FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Die Erschaffung von Präzisionsrehabilitation

Visual Analytics um das Ergebnis von personalisierten Rehabilitationsprozessen vorherzusagen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Georg Bernold, BSc
Matrikelnummer 01325845

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Univ.Ass. Renata Raidou, MSc PhD

Wien, 30. April 2019

Georg Bernold	Eduard Gröller

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Establishing Precision Rehabilitation

Visual Analytics for predicting the outcome of personalized rehabilitation processes

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Georg Bernold, BSc
Registration Number 01325845

to the Faculty of Informatics

at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Univ.Ass. Renata Raidou, MSc PhD

Vienna, 30th April, 2019

Georg Bernold

Eduard Gröller

Erklärung zur Verfassung der Arbeit

Georg Bernold, BSc
Niederhofstraße 39/34, 1120 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. April 2019

Georg Bernold

Danksagung

Mein Dank gilt allen, die mich im Rahmen meiner Diplomarbeit unterstützt haben.

Besonderer Dank gebührt Univ.Ass. Renata Raidou, MSc PhD, die mir als wissenschaftliche Beraterin stets mit Rat und Tat zur Seite stand. Weiters möchte ich Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller danken, dem die Betreuung der Diplomarbeit oblag. Außerdem möchte ich mich bei meiner Familie und meiner Partnerin Lisa bedanken, die mich in allen Situationen unterstützt und ermutigt haben. Letztlich möchte ich mich bei allen VAMED Angestellten bedanken, die an den Interviews teilgenommen haben, mir ihre Expertise zur Verfügung stellten und mir Einblick in dieses faszinierende Themengebiet gaben.

Acknowledgements

I would like to thank everyone who supported me in my project and this thesis.

Special thanks are due to my advisor Univ.Ass. Renata Raidou, MSc PhD, who always helped me with suggestions and ideas. Furthermore, I would like to thank Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller who supervised this thesis. In addition, I would like to thank my family and of course my girlfriend Lisa, who encouraged and supported me in any situations. Finally, I would like to thank all VAMED employees who participated in the interviews, supported me with their expertise and gave me insight into this fascinating topic.

Kurzfassung

Präzisionsmedizin ist ein vielversprechender Ansatz, der darauf abzielt durch die Einbeziehung individueller Faktoren die Behandlung von Patienten zu verbessern. Der Einfluss unterschiedlicher Behandlungsstrategien muss analysiert und vorhergesagt werden, um Präzisionsmedizin erfolgreich in der Rehabilitation zu etablieren, was dann als Präzisionsrehabilitation bezeichnet wird. Dies erfordert den Einsatz großer und komplexer Datensätze. Mit aktuellen Methoden ist die Erforschung verfügbarer Daten fordernd und nur beschränkt möglich. Diese Diplomarbeit zielt darauf ab, diese Einschränkungen aufzuheben, indem neue Strategien der Visuellen Analyse eingesetzt werden, welche explorative Aufgaben der Benutzer, die in den Präzisionsrehabilitationsprozess eingebunden sind, zu unterstützen.

Der Hauptbeitrag dieser Diplomarbeit ist eine Anwendung der Visuellen Analyse namens *preha*, die es klinischen und technischen Experten ermöglicht, ihre Datenanalyseaufgaben im Rehabilitationskontext zu erfüllen. Die Applikation wurde in Kooperation mit dem österreichischen Rehabilitationsanbieter VAMED entworfen und erarbeitet. Flexible digitale Instrumententafeln mit einer Vielzahl an Visualisierungen bilden die Benutzerschnittstelle. Ein weiterer Forschungsbeitrag ist eine Aufgabenanalyse, die mit klinischen und technischen Experten durchgeführt wurde. Für unsere Analysen verwenden wir die Datensätze von 46,000 Patienten, die aus der elektronischen Patientenakte von VAMED erstellt wurden. Ein umfassendes Werkzeug wurde entwickelt, um die Rehabilitationsdaten aufzubereiten. Das Ergebnis der Rehabilitation kann mittels einer auf Random-Forest-Regression basierenden Anwendung des maschinellen Lernens vorhergesagt werden. Das Design und die Implementierung einer interaktiven Treemap für die verwendete Instrumententafel sind ebenfalls ein Forschungsbeitrag dieser Diplomarbeit. Durch Nutzungsszenarien und Evaluierungen kann gezeigt werden, dass Visuelle Analyse dazu geeignet ist, *Präzisionsrehabilitation zu erschaffen.*

Abstract

Precision medicine is a promising approach aiming to improve a patient's treatment by taking individual variability into account. Means to analyze and predict the impact of varying treatment strategies to subcohorts of patients are required in order to successfully introduce precision medicine into rehabilitation, i.e., precision rehabilitation. Large and complex data sets are the necessary foundation for this kind of analysis. As the existing means of analysis are limited, the exploration of available data is currently challenging. The thesis aims to address this issue by designing new Visual Analytics strategies that support the exploratory tasks of users involved in precision rehabilitation.

The main contribution of the diploma thesis is a Visual Analytics application called *preha* that allows clinical domain experts and engineers to fulfil their data analytics tasks in the rehabilitation context. This application was designed and implemented in cooperation with employees of the Austrian rehabilitation provider VAMED. Flexible interactive dashboards including a variety of visualizations are used as the front end. Additionally, a user task analysis accomplished through interview sessions with clinical and technical experts is presented. For the data analysis, we use a data set of 46,000 patients from VAMED's electronic health record. A comprehensive tool for preprocessing rehabilitation data, including cleaning non-validated health assessment scores, is provided. Rehabilitation outcome is predicted by utilizing a machine learning application based on random forest regression. Another contribution is the design and implementation of an interactive tree map visualization for the used dashboard. Usage scenarios and evaluation sessions are performed to prove the feasibility of the proposed solution, showing that Visual Analytics is capable of *establishing precision rehabilitation.*

Contents

CHAPTER 1

Introduction

Precision medicine is a promising approach aiming to improve a patients treatment by taking individual variability into account [Nat11]. This is in contrast to the traditional one-size-fits-all approach that is rather focused on the best possible treatment for the average person. This is the case across a wide range of medical disciplines, especially rehabilitation. While the average patient is usually well known when it comes to the treatment process, there is currently no intention to consider the individual patient. If there would be the possibility to determine individual patient treatment plans by deploying a tool that enables to perform analytics on electronic health record (EHR) data, the way would be paved towards what we are going to call *precision rehabilitation*.

1.1 Motivation and Problem Definition

The current situation is quite far from what precision rehabilitation proposes. A variety of challenges of different origin is still to be tackled. From the top level perspective, the legal framework does currently not support a publicly funded rehabilitation system with its focus on individual variability [KEP12]. Of course, physicians at the rehabilitation facilities may vary the therapy to a certain degree, but to take this step they need to be aware of the patients progress at any time of the treatment. Typically, current rehabilitation information systems (REIS) do not feature functions that enable monitoring of rehabilitation progress to this extent.

A possible way to address this issue would be the utilization of data analysis. Data analysis is increasing in medicine and all its branching disciplines. It enables better understanding of the patients—subsequently aiding the improvement of the quality of the patients' treatment. Different types of rehabilitation (cardiologic, orthopedic, ...) face the challenge of utilizing more and more data, as currently the rehabilitation process only depends on a minority of factors like primary diagnosis (the reason for the rehabilitation) or insurance. To develop new rehabilitation processes through precision rehabilitation,

the parameters responsible for the rehabilitation progress of the patients need to be identified. Based on these parameters, the outcome of the rehabilitation can be constantly predicted with live data from the REIS.

Large datasets are the necessary foundation for performing this kind of predictive analysis. The datasets rise three major challenges: First, to gain knowledge it is necessary to utilize large datasets, which require significant computational resources. Second, medical data is often high-dimensional data, meaning that a single patient is associated with a large number of properties in the dataset. Finally, the data is heterogeneous (dichotomous, numeric, scales), which adds another layer of complexity in terms of interpretation. The exploration of the available data is currently a challenge, as the existing means of analysis are limited. Visual Analytics (VA) addresses these issues by utilizing tools that combine *"automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision making on the basis of very large and complex datasets"*[KAF+08]. The aim of this thesis is to establish precision rehabilitation, through the development and employment of a Visual Analytics application.

1.2 Aim of the Work

The contribution of this thesis is the design and implementation of a VA tool that enables us to visually explore and analyse available rehabilitation data and the parameters affecting the rehabilitation process. We aim to answer the following research questions in the course of the thesis:

1. How can we identify the parameters that are responsible for a successful rehabilitation within a large set of high dimensional data? Which are the most relevant and important parameters for a successful rehabilitation?

2. How can VA aid data analysts to extend their knowledge about the rehabilitation outcome to improve the rehabilitation process as such?

3. How can the predicted and the actual outcome of the rehabilitation be visualized and analyzed comparatively?

The dataset used for the analysis in this thesis is provided by the Austrian company VAMED. A total of 46,000 cases is extracted from the EHRs of five rehabilitation centres. Orthopedic and neurological patients are featured in the dataset. For each case typical demographic features like age, sex, or residence of the patient, medical indicators like the primary diagnosis (the cause for the rehabilitation) and health assessment scores are recorded. Particular attention was paid to ensure the compliance of the data usage in the scope of this diploma thesis with the general data protection regulation of the European Union.

1.2.1 Steps and Challenges

We can split this work in several steps, each of them covering different challenges:

Identification of Data Sources and Parameters. The first focus of the thesis is the collection of data sources (electronic health records, treatment plans, health assessment scores, ...) to determine all possible factors that may affect the rehabilitation. Then, the data needs to be classified into types (numeric, scale, text, enumeration, ...) so the analysis technique is definable. In this step, it is necessary to extract those parameters that really affect the outcome of the rehabilitation. An appropriate way of treating data inconsistencies and missing values needs to be defined as by Alemzadeh et al. [ANI+17] or Souverein et al. [SAKM+16]. As part of the data is lacking input validation, data cleansing needs to be performed automatically prior to data processing within the Visual Analytics platform, as described by Gschwandtner et al. [GAM+14] or Tran et al. [THJ17].

Requirements for a VA Platform. Healthcare data analytics is a field that still takes a mixed level of technical and medical experience, which is to be defined in order to identify the key users. To design a proper application, requirements need to be obtained from the dedicated users and relevant literature like by Klemm et al. [KOJL+14, KLG+16]. Requirements engineering is realized as the result of qualitative interviews with the potential users. Additionally, a proper technology to build the platform needs to be identified. Furthermore, the requirements for the data need to be obtained, as high-dimensional, large, heterogeneous and often inconsistent data offers a specific level of complexity.

Build the VA Platform for the Identified Data and Requirements. The purpose of Visual Analytics in this context is to enable the discovery of new knowledge and to aid decision making. So the challenge in this step is to determine fitting visualizations and combinations of those for Visual Analytics. A proper way to integrate the rehabilitation outcome prediction is to be determined as by Stein et al. [SBS+15] or Singh et al. [SNG+15]. A-priori knowledge of the dedicated users must be considered as much as the quantitative and qualitative complexity of the underlying data. The platform must enable the user to explore the data fast, interactively and intuitively, and obtain knowledge without major interruptions of the thinking process [CRM91, Shn94].

Verify and Validate the VA Platform with Users. Finally it is necessary to test the value of the implemented application. Evaluation principles as described by Munzner [Mun09] or Lam et al. [LBI+12] are considered. The users being evaluated are the users the system was designed for and users that are new to the field of data analytics.

1.3 Methodological Approach and Contributions

A thorough analysis of the dataset used for our research is needed. Close attention is paid to the size of the dataset and possible constraints resulting from this. We classify all features of the dataset by means of the statistical scales of measurement [Ste46]. For each feature data quality and consistency are considered. A data preprocessing strategy is developed in order to provide the desired data quality for our research.

An identification of potential users from the multidisciplinary variety of professions in rehabilitation is performed. Two professions represent the potential users of our application. Clinicians (domain experts) are deeply involved in the total rehabilitation process, they develop the clinical intervention strategy and have ultimate responsibility on the rehabilitation process [GFI+16]. Parts of the IT staff (engineers) are responsible for managing the data and the clinical applications. We perform a user task analysis to determine the user requirements for our application. As a result of individual interview sessions, we introduce nine tasks the users aim to perform. Additional requirements are identified by analyzing related literature from medicine, visualization, and data science.

In order to address the diverse requirements of our users, data, tasks and Visual Analytics itself we introduce our application *preha*. The heart of preha is a Visual Analytics dashboard based on the software kibana that provides the flexibility to power all proposed tasks. A machine learning module allows the user the prediction of rehabilitation outcome values by utilizing a random forest regression algorithm. A preprocessing module performs all necessary steps to transform the original EHR data to well structured, clean, and complete data.

Preha is evaluated by potential users that perform fictional assignments that are based on the initial tasks. The outcome of the evaluation is used as the basis for a critical reflection of our work. Issues with the proposed solution are provided, as well as potential future improvements.

Several contributions are provided by this diploma thesis. A Visual Analytics application that allows domain experts and engineers to fulfil their data analytics tasks in the rehabilitation context is the main contribution. A thorough user task analysis is performed with potential users from the rehabilitation field. We present a collection of techniques and algorithms for the proposed problem in the visualization design chapter. A comprehensive tool for preprocessing rehabilitation data is introduced. This tool allows for cleaning non validated health assessment scores. Another contribution is the design and implementation of an interactive tree map visualization for kibana.

