational Workflows." In: *ELPUB*. 2016, pages 87–90.

[76] Brian D Athey et al. "tranSMART: an Open-source and Community-driven Informatics and Data Sharing Platform for Clinical and Translational Research". In: *AMIA Summits on Translational Science Proceedings* 2013 (2013), page 6.

[77] Sigfried Gold et al. *ATLAS: Scientific Analyses on Standardized Observational Data*. `https://github.com/OHDSI/Atlas`. 2019.

[78] Jenna M Reps et al. "Design and Implementation of a Standardized Framework to Generate and Evaluate Patient-level Prediction Models using Observational Healthcare Data". In: *J Am Med Inform Assoc.* 25.8 (2018), pages 969–975. ISSN: 1067-5027.

[79] Gang Luo. "MLBCD: a Machine Learning Tool for Big Clinical Data". In: *Health Inf Sci Syst.* 3.1 (2015), page 3. ISSN: 2047-2501.

[80] Robert Chen et al. "ExplICU: A Web-based Visualization and Predictive Modeling Toolkit for Mortality in Intensive Care Patients". In: *Conf Proc IEEE Eng Med Biol Soc.* 2015-Novem (2015), pages 6830–6833. ISSN: 1557170X.

[81] Hasso Plattner, Christoph Meinel, and Larry Leifer. *Design Thinking Research*. Understanding Innovation. Springer, 2012. ISBN: 9783642216435.

[82] International Organization for Standardization. *ISO/IEC/IEEE 24765:2017 - Systems and Software Engineering – Vocabulary*. `https://www.iso.org/standard/71952.html` [retrieved: May 15, 2019]. Standard. Geneva, Switzerland, 2017. URL: `https://www.iso.org/standard/71952.html`.

[83] Jeff Patton and Peter Economy. *User Story Mapping: Discover the Whole Story, Build the Right Product*. " O'Reilly Media, Inc.", 2014.

[84] Avraham Leff and James T Rayfield. "Web-Application Development Using the Model/View/-Controller Design Pattern". In: *Proceedings 5th IEEE international enterprise distributed object computing conference*. IEEE. 2001, pages 118–127.

[85] Kelly D Lewis. "Web Single Sign-On Authentication Using SAML". In: *arXiv preprint arXiv:0909.2368* (2009).

[86] Hasso Plattner and Matthieu-P. Schapranow, editors. *High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine*. Springer-Verlag, 2014.

[87] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proc. 9th Python in Science Conference*. Edited by Stéfan van der Walt and Jarrod Millman. 2010, pages 51–56.

[88] A Rubinsteyn and S Feldman. *fancyimpute: A Variety of Matrix Completion and Imputation Algorithms Implemented in Python*. https://github.com/iskandr/fancyimpute Accessed: October, 2018. 2018. URL: `https://github.com/iskandr/fancyimpute` (visited on Sept. 27, 2018).

[89] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD í6. New York, NY, USA: Association for Computing Machinery, 2016, pages 785–794. ISBN: 9781450342322.

[90] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems 30*. Edited by I. Guyon et al. Curran Associates, Inc., 2017, pages 3146–3154.

[91] Carl Boettiger. "An Introduction to Docker for Reproducible Research". In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pages 71–79.

[92] George Hripcsak et al. "Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers". In: *Stud Health Technol Inform.* 216 (2015), page 574.

[93] Jonathan Chaffer and Karl Swedberg. *jQuery Reference Guide: A Comprehensive Exploration of the Popular JavaScript Library*. Packt Publishing, 2007. ISBN: 1847193811.

[94] Miguel Grinberg. *Flask Web Development: Developing Web Applications with Python*. 1st. O'Reilly Media, Inc., 2014. ISBN: 1449372627, 9781449372620.

[95] David Aguilar et al. *jsonpickle: Python Library for Serializing any Arbitrary Object Graph into JSON*. `https://github.com/jsonpickle/jsonpickle`. 2019.

[96] Rick Copeland. *Essential SQLAlchemy*. " O'Reilly Media, Inc.", 2008.

[97] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and Statistical Modeling with Python". In: *Proc. 9th Python in Science Conference*. Edited by Stefan van der Walt and Jarrod Millman. 2010, pages 92–96.