1.4 Structure of the Thesis

In the following Chapter 2, we will focus on the clinical background of rehabilitation, outcome measurement and the data, users and tasks providing the scope of this thesis. The state of the art, related work, and existing approaches in the fields of Visual Analytics

and predictive analytics in rehabilitation will be discussed in Chapter 3. Subsequently, we will describe the process of the user task analysis and present the design of our application in Chapter 4. The implementation details including selected technologies, programming languages and patterns will be elaborated on in Chapter 5. Chapter 6 introduces the results of our diploma thesis. Usage scenarios as well as a user evaluation are provided for the corresponding tasks. Finally in Chapter 7 we summarize our approach with regards to the stated research questions. Limitations of our work and possible future work are presented here.

CHAPTER 2

Clinical Background

The aim of this chapter is to provide the necessary background information on scientific basics around the focus of this thesis. As the thesis is based in the field of medical informatics, we intend to impart knowledge from medicine and computer science. Our first focus is rehabilitation medicine as such, including basic terminology and concepts. We will particularly focus on the Austrian rehabilitation system and its characteristics (e.g., in-patient rehabilitation). We present the topic of rehabilitation outcome measurement and its alignment in the context of rehabilitation. The scope conditions of this diploma thesis are introduced based on the following three characteristics: data, users, and tasks. Finally we introduce precision rehabilitation in the context of this thesis.

2.1 Precision Medicine

Precision medicine is an approach in medicine that describes the treatment of patients based on their individual variability. Precision medicine covers *"prevention and treatment strategies that take individual variability into account"* [CV15]. Even though this definition does not limit precision medicine to a specific medical discipline, it is often used in the context of drug research or cancer treatment [Nat11].

The idea behind precision medicine is not new. Collins et al. [CV15] mention blood typing, the fact that blood transfusions can only be performed between persons with compatible blood groups. This concept has been developed more than a century ago by the Austrian physician Karl Landsteiner [Lan00]. Despite the fact that the idea behind precision medicine is not novel, it re-appeared due to the latest progress made in the technical domain. Three main aspects are to be considered here. The first is the development of large-scale biologic databases, second, the powerful methods for characterizing patients (e.g., mobile health technology) and the third are computational tools for analyzing large datasets [CV15]. Data from electronic health records is also a

big advantage to precision medicine, as they provide the opportunity to integrate health care information and biological data [Nat11].

A few years before precision medicine, the term personalized medicine was used to describe this principle in the literature [HC10]. An important aspect that needs to be considered is the fact that the word "personalized" seemingly refers to an individual person. Medicine does not aim to develop treatment strategies for billions of people on this planet. What is needed is an optimal trade-off between the most precise and economically justifiable treatment strategies, to define patient subcohorts that get the most efficient treatment. The "Committee on A Framework for Developing a New Taxonomy of Disease" suggests to use the term precision medicine, as it is less likely to be misinterpreted as literally personalized treatment [Nat11].

The focus of precision medicine is based on two main components. A near-term focus is on cancer and a long-term aim to generate new knowledge for other diseases. Research for other diseases highly depends on data, as only large cohort studies enable the identification of subcohorts for precision medicine. There are approaches aiming to include more patients in cohort studies, subsequently improving participants access to information about their health and ongoing research [CV15].

2.2 Precision Rehabilitation

2.2.1 Rehabilitation in Austria

According to the World Health Organization [Wor11], rehabilitation is a medical discipline that contains *"set of measures that assist individuals who experience, or are likely to experience, disability to achieve and maintain optimal functioning in interaction with their environments"*. Medical rehabilitation is based on a holistic approach defining a human as an active part of society, according to Engel's biopsychosocial model [Eng77]. The main goal of rehabilitation is to enable the patients to actively re-participate in their life, regardless of the origin of their disease [HH13]. From an economic perspective it is desirable to enable the patient to practice a profession, pursue an education or at least prevent or delay retirement and need for care [GFI$^+$16].

Classifications

The boundaries of rehabilitation are not clearly defined. It is related to acute-medical treatment and covers medical, social, and job-related measures. Rehabilitation covers aspects of curative and physical medicine. In Austria, doctors represent the profession responsible for rehabilitation measures. The rehabilitation process itself is classified in the literature [KEP12, GFI$^+$16] as follows:

Phase I The early-mobilisation in acute care hospitals is the core of this Phase. This Phase mainly focuses on physical measures, but also includes ergotherapy, psychotherapy, or logopedics, depending on a patient's needs. Alternatively,

this Phase is called acute-rehabilitation, as the patients rely on medical treatment at this point.

Phase II This Phase is located in facilities that qualify as special clinics according to the legal regulations of the KAKuG (Austrian law for hospitals and sanatoriums) §2 (1) [Ö18b]. The major treatment in this Phase is in-patient rehabilitation. This is handled either subsequently ("Anschlussheilverfahren", further referred to as subsequent therapy) or up to 12 weeks after hospital stays or any other treatment related to rehabilitation ("Rehab-Heilverfahren", further referred to as rehabilitation therapy). The main requirement for this Phase is a stable state of the patient. Under specific circumstances, Phase II can be set in out-patient clinics.

Phase III The aim of out-patient Phase III is to stabilize the progress reached by rehabilitation in Phase II. Further progress of the existing disease shall be prevented, as shall the establishment of new diseases.

Phase IV Long-term rehabilitation to further stabilize the patient's state is the idea of this Phase. Phase IV is characterized by self-responsibility of the patient.

Neurological rehabilitation is based on a different classification from A-E, created by the Austrian society of neurological rehabilitation, as described in the literature [KEP12, GFI+16]:

Phase A This Phase describes an acute neurological disease or acute aggravations of such a condition. Intensive medical care for the patient is needed in this Phase.

Phase B Patients in this Phase are treated as in-patient in acute-neurological-rehabilitation wards. The patient is unable to cooperate in the treatment or perform activities of daily life.

Phase C In this Phase, the patients are able to cooperate in their treatment. The patients are conscious and able to participate in therapeutic measures up to three hours a day.

Phase D The patients have the ability to perform tasks of daily life independently or with the help of therapeutic appliances. They are able to participate in therapies for several hours a day.

Phase E The final Phase of this model requires the patients to be able to organize and spend several days without any help. They may need focused special neuro-rehabilitation measures to further strengthen therapy success.

Phases	
General	Neurology
Phase I	Phase A
	Phase B
Phase II	Phase C
	Phase D
Phase III	Phase E
Phase IV	-

Table 2.1. Correspondence table for the general and neurological rehabilitation Phase models. [GFI+16]

These two models correspond to each other [GFI+16] as described in Table 2.1.

Legal Basis

In Austria the extent of rehabilitation measures, the responsibility and criteria for claims are regulated in the general social insurance law (Allgemeines Sozialversicherungsgestz, ASVG) [Ö18a]. The institution responsible for the rehabilitation is further referred to as payer. The responsibility may be distinguished as follows [Ö18a]:

(a) If the reason for the disability is a work accident or an occupational disease, the accident insurance is responsible for the rehabilitation (§189 ASVG).

(b) If the reason for the disability is caused by psychatric, mental, or physical illness, the federal pension fund (PVA) is in charge (§§ 300-307c ASVG).

(c) Additional rehabilitation services are in the responsibility of the health insurance (§ 154a ASVG).

This thesis only takes rehabilitation measures in responsibility of the federal pension fund into account. According to the yearly report of the federal pension fund [Pen17], € 993,04 Mio. were spent on rehabilitation or preventive health care in 2017. The same year, 55.126 subsequent therapy and 68.663 rehabilitation therapy requests have been approved by the federal pension fund. The focus of this thesis lies on in-patient rehabilitation, which corresponds to Phase II of the general rehabilitation Phase model or Phase C and D of the neurological model (compare Table 2.1). Under specific circumstances, out-patient rehabilitation is also suitable for patients in these phases [KEP12], however we consider it out of scope for better comparability of the rehabilitation processes. Usually, in-patient rehabilitation is scheduled directly or within 12 weeks after a hospital stay [KEP12].

2.2.2 Rehabilitation Outcome Measurement

In order to constantly advance the effectiveness of medical treatment, there is need to improve the interventions applied to a patient. Furthermore, to make the effect of

this interventions visible, outcome (or status) measures have been defined as a tool of evidence-based medicine [Sto11, EJP13, LK14]. These measures are often referred to as "scores". Donovan et al. [DFE93] divided health outcome measures in six categories:

1. **General health measures**, to measure well-being, social health, emotional health (e.g., the Nottingham Health Profile [HMM$^+$81])
2. **Measures of physical function**, to measure level of physical impairment and disability (e.g., the Functional Independence Measure [KGHS87])
3. **Pain measures**, to reflect duration and intensity of pain (e.g., the Visual Analogue Scale [Sto11, EJP13])
4. **Social health measures**, to assess social support mechanisms and networks (e.g., the Social Health Battery [DFE93])
5. **Quality of life measures**, to measure the satisfaction of individuals with life (e.g., the Quality of Life index [FP85])
6. **Specific disease measures**, to assess issues for particular patient groups (e.g., the Oswestry low back pain questionnaire [FCDO80])

In order to understand the role of outcome measurement in the context of rehabilitation, we explain it in the context of the in-patient rehabilitation pathway. A graphical overview of the pathway created by Wade [Wad99] can be seen in Figure 2.1. First, the patient is referred to a rehabilitation facility, where his or her initial state is recorded. This covers the collection of demographic (e.g., age) and medical (e.g., state of the disability) information. Here is where the iterative rehabilitation process begins:

At the initial stage of each sequence, an *assessment*, that aims to quantify the present problems in an objective way [EJP13] is performed. This measurement is a standardized method for assessing those problems. The better the constructs used to perform the measurements, the more valid the measurement becomes, producing more reliable results. Measures with a high degree of confidence are referred to as standardized outcome measures. While some measures are easy to describe by objective measurement (e.g. physical items), abstract concepts (e.g. pain) are harder to observe. To overcome this issue, indicators that enable quantification are created (e.g. the Visual Analogue Scale [Sto11, EJP13] for pain measurement).

Based on the assessment, a rehabilitation *goal* is determined. This goal shall be a measurable state, which is checked at every reassessment. An interdisciplinary team including doctors, nurses, and therapists adapts the patients treatment plan according to the insights gained up to this point. It is worth mentioning, that these adaptions are only to be made in a limited range, as the *intervention* strategy is mainly determined by the payer of the rehabilitation. At this point, the planned intervention is carried out and the patient works through his or her therapy plan as scheduled. At the final stage

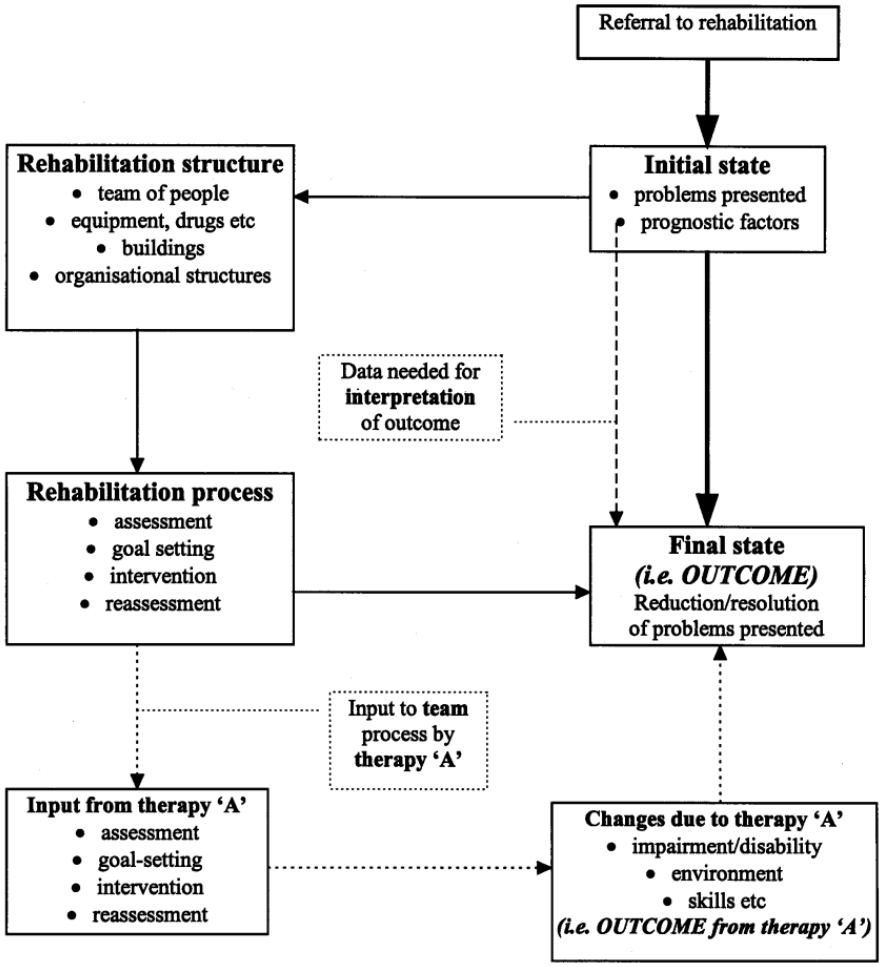

Figure 2.1. The in-patient rehabilitation pathway [Wad99].

of each iteration the reassessment is performed. Now the same measures as in all other assessments are applied, in order to achieve comparability across all assessments.

Again, at the time of discharge the reduction of the problems are presented by interpreting the trend that the measurements show over time. The final state of the patient at this point is called the *outcome* of the rehabilitation. Even though the process above is described iteratively, the frequency of the performed measurements differs in reality [KMH01, SO08]. Some measurements are only taken at admission, in order to set up the intervention strategy. At least two measures are needed to objectively measure the rehabilitation progress. Usually, they are performed at the time of admission and at the time of discharge.

Optionally, rehabilitation outcome measurement may also be used to determine the effect of individual therapies. As shown at the bottom of Figure 2.1. This aspect of outcome measurement is of major interest for all stakeholders in the rehabilitation process, as it

provides feedback on how individual therapies influenced the rehabilitation outcome of the patient. Unfortunately, the practicability of such instruments is limited as the global outcome does not reflect the input of particular professions. The global outcome is rather an integral result of teamwork [Wad99]. Predictions on the rehabilitation progress can be made utilizing outcome measurements [NTC+16].

A further approach towards standardized outcome measures worth mentioning is the ICF classification developed by the World Health Organization [O+01]. This classification framework has a focus on functioning, opposed to ICD-10 [Wor04], which is focused on diseases. For this reason it is considered a suitable component of outcome measures [KCS+13, Sto11, EJP13], by enabling a standardized encoding of their results. A current challenge is linking the encoding of the results of state-of-the-art standardized outcome measures to ICF encodings [EJP13]. ICF even features specific core sets that are tailored to the needs of specific patients or disease groups, like stroke, breast cancer, or rheumatoid arthritis. Even though there are standardized outcome measures that are suitable to achieve comparability for certain cases, ICF aims to be an international classification language which enables international comparison [Sto11]. Another positive aspect of ICF is the enhancement of communication among therapists and with their managers [EJP13].

Up to this point, rehabilitation outcome measurements were discussed as a process that is in the responsibility of the rehabilitation facility. Typically the measurement is performed by a clinician, we refer to this as Clinician Reported Outcome Measure (CROM) [LK14]. While CROMs are highly valuable and valid, there are circumstances under which their extensive use is limited, e.g., by lack of time [JHI+09]. Still it is desirable to maximize the number of measurements in order to monitor the rehabilitation progress. For this reason, Patient Reported Outcome Measures (PROM) [LK14] are used. A typical example for a PROM is a questionnaire filled by the patient before, during, and after the rehabilitation.

2.2.3 Scope Conditions

Drafting a solution that enables precision rehabilitation by visualizing large and complex datasets is quite challenging. Transforming this problem to Visual Analytics is tough, as plenty of interactive visualization and automated data analysis techniques are available. Choosing the appropriate technique is a hard decision to make. In order to support these design decisions, Miksch et al. introduced the so called Data–Users–Tasks Design Triangle [MA14]. This triangle is a high-level framework that aims to support the design process for researchers and practitioners in a simple and effective way. In this section we will describe the three aspects of the triangle in the scope of this diploma thesis. A graphical representation of the triangle can be seen in Figure 2.2. The vertices of the triangle describe the three domains to be considered in the design of the VA application. The edges between two points describe the relation of the domains. *Expressiveness* refers to visualizing the information content of the data, neither more nor less. *Effectiveness* takes the cognitive capabilities of the human and contextual information into account

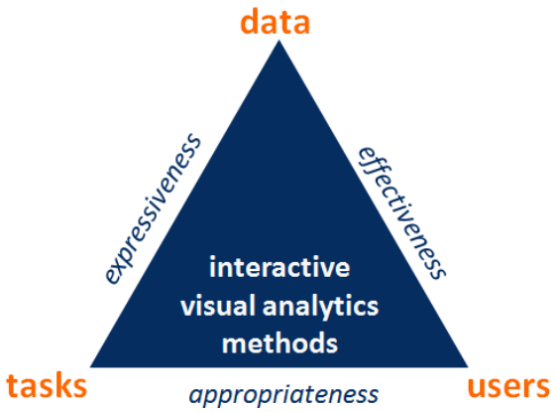

Figure 2.2. The Data–Users–Tasks Design Triangle [MA14].

to achieve the best possible visualizations. *Appropriateness* grades the benefit of the visualization process in relation to the given tasks, to achieve a cost-value ratio.

Data

Large and complex datasets are the major reason why Visual Analytics was introduced [WT04, KAF+08]. They are also at the core of this diploma thesis. We are facing the challenge of analyzing electronic health records and other clinical data sources, such as treatment plans. The very dataset used for this diploma thesis is provided by five rehabilitation facilities, operated by the Austrian company VAMED. Of those five rehabilitation facilities, two have an orthopedic department, one has a neurological department, and two have both. The data was collected over a time span of up to six years, from 2013 to 2019. A total number of 65,000 individual stays are featured in the dataset, 46,000 of which are patients. The remaining part is accompanying persons such as spouses. For each case, typical demographic features like age, sex, or residence of the patient have been collected beside medical indicators like the primary diagnosis (the cause for the rehabilitation) and scores, which were already introduced in subsection 2.2.2.

Every single feature in this dataset corresponds to a measurement scale and a data type. A measurement scale determines the appropriate statistical procedure for analyzing particular data and drawing conclusions from that data. The following terms are used to describe statistical scales of measurement [Ste46] (a summary can be found in Table 2.2):

1. **Nominal**: A nominal (or categorical variable) can be placed in categories. It is not possible to do calculations with nominals, as it does not represent a numeric value. Nominals can not be put in order. Examples for nominals in our dataset include health insurance corporation or sex. A dichotomous nominal is special in that it may only have two categories (e.g., smoker/non-smoker).
2. **Ordinal**: An ordinal represents values that can be put in order. It is used for ordering observations with corresponding ties, as the measurement sensitivity is not sufficient, e.g., data observed in a questionnaire with the Likert scale [Sto11].

3. **Interval Scale**: An interval scale is used to represent ordered numbers with meaningful divisions into a fixed and defined interval. If a score (e.g., the Barthel index [KEP12]) has a scale of 100, a difference of 10 points between 70 and 80 means the same as between 50 and 60.

4. **Ratio**: A ratio is an interval scale, with the major difference that the zero value is meaningful. Examples for ratios are height and weight.

	Nominal	Ordinal	Interval Scale	Ratio
Named Variables	✓	✓	✓	✓
Ordered Variables		✓	✓	✓
Fixed Interval between Variables			✓	✓
Meaningful Zero Values				✓

Table 2.2. Summary of the statistical scales of measurement.

In order to highlight the key component of this topic—the data complexity—an example dataset of artificial but realistic values, which can be seen in Table 2.3, will be described. Please note, that the structure of the table does not reflect the real structure of the data in VAMED's EHR. Besides typical demographic parameters like age, sex, or residence of the patient, the dataset features medical indicators like the primary diagnosis (the cause for the rehabilitation) and scores, which represent the outcome of a standardized test that is performed for all patients with the same primary diagnosis. Another level of complexity is added to a data object as some parameters are *multi-dimensional* themselves. This can be seen in the last column where each object of the array refers to a single therapy.

Furthermore it can be observed, that the score columns contain *missing data*, which may have two reasons. Either the score is not measured per intention, as it is not linked to the diagnosis or it is missing by mistake (e.g., it was not entered in the system after it was measured). Another aspect that becomes clear in the dataset below are *inconsistencies*: ScoreX of the first patient is "170m" (integer with unit in short form) while the second patient got a score of "13.2 meter" (float with unit in long form and space in-between). Such inconsistencies are a typical result of missing form input validation, which makes them non-standardized text values that need to be parsed in order to be processed. After

Demographic				Medical				Therapy[1]
Id	Age	Sex	Zip Code	Diagnosis	Insurance	ScoreX[2,3]	ScoreY[2,3]	
1	67	male	1234	J44	Insurance A	170m	A-B	{a,b,c}
2	85	female	1235	I67	Insurance B	13.2 meter	-	{a,b,c,d}
3	47	female	1236	M17.9	Insurance A	-	C	{d,e,f}

Table 2.3. Characteristics of all approaches compared to each other. The table above contains columns with the mentioned problems of multi-dimensional data[1], missing data[2] and inconsistencies[3].

the introduction of the used dataset, the basic taxonomy for measurement scales and data types, and highlighting the complexity of the dataset, we can now conclude on the following question by Miksch et al. [MA14]: *What kinds of data are the users working with?* The users are working with a large dataset, in terms of feature dimensions and entries. Each feature in the dataset corresponds to an individual measurement scale and data type, which is an important factor for score data. Some entries in the dataset contain missing data, so a strategy for imputation must be determined. For some of the features in the dataset, no input validation has been performed, which makes extensive data cleaning necessary.

Users

Next, we want to highlight the potential users of the VA application. Generally, the staff in rehabilitation centres is a multi-disciplinary team. The medical staff requirements depend on the medical discipline of the facility, and are clearly defined [GFI+16]. For a neurological rehabilitation for example, it includes clinicians, nurses, physiotherapists, speech therapists, dietologists, social workers, and psychologists. Additionally, a rehabilitation centre requires administrative staff like managers, accountants, and IT-staff to work efficiently. All of these roles are possible users of our application. In the course of our research we identified two main groups of users to focus on, characterized by their background knowledge: clinicians and IT-staff.

First, we chose the clinicians, as they are involved in the total rehabilitation process, with ultimate responsibility. They are also aware of the data in the EHR and are to some point familiar with data analysis for research purposes. As their asset is their knowledge about the rehabilitation process, we refer to these users as *Domain Experts*. Domain experts are users that have a thorough background knowledge in the field of rehabilitation. In our case, domain experts are clinicians that work in a rehabilitation facility. In the context of in-patient rehabilitation, the domain experts play a vital role. It is their job to coordinate the inter-disciplinary team that includes specialists like occupational therapists, psychologists, dietologists, speech therapists, and many more. Furthermore, the treatment of diseases that are linked to the patients' rehabilitation is under the domain experts' responsibility. They also decide on the treatment strategy, as described in Section 2.2.2. In this context, the domain expert takes a dual role as operator and controller in terms of analyzing, steering, and monitoring the rehabilitation process. In Austria, rehabilitation is always bound to a clinician with ultimate responsibility [GFI+16].

Additionally, there is need for users that are more skilled on the technological aspects of rehabilitation. We chose the part of the IT-staff that is responsible for medical applications. This is because they have deep knowledge of EHR data and are used to analyse data in the rehabilitation context. We refer to them as *engineers*. In the setting of this thesis, they are responsible for the IT systems that are being operated for a rehabilitation facility. Typical applications to mention here are the rehabilitation information system (REIS) or the enterprise service bus (ESB). The REIS is an information system that is designed to manage the operational tasks of the rehabilitation facility such as medical, administrative or financial. The REIS is not the only information system that is being operated in a

modern rehabilitation center, numerous others such as therapy planners, billing systems or hospitality systems aid the patients' rehabilitation process. In order to connect all these systems to each other, an integration platform is needed. The ESB is responsible for managing application interfaces between all systems, including the respective protocol or message format. A solid technical background is required for the tasks of the engineers, that range from database operations to application load monitoring and customization of clinical forms in the REIS. As rehabilitation is a complex medical domain with a multitude of specific requirements, thorough knowledge of the rehabilitation process is mandatory for the engineers.

As already mentioned, rehabilitation is an inter-disciplinary process. The described user groups collaborate in order to improve the rehabilitation process of the patients.

Tasks

We will now conclude the analysis of our scope conditions with the last vertex of the Data–Users–Tasks Design Triangle [MA14], the tasks. At first we focus on the very basics when it comes to the description of the tasks. Task is a broad term, that may be understood in several ways. In order to overcome this issue, we discuss a general taxonomy on tasks in visualization based on the work by Brehmer and Munzner [BM13, Mun09]. A task can be described with varying levels of granularity, as the following examples from the medical domain show [Mun09]. If users stated that their objective was to cure a disease, we would consider the task description as high-level. On the other side, if users state they wanted to investigate microarray data showing gene expression levels and the network of gene interactions, we would view this as a low-level description of the task.