[98] Guillaume Lemaitre, Fernando Nogueira, and Christos K Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: *The Journal of Machine Learning Research* 18.1 (2017), pages 559–563.

[99] Zhongheng Zhang et al. "Decision Curve Analysis: A Technical Note". In: *Annals of Translational Medicine* 6.15 (2018), pages 308–308. ISSN: 23055839.

[100] J. D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3 (2007), pages 90–95.

[101] S Brugman. *Pandas-Profiling: Exploratory Data Analysis Reports in Python*. `https://github.com/pandas-profiling/pandas-profiling`. 2019.

[102] Josiah L Carlson. *Redis in Action*. Manning Publications Co., 2013.

[103] *Celery Distributed Task Queue*. `http://celeryproject.org/`. 2020.

[104] Xiangrui Meng et al. "Mllib: Machine Learning in Apache Spark". In: *The Journal of Machine Learning Research* 17.1 (2016), pages 1235–1241.

[105] P Szabó and J Galanda. "SageMath for Education and Research". In: *15th International Conference on Emerging eLearning Technologies and Applications*. IEEE. 2017, pages 1–4.

[106] Elisabeth Scheufele et al. "tranSMART: An Open Source Knowledge Management and High Content Data Analytics Platform". In: *Proc. AMIA Summit on Translational Science* 2014 (2014), pages 96–101. ISSN: 2153-4063.

[107] KP Suresh. "An Overview of Randomization Techniques: An Unbiased Assessment of Outcome in Clinical Research". In: *Journal of human reproductive sciences* 4.1 (2011), page 8.

[108] Stephen D Bay et al. "The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation". In: *ACM SIGKDD Explorations Newsletter* 2.2 (2000), pages 81–85.

[109] Jerry L Hintze and Ray D Nelson. "Violin Plots: A Box Plot-Density Trace Synergism". In: *The American Statistician* 52.2 (1998), pages 181–184.

[110] Asghar Ghasemi and Saleh Zahediasl. "Normality Tests for Statistical Analysis: A Guide for Non-statisticians". In: *International journal of endocrinology and metabolism* 10.2 (2012), page 486.

[111] Justin Collins et al. "Meaningful Analysis of Small Data Sets: A Clinicians Guide". In: *Clinical and Translational Research* 2 (2017), pages 16–19.

[112] Viswanath Venkatesh et al. "User Acceptance of Information Technology: Toward a Unified View". In: *MIS quarterly* (2003), pages 425–478.

[113] Claire Anderson. "Presenting and Evaluating Qualitative Research". In: *American journal of pharmaceutical education* 74.8 (2010).

[114] Jakob Nielsen and Thomas K Landauer. "A Mathematical Model of the Finding of Usability Problems". In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. 1993, pages 206–213.

[115] Richard M Heiberger, Naomi B Robbins, et al. "Design of Diverging Stacked Bar Charts for Likert Scales and Other Applications". In: *Journal of Statistical Software* 57.5 (2014), pages 1–32.

[116] Lee J Cronbach. "Coefficient Alpha and the Internal Structure of Tests". In: *Psychometrika* 16.3 (1951), pages 297–334.

[117] Sebastian Celis and David R Musicant. "Weka-Parallel: Machine Learning in Parallel". In: *Carleton College, CS TR*. Citeseer. 2002.

[118] Irit Dinur and Kobbi Nissim. "Revealing Information While Preserving Privacy". In: *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2003, pages 202–210.

[119] Nicolas Papernot et al. "Towards the Science of Security and Privacy in Machine Learning". In: *arXiv preprint arXiv:1611.03814* (2016).

[120] Cynthia Dwork, Aaron Roth, et al. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pages 211–407.

[121] Jonathan Badger. *InspectOMOP: Extraction of EHR Data from Relational Databases in the OHDSI OMOP Common Data Model (CDM)*. `https://github.com/jbadger3/inspectomop`. 2019.

[122] Waqas Hameed et al. "Does Courtesy Bias Affect How Clients Report on Objective and Subjective Measures of Family Planning Service Quality? A Comparison between Facility- and Home-Based Interviews". In: *Open access journal of contraception* 9 (2017), page 33.

[123] Qiang Yang et al. "Federated Machine Learning: Concept and Applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pages 1–19.