This example clearly demonstrates how strong the variation in the level of granularity may be. The high-level task gives us just a broad idea of what the users' task is, without giving any hint on what the users really want to do. On the other side we have the low-level task, that features a precise formulation on what the users want to do, without giving any information on his motivation and background. This is what Brehmer and Munzner refer to as "gap"; there must be some kind of task specification with a medium-level of granularity that tells the designer of a visualization what specific task the user wants to perform, including motivation and background information. A multi-level typology of abstract visualization tasks is introduced [BM13] (see Figure 2.3) to link the previously disconnected levels of granularity in multiple levels. The typology aims to describe tasks in order to answer three questions: *why* the task is performed, *how* the task is performed, and *what* are the task's inputs and outputs. On the *why* part the following terms appear: A visualization is usually either used in order to present information or to discover and analyze new information, this is referred to as *consume*. In some cases visualizations are just enjoyed, though. If a task is performed to generate a new artefact, such as transformed or derived data, annotations, recorded visualization interactions, or screen shots of static visualizations, the term *produce* is used. *search* is all about finding elements of interest in the visualization. There is a further distinction whether or not the identity or location of the search target is known a-priori. After a set of targets has been

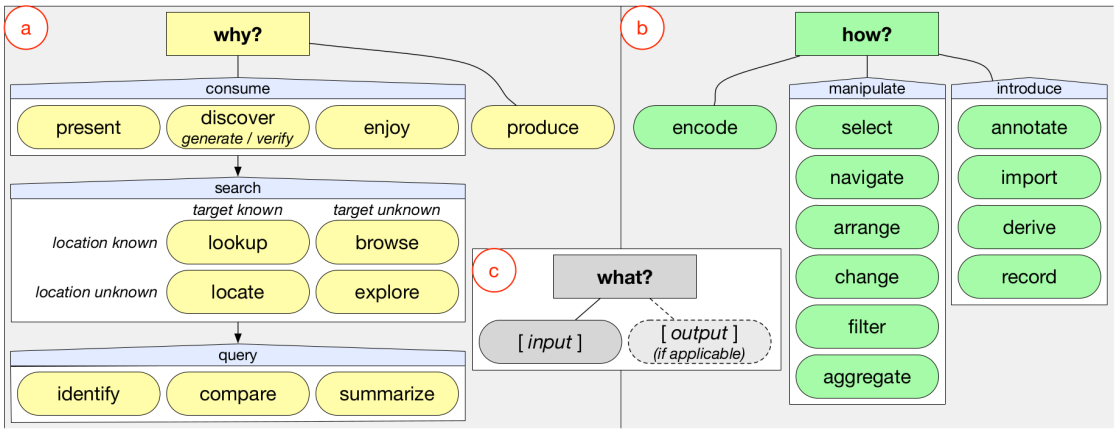

Figure 2.3. The multi-level typology of abstract visualization tasks described by Brehmer and Munzner. The typology covers the questions why, how, and what to justify the visualization tasks. [BM13].

found, querying is performed. The process when users identify, compare, or summarize these targets is called *query*.

Taking into account *how* the task is performed, the following taxonomy is introduced: Transforming data into a visualization is called *Encode*. A *Manipulate* task deals with the modification of existing elements in a visualization. Methods for manipulation come from the fields of interaction and visual encoding (e.g., focus + context [Car99]). *Introduce* tasks refer to adding new elements to an existing visualization. Possible examples here are adding textual annotations or recording visualization elements as artefacts.

Finally, there is a distinction to be made in order to classify *what* is being visualized. In this case, Brehmer and Munzner [BM13] tend to give examples rather than a rigid taxonomy:

- **Primitives:** Values, extrema, ranges, distributions, anomalies, clusters, correlations
- **Graph-specific objects:** nodes, links, paths, graphs, connected components, clusters, groups.
- **Time-oriented primitives:** points, intervals, spans, temporal patterns, rates of change, sequences, synchronization.
- **Interaction operands:** pixels, data values, data structures, attributes, geometric objects, geometric surfaces, visualization structures.

The tasks, as they have been defined by the engineers and domain experts in the course of the user task analysis, will now be introduced using the taxonomy described above. For

each task, we emphasize the why, what, and how. Additionally, we are going to provide a visual notation for each task, that enables further understanding. Tasks prefixed with *Eng* belong to the engineers, while *Exp* tasks correspond to the domain experts. Furthermore, all tasks have an id (e.g., Eng1, Exp2,...). This id corresponds to the level of complexity of the task and amount of knowledge needed, the easiest task starting at 1.

Eng1: Meaningful partitioning - Provide meaningful subsets of data

As it is the case with all clinical facilities, rehabilitation facilities carry out research to improve the quality of care in the long term. In the interviews, a domain expert stated that engineers' support is needed to get population data for retrospective studies. For this reason it is the engineers' task to design visual queries that *produce* a subcohort based on a given set of characteristics. The *input* data for this task is the total underlying data structure. At the *output* of the task results a subset of the data, representing a subcohort containing only relevant features and entries. The visual queries are implemented by a set of *filters*.

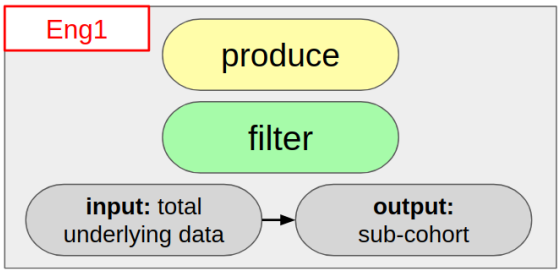

Figure 2.4. The task description of meaningful partitioning as visual notation [BM13].

Eng2: Assessment templates - Prepare templates for patient health assessment

In order to achieve the best possible rehabilitation outcome it is inevitable to involve the patient when it comes to the discussion of the outcome measurements. It is the engineers task to provide the data in the desired format to the domain experts. The domain experts stated in the interview, that visualizations of the results are easier to understand for the patients. The development of the scores over time are of particular interest, as well as the relation to the average performance of comparable patients. Again it is the engineers' task to *produce* a template that is further used by the domain experts. The *input* data structure used for this task contains the rehabilitation outcome measurements as well as demographic data of the patients, which is used when it comes to the comparison with other patients. The *output* is a dashboard template. The engineers must import visualizations to the template and *annotate* additional information on the displayed score results. As the dashboard is to be shown to patients that usually have no clinical expertise, it is necessary to *arrange* the visualizations in a clean way, *filter* only necessary data and *change* visualizations to be as simple as possible.

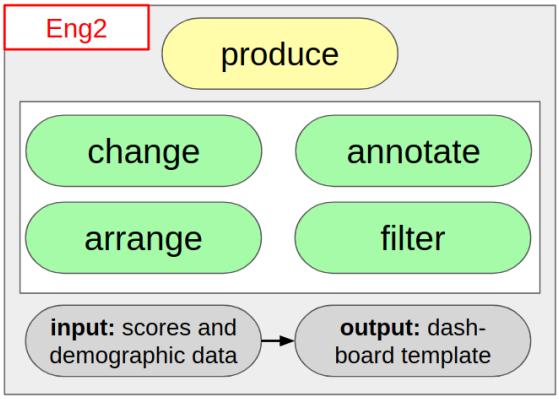

Figure 2.5. The task description of assessment templates as visual notation [BM13].

Eng3: Benchmarking templates - Prepare templates for clinical benchmarking

Clinical benchmarking is a tool used by healthcare facilities to monitor and improve their quality and efficiency [CCCD12]. The process is often installed to respond to external pressure from payers, government regulators, and affiliated healthcare delivery organizations [RMM01]. Rehabilitation outcome measures as described in Section 2.2.2 are the fundamental data source for this kind of analysis. The task of clinical benchmarking is described quite frequently in the interviews by engineers as well as domain experts. The engineers' task is to *produce* a template that is used by the domain experts (or other entities) for clinical benchmarking. The *input* is a data structure containing outcome measures and clinical effort, e.g., the total minutes used for therapy. The *output* is a dashboard template for clinical benchmarking. New entities must be introduced to the dashboard like the *import* of the required visualizations and *annotations* for additional descriptions. Those visualizations need to be manipulated to fit their very purpose. This can be done by *aggregating* data, *arranging* visualizations, *filtering* the underlying data or *changing* some visualizations.

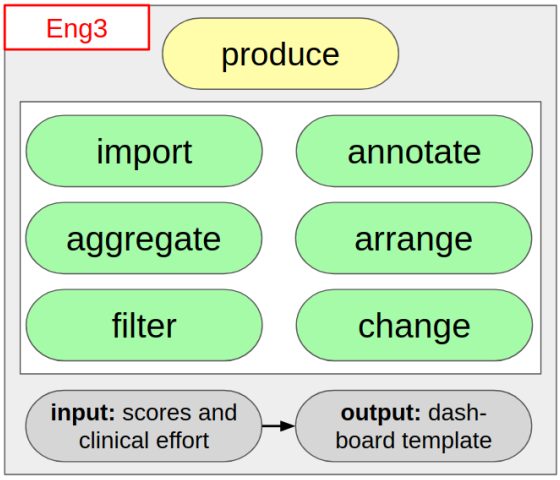

Figure 2.6. The task description of benchmarking templates as visual notation [BM13].

Eng4: Outcome predictions - Use machine learning to predict rehabilitation outcome

An aspect of particular interest of one domain expert in the interviews is the potential application of machine learning algorithms on rehabilitation data. All kinds of data that have been collected over the years can be utilized to predict specific rehabilitation outcome scores. This enables the domain experts to predict the outcome of the planned intervention strategy at the start of the rehabilitation. From the engineers' point of view, this prediction enables insight on correlations of certain features in the dataset—aiding them to gain further knowledge of the underlying dataset. A byproduct of the prediction is the influence of specific features on the rehabilitation success. This information can further be used to adjust the intervention strategy. In order to visually *discover* the data, the engineers must be able to *browse* the dataset to *identify* the parameters responsible for a successful rehabilitation. The full data structure is the *input* for this kind of analysis. *Deriving* the prediction is possible by *filtering* and *aggregating* the dataset.

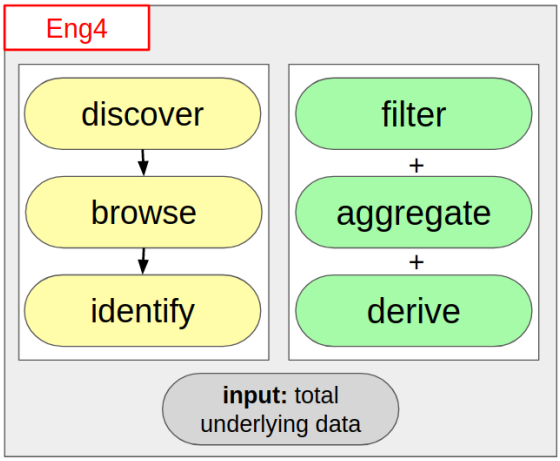

Figure 2.7. The task description of outcome predictions as visual notation [BM13].

Exp1: Outcome presentation - Show rehabilitation outcome to patients

This task is built on top of Task Eng2. In order to *present* visual results to the patient and *summarize* the patient's performance, e.g., with mean performance, the data is *looked up* using the dashboard created in Task Eng4 as *input*. No modifications besides adjustments to achieve improved values for comparison are needed. The domain experts must be able to *select* items of interest and perform *navigation* tasks such as zooming.

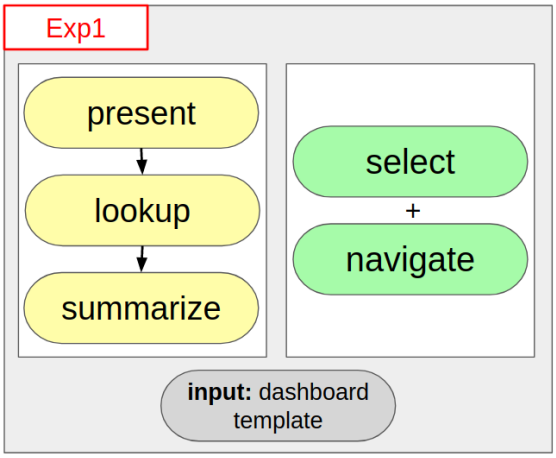

Figure 2.8. The task description of outcome presentation as visual notation [BM13].

Exp2: Clinical benchmarking - Perform clinical benchmarking

In this task the domain experts or other professions perform the clinical benchmarking based on the template from Task Eng3 as *input*. One person stated the following in the interview: *"The clinical benchmarking is the justification for the rehabilitation. There is need for measures that tell us, if the rehabilitation makes sense."*. An example here would be the average increase of a specific health assessment by day. The benchmarks must be *summarized*, so they can be *looked up* and *discovered* in a clear way. It is not desirable to change the clinical benchmarks frequently, as they need to be monitored over time. The users must be able to *navigate* through the time axis of the visualization, *selection* of specific points of interest may be helpful. If a benchmark is taken for a particular case, *filtering* and *aggregation* will be applied.

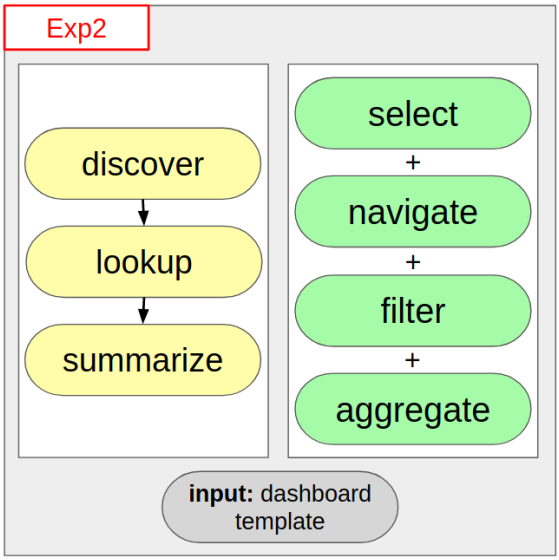

Figure 2.9. The task description of clinical benchmarking as visual notation [BM13].

Exp3: Clinical exploration - Explore clinical dataset

All domain experts in the interviews mentioned, that they want means to explore the data to purely enjoy the data. One domain expert referred to this as *"curiously playing around with the data"*. The domain experts state that this very unique way of looking into the data may lead to new ideas for scientific research. The motivation for this task is to *enjoy* the visualization, while *exploring* the data in order to *identify* features that are of particular interest for retrospective studies. No restrictions are made on the *input* data, in order to preserve the explorative intention of the approach. All tools are available for the users of this task from *filtering*, *arranging*, *aggregating* to even *encoding* new visualizations.

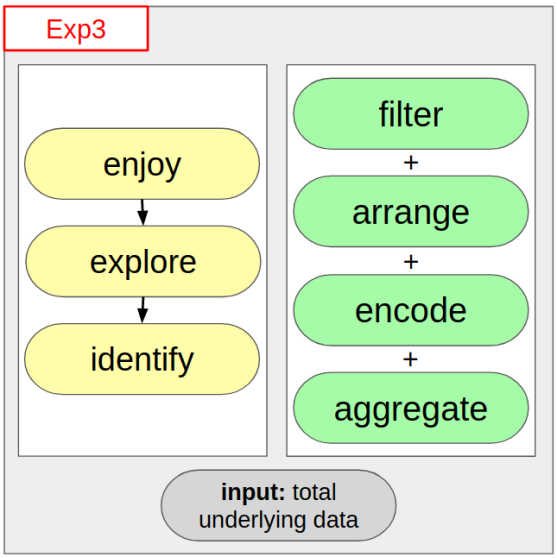

Figure 2.10. The task description of clinical exploration as visual notation [BM13].

Exp4: Clinical analysis - Analyse data for clinical studies

Scientific research is part of the domain experts clinical work in the rehabilitation facilities. The overall goal is to enforce a constant improvement of treatment strategies and assessment tools. For example, it may be the case that a domain expert requires the data of all osteoarthrosis patients of the past five years to determine a set of correlating features in the cohort. The domain experts use the dashboard to *discover*, *lookup*, and *compare* data corresponding to a treatment strategy or assessment tool. In this task the domain experts use the data structure resulting from the visual queries created by the engineers in Task Eng1, as *input*. Means for *selection*, *navigation*, *filtering*, or *aggregation* support the domain experts in their task.

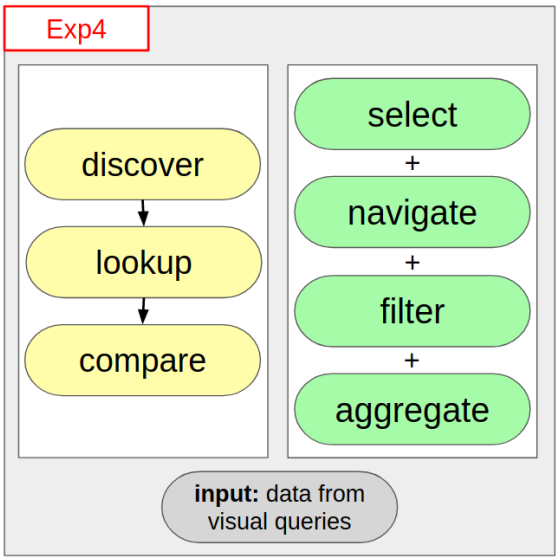

Figure 2.11. The task description of clinical analysis as visual notation [BM13].

Exp5: Intervention planning - Use machine learning for intervention planning

Similar to task Eng4, machine learning is a tool that the domain experts want to use also. One domain expert states that machine learning will be an important tool for *"calculating and extracting the main factors for a successful rehabilitation"*. It is the clinicians' goal to modify the clinical intervention in order to maximize the rehabilitation outcome. There is need to make the machine learning functionality available for the domain experts, with respect to their a-priori knowledge in this field. Correlations can be *discovered* by *browsing* through the data to *identify* or *compare* outcome measures of interest. The *input* for this task is the data structure, containing all health assessments that are present in at least 0.5% cases in the total cohort. Outcome predictions for varying subcohorts can be *derived* by applying *filters* to the population.

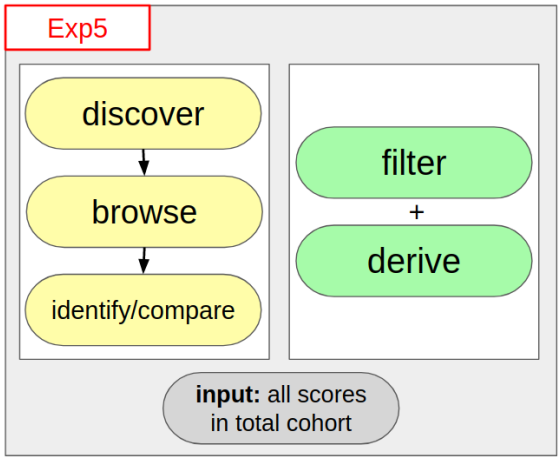

Figure 2.12. The task description of intervention planning as visual notation [BM13].

2.3 Establishing Precision Rehabilitation

Precision medicine is a new trend in medicine, that aims to improve the patients' treatment, by taking individual variability into account. The Austrian rehabilitation system is based on a classification system for rehabilitation phases. Certain legal aspects apply to it. Rehabilitation outcome measurement is a possibility to quantify the effect of rehabilitation interventions and makes them visible. The scope conditions for our visualization are defined by users, data, and tasks.

This clearly sets the scope for our Visual Analytics application. We apply the principles of precision medicine to the field of rehabilitation. It is our aim to create an application that allows domain experts and engineers to perform analytic tasks. The patient data we aim to visualize is stored in electronic health records. We predict rehabilitation outcome measurements based on patient characteristics. It is our task to identify characteristics that are responsible for a successful rehabilitation within a large, complex, and high-dimensional dataset.

To the best of our knowledge, there is no previous work on the topic of Visual Analytics for precision rehabilitation. Research on personalized neurological rehabilitation was carried out by the so called "NeuroRehabLab". Publications dealing with personalized rehabilitation tasks [FiB15] and a web-tool for the generation of personalized cognitive rehabilitation training [FBiB18] were made based on standardized outcome measures. Approaches as the ones described show the importance of standardized rehabilitation outcome measurement in the context of precision rehabilitation. Though, none of them applied Visual Analytics for the present issues. We will thoroughly introduce related work in Chapter 3. We aim to support and extend approaches as the ones described by providing a powerful Visual Analytics application.

CHAPTER 3

State of the Art

This chapter, will focus on the state of the art within the fields related to this thesis. We intend to summarize basic knowledge in the specialized fields of medical visualization. First we will introduce Visual Analytics, the fundamental research field where this diploma thesis belongs, and related techniques. Next, we will introduce possible applications of visualization in rehabilitation, these may be related to: visualizations used in rehabilitation, visualizations used for population studies and visualizations for the analysis of electronic health record data. Furthermore, we will shift our point of view to the underlying data, presenting fundamental techniques in visualization of EHRs and population data.

3.1 Visual Analytics

For large and complex datasets with a multitude of features, as within the scope of this diploma thesis, simple visualizations are not enough. This is caused by the characteristics of the dataset: A high number of patients, heterogeneous features, and missing data. There is a need for tools that enable interactive exploration of such datasets and help the human brain to process it. One approach to address these challenges is Visual Analytics (VA), which was introduced by Wong and Thomas in 2004 [WT04]. Visual Analytics is a subfield of visualization that evolved from the fields of information visualization and scientific visualization [KMS+08]. A wide range of applications for Visual Analytics may be found in medical visualization [KMS+08].

"Visual Analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets" [KAF+08]

This definition by Keim et al. is used in the literature to briefly describe the main focus of VA. Visual Analytics is rather a collection of techniques than a single tool that aims

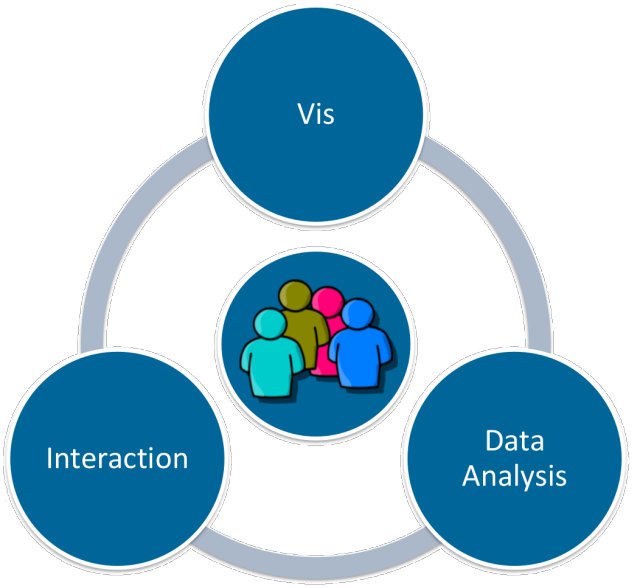

Figure 3.1. The components of Visual Analytics from a high level perspective.

to reach the following goals [KAF+08]:

(a) Synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data.

(b) Detect the expected and discover the unexpected.

(c) Provide timely, defensible, and understandable assessments.

(d) Communicate these assessment effectively for action.

Visual Analytics may be split into three logical components: visualization, interaction, and data analysis. The logical components are visualized in Figure 3.1. Visualization aims to provide knowledge via figures. Figures are richer in information, they make structures more visible, are more memorable and faster to grasp than text [PBG98]. Interaction allows the user to control the visualizations, which enables them to gather more information than through a simple non-interactive interpretation. Data analysis covers aspects like data mining, machine learning, or statistics which map the data to visual models.

The mindset behind Visual Analytics is to combine strengths from a multitude of areas, including (but not limited to) information analytics, statistical analytics, data management, and knowledge presentation. Furthermore, the strengths of the users in terms of cognition, perception, and interaction [KMS+08] are taken into account. To summarize, the result of Visual Analytics is a comprehensive tool that covers the whole

path from complex datasets to powerful visualizations, combining a wide range of methods and techniques.

In this diploma thesis Visual Analytics is applied to provide visualizations for insight in a large and complex dataset, utilize data analysis to study its features, and provide means to integrate a prediction of the rehabilitation outcome in the visualization.

3.1.1 Fundamental Techniques

As Visual Analytics forms the core of this thesis, a wide range of its extensive tools will be utilized. This section briefly introduces the fundamental techniques and methods of Visual Analytics, as collected by Raidou [Rai17]:

Visual Information Seeking Mantra

The visual information seeking mantra [Shn96] describes a design guideline for visualizations following aspects of human perception. *Overview first*: the first depiction for the user to see should be an overview to get a first impression of the data. Next, the user can *zoom and filter* to get a more detailed view on the items of interest and filter out uninteresting items. Finally through *details-on-demand*, details on the items of interest can now be shown to the user. The mantra can be summarized as follows: *Overview first, zoom and filter, then details-on-demand.*

Visual Analytics Seeking Mantra

As the visual information seeking mantra does not fully support the needs of Visual Analytics, it was adapted and extended by Keim et al. [KAF+08]: *Analyze first – Show the important – Zoom, filter and analyze further – Details-on-demand.* Contrary to the first mantra, an initial analysis is performed to show the important aspects of the data.

Data-Users-Tasks Design Triangle

The *Data-Users-Tasks Design Triangle* [MA14] is a framework, created to support designers of Visual Analytics applications by taking three aspects into account: The dataset the application is built on, the users the application is designed for, and the tasks the users aim to fulfill with the application.

Multiple (Coordinated) Views

Multiple (Coordinated) Views [WBWK00] is a typical method applied in Visual Analytics. Multiple views are applied to provide the user with different perspectives on the data at the same time. This way, the user is able to get more information from the data, while preserving the general context of the visualization. This method is often used in combination with Brushing and Linking, to further improve the visualization of relationships in the data.

Brushing and Linking (B/L)

The process of selecting one or more interesting items in one view and highlighting it in an other one is called *Brushing and Linking* [Kei02, BMMS91, BC87]. This way multiple characteristics of an item can be explored across different views, providing additional information compared to the individual views.

Focus + Context (F+C)

Focus + Context [Car99] visualization is a concept from the information visualization field: Interesting subsets of data are shown in a more detailed way, while less relevant parts of the data are shown with less detail in order to preserve the context, which supports user navigation and orientation.

Overview + Detail

In *Overview + Detail* [CKB09] at least two views are presented: One with a general overview of the entire visualization and the other one with a more detailed view of a data subset. Therefore, the advantages of Multiple Views and Focus + Context are combined.

Dashboards

Dashboards [SCB+19] are widely used in data visualization. They feature visual data representation in a tiled layout of simple charts. Interactive dashboards may be used for real-time monitoring of dynamically updating data. In general, dashboards support strategic, tactical, or operational decisions and enhance communication and learning. A major reason for the importance of dashboards is the fact that they often are a first encounter with the data.

The introduced techniques of Visual Analytics are to be seen as generally adoptable and applicable methods, that are taken into account in the design of our VA application. As precision rehabilitation features a set of domain-specific challenges that remain unanswered, such as visualizing EHR data, visualizing rehabilitation data and visualizing population data. For this reason, there is a significant need to introduce more relevant and state of the art techniques. In the remaining chapter, we will take a look at visualization approaches related to the topic of this diploma thesis, i.e., rehabilitation (Section 3.2). Next, in Section 3.3 we look into methods for visualizing data stored in electronic health records. We close this chapter in Section 3.4 by introducing methods for visualizing population study data.

3.2 Visualization in Rehabilitation

A variety of visualization techniques are used in rehabilitation. We can roughly categorize these techniques into three main groups. The first group plays an active part in the patients' therapy, usually in the form of visual feedback to the patients themselves. Secondly, visualization is used by therapists to evaluate exercises performed by patients.