[124] Mitchell H. Rosner and Mark D. Okusa. "Acute Kidney Injury Associated with Cardiac Surgery". In: *Clin J Am Soc Nephrol* 1.1 (2006), pages 19–32.

[125] Shu Yi Ng et al. "Prediction of Acute Kidney Injury within 30 Days of Cardiac Surgery". In: *J Thorac Cardiovasc Surg* 147.6 (June 2014), 1875–1883.e1. ISSN: 0022-5223.

[126] Simon Sawhney et al. "Acute Kidney Injury - How Does Automated Detection Perform?" In: *Nephrol. Dial. Transplant.* 30.11 (2015), pages 1853–61. ISSN: 1460-2385.

[127] Alistair E W Johnson et al. "MIMIC-III, a Freely Accessible Critical Care Database". In: *Scientific data* 3 (2016). ISSN: 2052-4463.

[128] Benjamin Letham et al. "Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model". In: *Annals of Applied Statistics* 9.3 (2015), pages 1350–1371. ISSN: 19417330. arXiv: `arXiv:1511.01644v1`.

[129] Anja Haase-Fielitz, Michael Haase, and Prasad Devarajan. "Neutrophil gelatinase-associated lipocalin as a biomarker of acute kidney injury: a critical evaluation of current status". In: *Annals of clinical biochemistry* 51.3 (2014), pages 335–351.

[130] Charuhas V. Thakar et al. "A Clinical Score to Predict Acute Renal Failure after Cardiac Surgery". In: *J Am Soc Nephrol* 14.8 (Aug. 2004), pages 2176–7. ISSN: 1046-6673.

[131] H. Palomba et al. "Acute Kidney Injury Prediction following Elective Cardiac Surgery: AKICS Score". In: *Kidney International* 72.5 (Sept. 2007), pages 624–631. ISSN: 0085-2538.

[132] Rajendra H. Mehta and others. "Bedside Tool for Predicting the Risk of Postoperative Dialysis in Patients Undergoing Cardiac Surgery". In: *Circulation* 114.21 (2006), pages 2208–2216. ISSN: 97322.

[133] Sarah C Huen and Chirag R Parikh. "Predicting Acute Kidney Injury after Cardiac Surgery: a Systematic Review." In: *Ann Thorac Surg* 93.1 (Jan. 2012), pages 337–47. ISSN: 1552-6259.

[134] Paul Thottakkara et al. "Application of Machine Learning Techniques to High-Dimensional Clinical Data to Forecast Postoperative Complications". In: *PLoS ONE* 11.5 (2016), pages 1–19. ISSN: 19326203.

[135] Matthieu Legrand et al. "Incidence, Risk Factors and Prediction of Post-operative Acute Kidney Injury Following Cardiac Surgery for Active Inefective Endocarditis: an Observational Study." In: *Crit. Care* 17.5 (Jan. 2013), R220. ISSN: 1466-609X.

[136] J Van Eyck et al. "Data Mining Techniques for Predicting Acute Kidney Injury after Elective Cardiac Surgery". In: *Crit. Care* 16.Suppl 1 (2012), P344.

[137] Rohit J. Kate et al. "Prediction and Detection Models for Acute Kidney Injury in Hospitalized Older Adults". In: *BMC Med. Inform. Decis. Mak.* 16.1 (2016), page 39. ISSN: 1472-6947.

[138] Marine Flechet et al. "AKIpredictor, an on-line Prognostic Calculator for Acute Kidney Injury in Adult Critically Ill Patients". In: *Intensive Care Medicine* 43.6 (2017), pages 764–773. ISSN: 14321238.

[139] Hyung-Chul Lee et al. "Derivation and Validation of Machine Learning Approaches to Predict Acute Kidney Injury after Cardiac Surgery". In: *J. Clin. Med.* 7.10 (2018), page 322. ISSN: 2077-0383.

[140] Nenad Tomasev et al. "A Clinically Applicable Approach to Continuous Prediction of Future Acute Kidney Injury". In: *Nature* 572.7767 (2019), pages 116–119.

[141] Duminda N. Wijeysundera et al. "Derivation and Validation of a Simplified Predictive Index for Renal Replacement Therapy After Cardiac Surgery". In: *JAMA* 297.16 (Apr. 2007), page 1801. ISSN: 0098-7484.