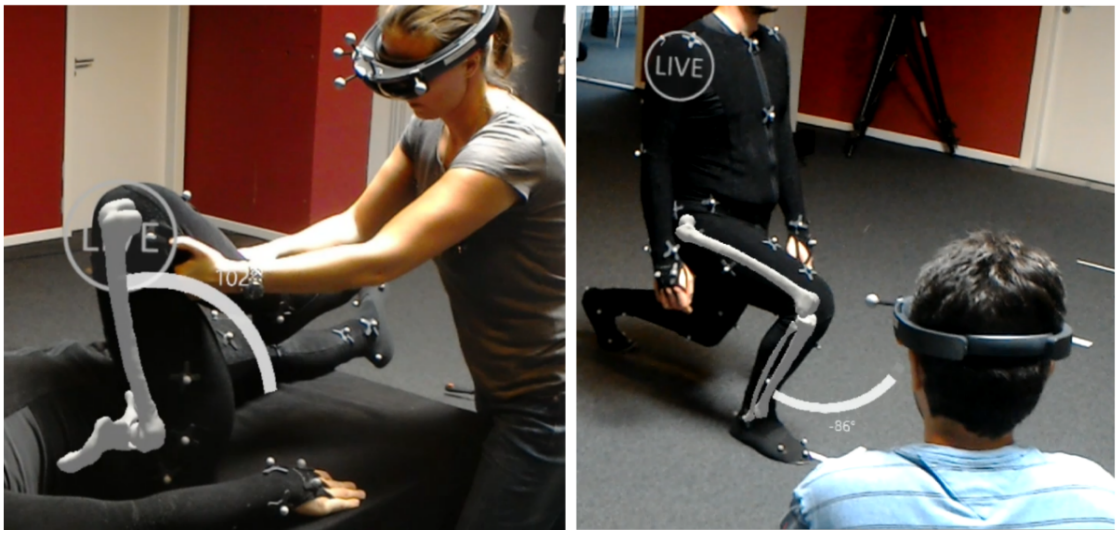

Figure 3.2. Examination of the joint range of motion during passive (left) and active (right) movements using augmented reality [DdOL+18].

Another important aspect highlighted here is the usage of visualization as a tool for research collaboration.

Rincon et al. [RYS16] visualize muscle movement in a virtual reality application. A motion sensor captures the muscle movement via an electromyography (EMG) signal, which is translated to movement in a video game. This enables the patients to perform motions that may not be possible in their current rehabilitation process, e.g., after a brain injury. This serious gaming approach tackles two issues: reducing post-stroke depression by providing an encouraging environment and training patients that are going to get a prosthesis. Virtual reality is also used in the OctaVis system [DZK+12]. Patients suffering from neurological disorders can use this system to make a virtual shopping trip. This scenario checks spatial orientation (for finding shopping items) as much as the memory function (for remembering the shopping list).

Numerous visualization approaches were utilized in order to assist doctors or therapists in terms of test-result evaluation. Data visualization has been applied to tele-rehabilitation, a form of rehabilitation where the patients perform therapies mostly at home [ML17]. The patients track their motion with sensors integrated into their smartphone. This data is visualized for doctors and therapists to check the quality of rehabilitation, in order to answer questions such as: Are the therapies carried out correctly? Can improvements be detected? Augmented reality may be used to evaluate the range of motion (ROM) of the joints of orthopedic patients [DdOL+18]. Therapists or doctors see a holographic overlay that represents the movement of the joints (including bones), while the patient performs the movement (see Figure 3.2). In periodic sessions the patient's ROM improvement is monitored.

The last and probably most relevant usage of visualization in rehabilitation is the

presentation of clinical data, e.g., for studies. The idea behind this is to aid basic and clinical scientists by using interactive visualization of data from randomized control trials (RCTs) stored in the Centralized Open-Access Rehabilitation database for Stroke (SCOAR) [LSR+16]. This enables the scientists to quickly visualize relationships among variables and efficiently share data. Furthermore the generation of hypotheses is supported as well as searching current literature based on the underlying dataset. Also, clinical trial design is improved through more accurate and comprehensive power analyses. As a follow up to this work, Lohse et al. [LPW+18] improve the amount and quality of underlying data in SCOAR by applying text-based analyses.

3.3 Visualization of Electronic Health Records

In medical visualization, the electronic health record serves as an excellent data source. Its data is structurally persisted in a standardized way for each patient and stored for years. The importance of population data and retrospective cohort analysis is highlighted in Section 3.4. In this section we will present fundamental techniques to visualize EHRs used in recent publications.

A typical technique applied to visualize data from electronic health records is *filtering* or *querying* through a user interface, referred to as "Visual Queries" by Götz et al. [GWP14]. Defining sub-cohorts is an iterative process that involves performing queries on the study population. In environments where visual querying is not available, clinicians rely on database experts or other technologists to create SQL queries for them [ZGP15]. With visual queries, users can build queries in several ways: adding filter elements to a query via drag and drop [GWP14, KPS16], choosing subspaces in visualizations [ZGP15, AHN+17], or selecting a range in histograms [RSN+19].

Another important technique is *temporal event analysis*, with mining-based and visualization-based methods. Electronic health records usually store events associated with the time they occur (e.g., patient receives medication). With temporal event analysis, patterns in the timing of this event can be determined to gain insights from clinical event sequence data. This information can be used to determine patterns and correlation with the patients outcome measures. Visualization-based temporal event analysis of EHR data is used in a multitude of applications [GWP14, ZGP15, RPOC18, KPS16].

Hierarchical data is often used in electronic health records, therefore techniques for *hierarchical data visualization* are applied. As an example for hierarchical data, we take a look on the ICD-10 classification system: a bilateral form of primary osteoarthritis (joint disease) of the knee is encoded as M17.0. M17 corresponds to all forms of osteoarthritis of the knee. M15 to M19 correspond to all forms of osteoarthritis (a specific form of joint disease). M corresponds to all *"diseases of the musculoskeletal system and connective tissue"* [Wor04]. This hierarchical structure is described in Table 3.1. To visualize such structures from EHRs, sankey diagrams may be used [ZGP15]. Another approach done by Krause et al. [KPS16] is the utilization of tree maps.

Level	ICD10 hierarchy	Example
1	Chapter	Diseases of the musculoskeletal system and tissue (M00-M99)
2	Section	Osteoarthritis (M15-M19)
3	Group	Osteoarthritis of knee (M17)
4	Diagnosis	Bilateral primary osteoarthritis of knee (M17.0)

Table 3.1. An example for hierarchical data: The structure of a diagnosis in ICD10 [Wor04].

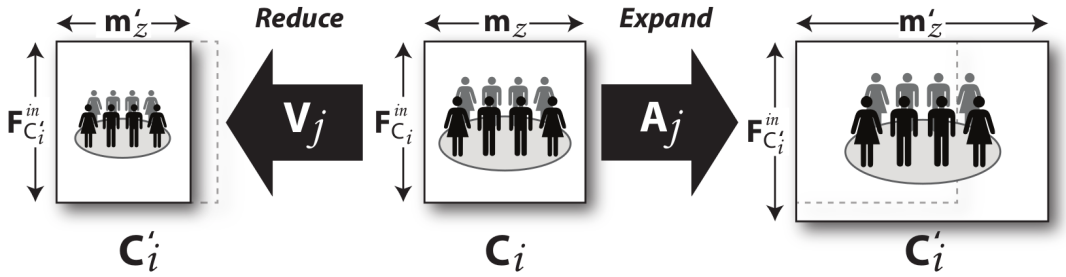

Figure 3.3. Redefining a cohort as described by Zhang et al. [ZGP15].

Visual Analytics is a very broad field, featuring a varied literature. It is impossible to cover all aspects of Visual Analytics in a single chapter. For this reason we will refer to additional work when we will be discussing the conceptual design choices of our work.

3.4 Visualization of Population Data

Visualizing population data has recently become a common task in medical visualization. A frequent problem in population-data visualization is analyzing cohorts. Especially in medicine, cohort analysis is used to identify risk factors among sub-populations. Cohort analysis can be performed in prospective or retrospective. In retrospective analysis, cohort analysis is performed on a previously collected dataset, while in prospective cohort analysis the data is collected prior to the analysis. Retrospective is used, e.g., for determining the behavior of a cohort concerning a specific treatment [RCMA+18]. Prospective studies can possibly be used to predict, e.g., the course of disease for the health status of the population.

Zhang et al. [ZGP15] describe the process of cohort analysis as an iterative task: Physicians propose hypotheses and test them against an initial cohort $\mathbf{C}_i$. As it is unlikely that the hypothesis is perfectly defined from the beginning, the physicians want to check the hypothesis against a redefined cohort $\mathbf{C}'_i$ of the population. Each cohort is described across two dimensions: its members $\mathbf{m}_z$ and a set of features $\mathbf{F}^{in}_{C_i}$ associated with each member. The process is repeated until all hypotheses are tested and all related subsequent questions are answered. Figure 3.3 demonstrates the iterative process described above. The cohort can be expanded by applying an analytic $\mathbf{A}_j$ that adds additional members. In order to diminish the cohort, interactive visualizations $\mathbf{V}_j$ may be used.

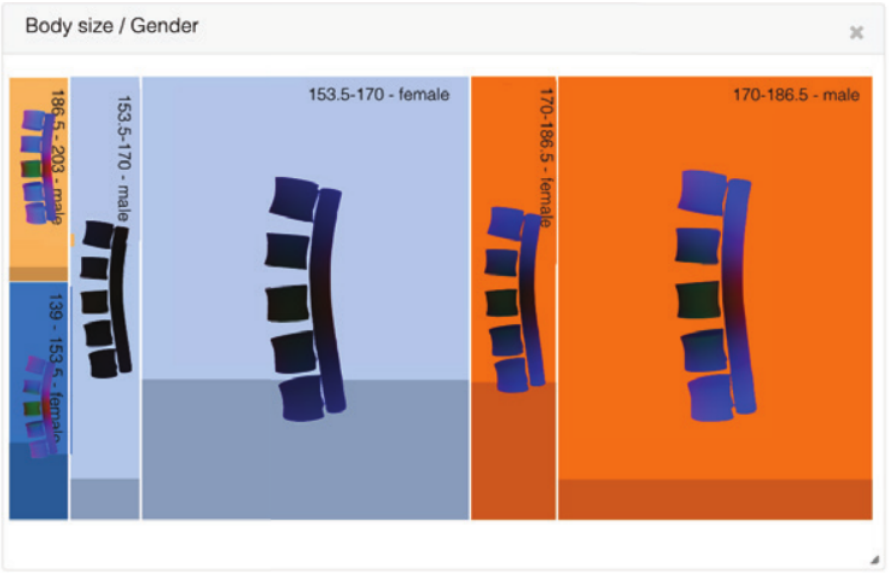

Figure 3.4. Example for comparative visualization. Different spine positions related to body size are compared between male and female patients [KOJL+14].

A cohort comprises a very high number of features. One of the most important tasks in cohort analysis is testing those features for associations with regression analysis. Showing the associations in a visualization can be challenging. Usually this task is performed with heatmaps [AHN+17]. If the number of features is large, 3D heatmaps are used [KLG+16].

As mentioned before, the definition of subcohorts is an important task in cohort analysis [ZGP15, RSN+19]. Traditional hypothesis-driven and statistics-focused approaches may fail to identify a subpopulation. Subspace clustering solves this problem by automatically discovering subspaces through clustering algorithms. Visualizations are used to highlight potential subpopulations [AHN+17].

In the context of population-data visualization it is worth mentioning the related field of *Comparative Visualization*. Comparative visualization is typically used for imaging data, which is not a part of this diploma thesis. Still, it is largely used, as part of population visualization (e.g., in a paper on image-centric cohort visualization by Klemm et al. [KOJL+14], see Figure 3.4) and related to previous work mentioned in this section. For this reason, it will be briefly introduced.

Comparative visualization is used in information visualization and scientific visualization to display multiple data instances, enabling visual comparison. Gleicher et al. [GAW+11] introduced a general taxonomy for comparison tasks in visualization in 2011. The taxonomy is structured by considering the relationships between the corresponding parts of the different data instances. Three categories are introduced, all of them are summarized in Figure 3.5:

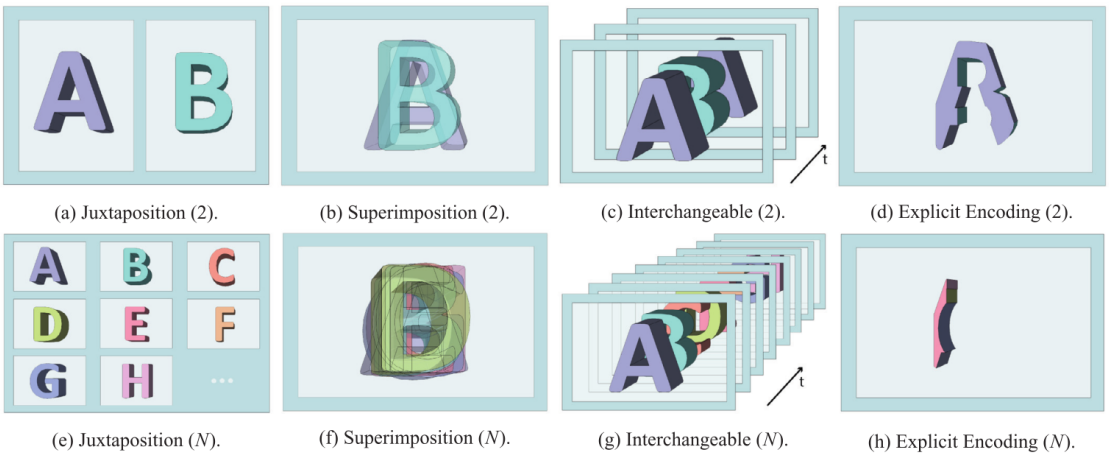

Figure 3.5. Illustrations of four approaches to compare visualizations. The image is taken from [KCK17].

- Juxtaposition (a,e): The images (two or more) are displayed side-by-side, each in its own coordinate system.
- Superimposition (b,f): The images (two or more) are displayed on top of each other, in a common coordinate system. As this view introduces occlusion related issues, further modifications (filtering, opacity,...) may be necessary.
- Explicit Encoding (d,h): The images (two or more) as displayed are the result of a common composite feature. Examples would be boolean intersection (as it is the case in Figure 3.5), union, or difference.

Hybrid approaches combine two or more methods from the above taxonomy to overcome the issues of a single approach. The original taxonomy [GAW+11] was updated by Kim et al. [KCK17] in 2017. A new category called "Interchangeable" is introduced here:

- Interchangeable (c,g): All images for the comparison are represented in a separate visualization layer. This causes only one image to be displayed at a time. The data instances can be changed manually or automatically and the transition can be modified if required (e.g., for a smoother transition). All data instances share the same coordinate system.

The intention of this chapter is to give an overview of related visualization approaches and techniques. In upcoming sections we will review literature in more detail with regard to the applied approach.

CHAPTER 4

Visualization Design

Up to this point, we presented the State of the Art and the Background related to this thesis. While these chapters mainly focused on characterizing the problem space and general approaches towards visualization problems, this chapter initially highlights steps for the solution of the specific problem of this thesis. First of all, we will start with describing the process of the user-task analysis. This section explains in detail how the tasks that were introduced in Section 2.2.3 were derived. The design chapter continues with a description and schematic graphical overview of the application workflow. Finally, approaches to realize the introduced tasks are discussed and analyzed.

4.1 User-Task Analysis

In the previous chapter we introduced the granularity of task descriptions. While the high-level task of this thesis, i.e., improving the quality of rehabilitation, was clear from the beginning, some aspects were left untouched. Even the most advanced and sophisticated system will not have a major impact on the rehabilitation process, if it is not developed with respect to the users' requirements and work flow. It is the purpose of the user-task analysis, to determine the users' work flow and practises. The resulting tasks are the main input to the design choices of a visualization.

In the nested model by Munzner [Mun09], the process described above is part of the top level (domain problem and data characterization) and there are several aspects to consider. For each visualization there is a specific target domain with its own vocabulary to describe its data and problems. Also there is an existing work flow of how the users utilize the data to solve their problems. The output of this level in Munzner's model is a detailed set of actions carried out by the target users. A graphical overview of the model can be seen in Figure 4.1. Similarly, in the seven evaluation scenarios described by Lam et al. [LBI+12], the user-task analysis is represented by the first scenario: "Understanding Environments and Work Practices". The goal of this scenario is to understand work,

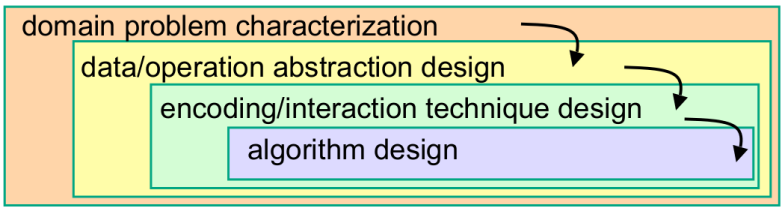

Figure 4.1. The nested model by Munzner [Mun09].

analysis, or information processing practices of the target users. This includes the analysis of work flows, work practices, working environment conditions, and current tools in use. Especially the latter is of interest as current tools have a huge impact on the user's needs and expectations. The output of this scenario are design implications.

There are several methods for a user-task analysis, this thesis aims to focus on the common ones. According to Lam et al. [LBI+12] using *Field Observations* is the most common method to determine information on visualization use and work practise. The goal of these observations is to analyse the user's tasks in a real-world setting, where the users act completely free. Field observations can be followed by a questionnaire for a better understanding of the observed aspects. On the other hand there are *Laboratory Observations* that happen in laboratory settings. This allows the observer to gain more control of the study situation.

Another method for user-task analysis are *Interviews*. In contextual inquiry interviews, users are first observed and then interviewed, all in the course of their daily routine in their work environment. Another common interview method in information visualization is to interview domain experts in a laboratory context. An example for this is work by Brewer et al. [BMA+00], who interviewed six domain experts in geovisualization. In general, three types of interviews can be distinguished: structured, semi-structured, and unstructured interviews. Structured interviews follow a predetermined set of questions, that are exactly repeated in each individual interview. Semi-structured interviews on the other hand have a rough framework of questions that are to be explored, but allow leeway depending on the answers of the person being interviewed (e.g., as used by Pretorius et al. [PvW08] or Brewer et al. [BMA+00]). Unstructured interviews have no set of predetermined questions, the interviewers just aim to cover certain topics.

A rather novel approach to user-task analysis are so-called *Creative Visualization-Opportunities (CVO) Workshops* (see Figure 4.2. Kerzner et al. [KGD+19] analysed a total of 17 workshops from ten visualization contexts over the course of two years to develop a framework for CVO workshops. CVO workshops enable researches and domain experts to explore opportunities for visualization in a domain within a few days of collaboration.

All of the above introduced methods were considered for the user-task analysis of this diploma thesis. The first approach that had to be dropped are the CVO workshops as they consume too much personnel resources over the course of several days. Observations on the

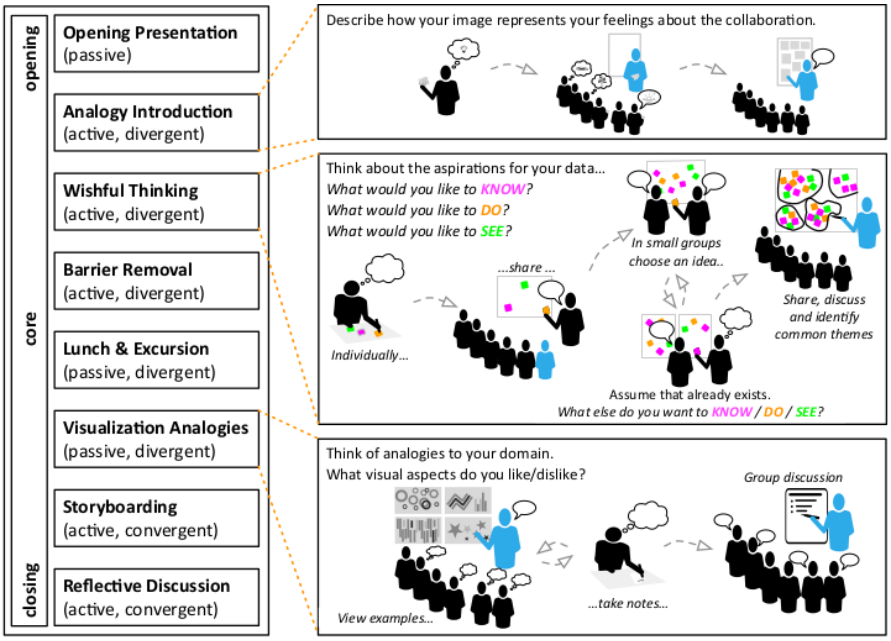

Figure 4.2. An example of a full-day CVO workshop as described by Kerzner et al. [KGD+19]. The workshop day consists of eight methods (left), three of which are described in detail (right). For each method there is a description if it is active or passive in terms of participation style. Furthermore it is stated whether the method is used to bring up new ideas and widen the idea space (divergent) or winnow the idea space (convergent).

other hand are also hard to conduct, as domain experts and engineers work decentralized and distributed all over Austria. Furthermore, CVO workshops are better suited for teams that work in collaborative environments. In contrast, interviews are easier to implement as they can easily be conducted remotely. Unstructured interviews are avoided as the guideline they feature is too vague, structured interviews on the other hand leave too little space for the persons interviewed to freely share their thoughts and ideas [LBI+12]. Furthermore, the two groups of users have different backgrounds and professions and are therefore not suitable for sharing the same prepared set of interview questions. We conclude to work with semi-structured interviews. Semi-structured interviews represent the "golden mean", as they are not as resource-consuming as workshops, not bound to a location, and feature the right degree of guideline for the interview.

Interviews were held with six individuals, four of which are engineers and two are domain experts (as described in Section 2.2.3). Six participants are considered sufficient for our needs, as this diploma thesis is a novel approach, that is to be reviewed before taking further steps with a larger group of users. For the same reason we restricted the scope to neurological and orthopedic rehabilitation only. Five of the participants are potential users of the Visual Analytics application, one is a data-science and data-

visualization expert, who was treated as engineer. All of them are used to work in multidisciplinary teams, sharing their experience from technology and rehabilitation medicine. The interviews were designed to get an idea of current technologies and work practise in rehabilitation. The five potential users had never used a Visual Analytics application before. All interviews were performed individually, the domain experts were interviewed via telephone.

The organized interview sessions were about 30 minutes long, except for the interview with the data-science expert,which lasted for about an hour. All interview sessions were recorded. The audio recording was later transcribed. Two main topics were subject of the interview. The first topic is about finding out current work practises of visualization in rehabilitation. The second topic is a look in the future. A set of questions was prepared for both topics, simply in order to maintain the flow of the interview. All interviews developed their own pace and drifted away from the guideline to an individual point of view, the desired output.

Domain-Experts Interview

The questions for the domain-experts interview were designed to determine current workflows in the rehabilitation context. We aimed to use "do you? questions" in order to make as few assumptions as possible. The interview was started with brief statements on the domain experts current position and their background, especially in data visualization. The following questions were asked in order to learn about the current exploration and visualization of data:

- Do you share information? How do you share it (meetings, video conferences, ...)?
- How are collaborations with IT-staff like? Do you need them to look up things in data? (e.g., referrer statistic)
- Do you use visualizations? If yes: which?
- Do you browse through data just out of curiosity?
- Do you use any other data source to obtain information? If yes: which?
- Do you conduct studies with EHR data?
- Do you use applications to deal with data (Excel, ...)?
- Do you write reports, summaries (for management, congress, ...)?
- Do you compare groups of patients with each other? (subcohorts)
- Which of the above do you actually perform? How often?
- Do you let the data influence your treatment?
- What influences your treatment?

- When you do analysis, what data do you look at? (Based on categories, characteristics)
- How is the process of looking into the data, describe!

Furthermore, we aimed to ask the domain experts about their expectations concerning the thesis. We also focused on possible future trends and their impacts on rehabilitation workflow and tasks.

- What of the above would you be interested in? What else?
- Are you satisfied with the way tasks are fulfilled today?
- What are tasks that should be improved or made in a different way? Can you think of limitations (tools, data, skills, ...)?
- Are there any things that you can imagine to be more automatic?

Engineers Interview

The engineers interview questions were quite similar. We largely focused on current workflows and also used "do you? questions" in order to make as little assumptions as possible. Again, we asked for brief statements on current positions and background knowledge, with a focus on artefacts that are produced with data. We prepared the following questions for the interviews:

- Do you share information? How do you share it (meetings, video conferences, ...)?
- How are collaborations with domain experts like? Do they want you to look up things in the data? (e.g., referrer statistic)
- Do you use visualizations? If yes: which?
- Do you use any other data source to obtain information? If yes: which?
- Do you use applications to deal with data (Excel, ...)?
- Do you write reports, summaries (for management, congress, ...)?
- Which of the above do you actually perform? How often?
- Would you say, data has influence on a patients treatment?
- When you do analysis, what data do you look at? (Based on categories, characteristics)
- How is the process of looking into the data, describe!

And again, we prepared questions on possible future usage of data analytics and expected future trends.

- What of the above would you be interested in? What else?
- Are you satisfied with the way tasks are fulfilled today?
- What are tasks that should be improved or made in a different way? Can you think of limitations (tools, data, skills, ...)?
- Are there any things that you can imagine to be more automatic?

4.2 Application Overview

The application we aim to introduce consists of a set of independent modules that are interconnected. This makes the components more interchangeable, which may come in handy when one module has to be replaced, e.g., for technological reasons. In Chapter 5, we will therefore highlight all interfaces between the components. We refer to the full application with all modules as *preha*, a combination of **pre**cision and **reha**bilitation. The *preprocessing* module is responsible for collecting the data from various sources (e.g., database tables), reformatting all data to a single data structure, and standardizing the quality of the data. The *data storage* is the primary persistence unit for preha. Once the preprocessing module stores data in it, the data will not be modified further. The *Visual Analytics dashboard* is the user interface of the application. It features dashboards with rich sets of visualizations that are used for data analysis by domain experts and engineers. This dashboard is where visual queries are built, subcohorts are created, or advanced data-analytics tasks are initiated. The *data analytics module* is responsible for advanced data-analytic tasks, such as machine learning and predictions. The latter three modules—data analytics, Visual Analytics dashboard and data storage—can interact in an iterative process, as described by Zhang et al. [ZGP15]. Figure 4.3 shows an overview of all modules from a top level perspective. All software used for preha is open source, as further discussed in Chapter 5.