[142] Wuhua Jiang et al. "Validation of Four Prediction Scores for Cardiac Surgery-Associated Acute Kidney Injury in Chinese Patients". In: *Braz J Cardiovasc Surg* 32.6 (2017), pages 481–486. ISSN: 1027638.

[143] Karel G M Moons et al. "Prognosis and Prognostic Research: Application and Impact of Prognostic Models in Clinical Practice". In: *Brit. Med. J.* 338 (June 2009), b606. ISSN: 1756-1833.

[144] Rich Caruana et al. "Intelligible Models for HealthCare". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pages 1721–1730.

[145] Bryce Goodman and Seth Flaxman. "European Union Regulations on Algorithmic Decision-making and a "Right to Explanation"". In: *arXiv preprint arXiv:1606.08813* (2016), pages 1–9. ISSN: 0738-4602. arXiv: 1606.08813.

[146] Jose Antonio Lopes and Sofia Jorge. "The RIFLE and AKIN Classifications for Acute Kidney Injury: a Critical and Comprehensive Review". In: *Clin Kidney J* 6.1 (2013), pages 8–14.

[147] Anne Elixhauser et al. "Comorbidity Measures for Use with Administrative Data". In: *Medical Care* 36.1 (1998), pages 8–27. ISSN: 257079.

[148] Pablo A Estévez et al. "Normalized Mutual Information Feature Selection". In: *IEEE Transactions on neural networks* 20.2 (2009), pages 189–201.

[149] O'Neal, Jason and others. "Acute Kidney Injury Following Cardiac Surgery: Current Understanding and Future Directions." In: *Crit. Care* 20.1 (2016), page 187.

[150] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[151] Steven R Austin et al. "Why Summary Comorbidity Measures such as the Charlson Comorbidity Index and Elixhauser Score Work". In: *Medical care* 53.9 (2015), e65.

[152] Jeffrey Lebensburger et al. "Protective Role of Hemoglobin and Fetal Hemoglobin in Early Kidney Disease for Children with Sickle Cell Anemia". In: *American journal of hematology* 86.5 (2011), pages 430–432.

[153] Anuja Mittalhenkle et al. "Cardiovascular Risk Factors and Incident Acute Renal Failure in Older Adults: The Cardiovascular Health Study". In: *Clinical Journal of the American Society of Nephrology* 3.2 (2008), pages 450–456.

[154] Zhongheng Zhang. "Too Much Covariates in a Multivariable Model May Cause the Problem of Overfitting". In: *Journal of thoracic disease* 6.9 (2014), E196.

[155] Jessica Lin et al. "Experiencing SAX: A Novel Symbolic Representation of Time Series". In: *Data Mining and knowledge discovery* 15.2 (2007), pages 107–144.

[156] Zachary C Lipton et al. "Learning to Diagnose with LSTM Recurrent Neural Networks". In: *arXiv preprint arXiv:1511.03677* (2015).

[157] David Alvarez-Melis and Tommi S. Jaakkola. "On the Robustness of Interpretability Methods". In: arXiv (2018). ISSN: 207683. arXiv: 1806.08049.

[158] DB Toll et al. "Validation, Updating and Impact of Clinical Prediction Rules: a Review". In: *J. Clin. Epidemiol.* 61 (2008).

[159] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models". In: *arXiv e-prints*, arXiv:1802.01933 (Feb. 2018), arXiv:1802.01933. arXiv: 1802.01933 [cs.CY].

[160] Ting Li Su et al. "A Review of Statistical Updating Methods for Clinical Prediction Models". In: *Statistical Methods in Medical Research* 27.1 (2018), pages 185–197. ISSN: 14770334.

[161] Jana Hoffman et al. "Using Transfer Learning for Improved Mortality Prediction in a Data-Scarce Hospital Setting". In: *Biomedical Informatics Insights* 9.0 (2017). ISSN: 1178-2226.

[162] Gareth James et al. *An Introduction to Statistical Learning*. Volume 112. Springer, 2013.

[163] Noelia Sánchez-Maroño et al. "Filter Methods for Feature Selection – A Comparative Study". In: *Intelligent Data Engineering and Automated Learning - IDEAL 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pages 178–187. ISBN: 978-3-540-77226-2.

[164] Baofeng Guo and Mark S Nixon. "Gait Feature Subset Selection by Mutual Information". In: *IEEE Transactions on Systems, MAN, and Cybernetics-Part a: Systems and Humans* 39.1 (2008), pages 36–46.

[165] RPL Durgabai and Y Ravi Bhushan. "Feature Selection Using ReliefF Algorithm". In: *International Journal of Advanced Research in Computer and Communication Engineering* 3.10 (2014), pages 8215–8218.

[166] Hui Zou and Trevor Hastie. "Regularization and Variable Selection via the Elastic Net". In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2 (2005), pages 301–320.

[167] Ron Kohavi and Dan Sommerfield. "Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology." In: *KDD*. 1995, pages 192–197.

[168] Miron B Kursa, Aleksander Jankowski, and Witold R Rudnicki. "Boruta – A System for Feature Selection". In: *Fundamenta Informaticae* 101.4 (2010), pages 271–285.

[169] Isabelle Guyon et al. "Gene Selection for Cancer Classification Using Support Vector Machines". In: *Machine learning* 46.1-3 (2002), pages 389–422.

[170] Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. "The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures". In: *PloS one* 6.12 (2011), e28210.

[171] Jianping Hua, Waibhav D Tembe, and Edward R Dougherty. "Performance of Feature-Selection Methods in the Classification of High-Dimension Data". In: *Pattern Recognition* 42.3 (2009), pages 409–424.

[172] Farideh Bagherzadeh-Khiabani et al. "A Tutorial on Variable Selection for Clinical Prediction Models: Feature Selection Methods in Data Mining Could Improve the Results". In: *Journal of Clinical Epidemiology* 71 (2016), pages 76–85.

[173] Qi Dong, Michael R Elliott, and Trivellore E Raghunathan. "A Nonparametric Method to Generate Synthetic Populations to Adjust for Complex Sampling Design Features". In: *Survey Methodology* 40.1 (2014), page 29.

[174] Noseong Park et al. "Data Synthesis Based on Generative Adversarial Networks". In: *arXiv preprint arXiv:1806.03384* (2018).

[175] Lei Xu and Kalyan Veeramachaneni. "Synthesizing Tabular Data using Generative Adversarial Networks". In: *arXiv preprint arXiv:1811.11264* (2018).

[176] Kenji Kira and Larry A Rendell. "A Practical Approach to Feature Selection". In: *Machine Learning Proceedings 1992*. Elsevier, 1992, pages 249–256.

[177] Ryan J Urbanowicz et al. "Benchmarking Relief-Based Feature Selection Methods for Bioinformatics Data Mining". In: *Journal of biomedical informatics* 85 (2018), pages 168–188.

[178] Daniel Homola et al. *boruta_py: Implementations of the Boruta all-relevant feature selection method*. `https://github.com/scikit-learn-contrib/boruta_py`. 2020.

[179] Malik Yousef et al. "Recursive Cluster Elimination (RCE) for Classification and Feature Selection from Gene Expression Data". In: *BMC Bioinformatics* 8.1 (2007), page 144.

[180] Jin Li, Maggie Tran, and Justy Siwabessy. "Selecting Optimal Random Forest Predictive Models: A Case Study on Predicting the Spatial Distribution of Seabed Hardness". In: *PloS one* 11.2 (2016), e0149089.

[181] Lisa Torrey and Jude Shavlik. "Transfer Learning". In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI global, 2010, pages 242–264.

[182] Viktor Losing, Barbara Hammer, and Heiko Wersing. "Incremental On-line Learning: A Review and. Comparison of State of the Art Algorithms". In: *Neurocomputing* 275 (2018), pages 1261–1274.

[183] Amir Saffari et al. "On-line Random Forests". In: *2009 IEEE 12th International Conference on Computer Vision Workshops*. IEEE. 2009, pages 1393–1400.

[184] Mohammed Khalilia et al. "Clinical Predictive Modeling Development and Deployment through FHIR Web Services". In: *AMIA Annual Symposium Proceedings*. Volume 2015. American Medical Informatics Association. 2015, page 717.

Colophon

LaTeXwas used to typeset this document, which was based on the `osm-thesis` template provided by the Operating Systems and Middleware (OSM) Group of the Hasso Plattner Institute. Most of the graphs and plots were developed using the chart libraries `matplotlib` and `Seaborn`. Bibliography by `biber`.