4.3 Preprocessing

The preprocessing module is the point where the unprocessed, raw data enters the application. The raw data is extracted from the EHR as CSV file. The preprocessing module is the central gateway for all data being used in preha. Especially considering the data analytics aspect of our application, it is desirable to have a lot of features available. In Section 2.2.3, we mentioned that preha supports data from different data sources including the electronic health record and the therapy planner. In order to get more features it is necessary to combine all obtainable data sources by joining them into a single data structure. This is the first core task of the preprocessing unit. A common characteristic in all data sources, the case id, is used as a key for join operations. So,

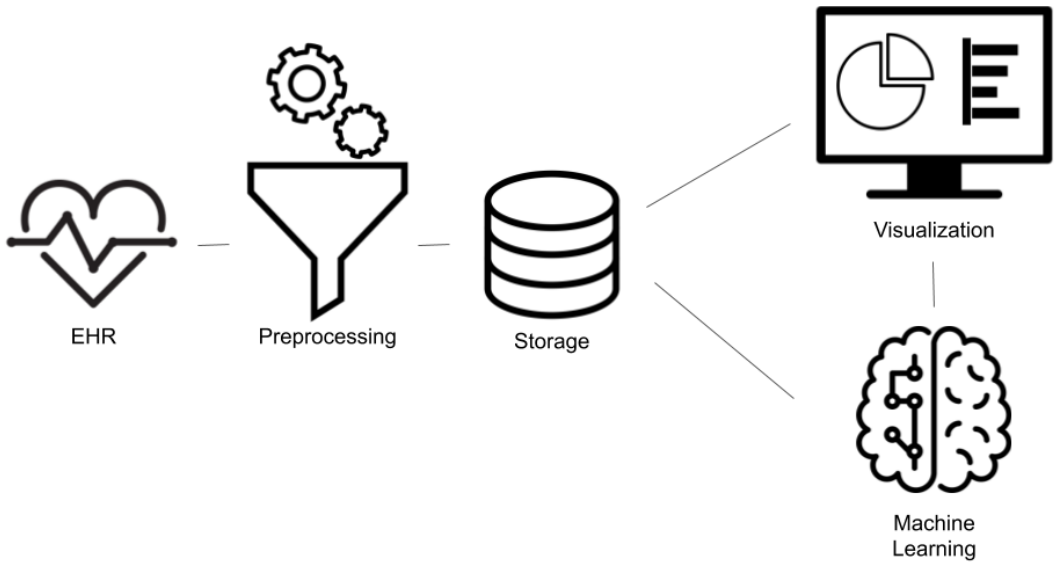

Figure 4.3. An overview of preha with all its modules. The EHR is not a part of preha, it is just displayed for the purpose of completeness.

before we can join the data sources, there a is need to standardize the quality of the data used in preha.

Up to this point, we already stated that dealing with data of poor quality is quite challenging. The better the quality of the underlying data is, the more reliable are the results of data analytics. Earlier, we exemplarily introduced cases for multi-dimensional data, missing data, or inconsistent data. This is rather a superficial point of view, as there are way more complex aspects of the dataset that can become problematic if not treated appropriately before the data is integrated in the application. Such data is often referred to as "dirty" data. Kim et al. [KCH+03] specify dirty data as either missing data, wrong data, or non-standardized representations of the same data. Furthermore, data is often not formatted in a way that allows data analytics to be performed right away. In our dataset, the data is either entered by a human (e.g., through forms) or automatically generated (e.g., by the REIS). It is the task of the preprocessing module to solve data quality problems by applying a variety of mechanisms that aim to address these issues.

The process just described is called data quality control [GAM+14, GE18] and is mandatory, before any kind of data analytics can be applied. Preha's preprocessing module covers all three tasks of data quality control: (1) data profiling, (2) data wrangling, and (3) data cleansing. *Data profiling* deals with the identification and communication of data quality problems. *Data wrangling* is about modifying the structure of the data to make it suitable for further processing (e.g., removing unnecessary rows or columns,

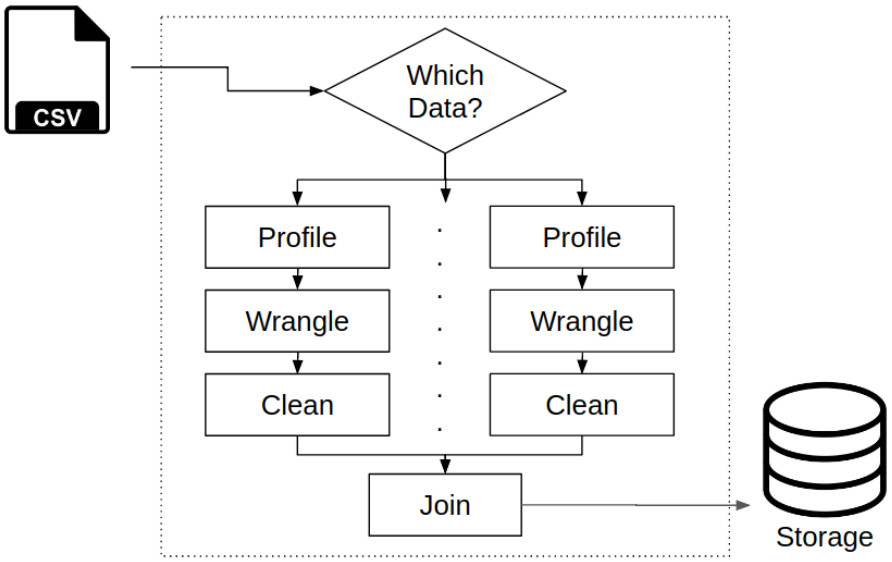

Figure 4.4. A schematic overview of the preprocessing module. The boundaries of the system are outlined by the dashed line.

splitting variables, merging data from different sources, etc.). As the name suggests, *data cleansing* or data cleaning is the process of correcting dirty data by repairing or removing it. The workflow of preha's data quality control process can be seen in Figure 4.4. In the upcoming sections we will describe how the three tasks are applied in our preprocessing module.

Data Profiling

Data profiling is the initial step of data quality control, focused on analyzing the given data, before any form of modification is applied. The goal of data profiling is the classification of dirty data, that allows the identification of errors and inconsistencies. Before we continue with the discussion of data profiling strategies that may be potentially used in the course of this thesis, we first need to define the types of potential quality issues in our dataset. Gschwandtner et al. [GGAM12] provide a comparison of taxonomies for dirty data, with a focus on dirty time-oriented data. This comparison is based on a meta review of several different approaches of taxonomies for dirty data. We will briefly introduce the two taxonomies that appear to be the most suitable ones for the context of this thesis, which is not focused on time-oriented data.

Kim et al. [KCH+03] introduce a tree-like taxonomy that is focused on three main categories of dirty data: missing data (e.g., the primary diagnosis is missing for a patient), not-missing but wrong data (e.g., a date in the format YYYY-MM-YYYY) and not-missing, not wrong but unusable data (e.g., inconsistent use of decimal separators in floating point numbers). These three categories further branch down to 33 leave-level dirty data types. A very positive aspect of this taxonomy is that there is a distinction between whether the dirty data could have been prevented or not. This is especially

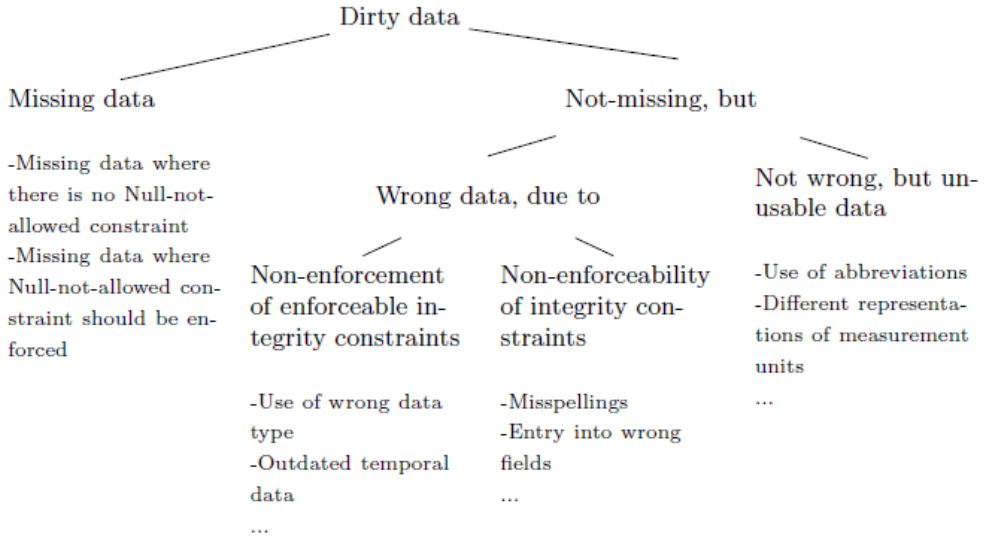

Figure 4.5. The classification of dirty data by Kim et al. [KCH+03]. The figure is taken from Gschwandtner et al. [GGAM12]

interesting for this thesis, as our dataset contains a lot of non-validated fields, which we consider as preventable dirty data. A disadvantage of this taxonomy is, that there is no way to describe a combination of more than one type of dirty data. An example for this would be data that is misspelled and in the wrong order. An overview of the taxonomy can be seen in Figure 4.5.

Müller et al. [MF05] provide a rougher classification of data anomalies (refer to Figure 4.6). At the top-level, they distinguish between syntactical anomalies, semantic anomalies, and coverage anomalies. Syntactical anomalies refer to errors in the data format and values such as irregularities (e.g., non standardized use of units) or domain format errors (e.g., a date of the format DD/MM/YYYY in a YYYY/MM/DD field). Semantic anomalies include inconsistencies like contradictions (e.g., a mismatch between date of birth and age) or duplicated entries. Furthermore, there is a category for missing values, the so-called coverage anomalies. These may be missing features as well as missing entries in a separate data source (e.g., there is no entry in the diagnosis table for a given patient). A big advantage is the more generic description of data anomalies, e.g., compared to Kim et al. [KCH+03]. The generic and not as detailed approach by Müller is sufficient for the scope of this diploma thesis. It is not our aim to discuss very specific issues with the data, as featured in Kims taxonomy [KCH+03]. Therefore, in the remaining part of this section we will refer to the taxonomy described by Müller et al. [MF05].

The next part of the preprocessing is finding out whether the data is dirty or not. There are several strategies that can be applied in order to detect specific kinds of dirty data. For example, domain format errors can be recognized by applying a regular expression on every entry of a feature. Entries that do not match the regular expressions can be seen as dirty.

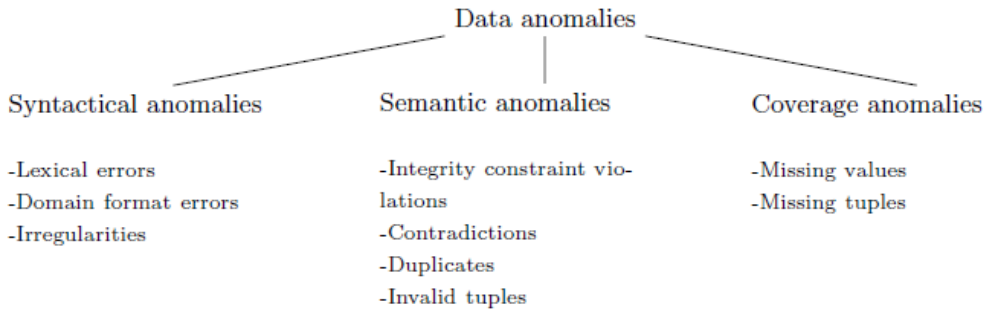

Figure 4.6. The classification of dirty data by Müller et al. [MF05]. The figure is taken from Gschwandtner et al. [GGAM12]

Patient Id	Case ID	Age	Sex
1	1	67	male
1	2	67	male
2	1	85	female

Table 4.1. Example for the structure of the demographic table.

Moreover, integrity constraint violations could be detected in scores by applying simple limit checks (e.g., the measured time of a 10 meter walking test can never be negative). While simple rules, like for the two examples described, can be automatically applied, they give no insight into the reason for the data quality issues. Also, assumptions need to be made for rules to be applied. In order to overcome this issue, it is recommended to involve data domain experts in the data profiling process [GAM+14, KCH+03, GGAM12]. Gschwandtner et al. [GE18] refer to these persons as "humans-in-the-loop". We conclude, that our data profiling strategy is to establish a set of rules for each feature, in order to keep the potential automation level very high for performance reasons. For the reason of preserving domain context, we validate our rules in cooperation with the engineers, the data domain experts in our case.

Data Wrangling

Before we describe the process of data wrangling in detail, the initial structure of the data will now be described. The EHR data is stored in a relational SQL database. As it is typical for SQL databases, the data is logically grouped in tables. Tables of the EHR used in this thesis are demographic information (see Table 4.1), diagnoses (see Table 4.2), therapies and scores (see Table 4.3). The information of all tables can be joined by using a case ID and a patient ID. The relation between the data in the tables is not *1:1* (e.g., one entry in the demographic table corresponds to one entry in the diagnosis table).

While the above described is a logical and consistent structure for a database, it is not desirable when it comes to predicting values as we need to re-structure all data to a unique tuple in a single table. For the purpose of our analysis this tuple is a case. In other words we want to have the stay of a patient at a facility with all recorded data in a

Patient Id	Case ID	Type	ICD-10 Code
1	2	Primary	J44
1	2	Secondary	I67
1	2	Secondary	E11.30
2	1	Primary	M17.9

Table 4.2. Example for the structure of the diagnoses table.

Patient Id	Case ID	Type	Admission Value	Discharge Value
1	2	ScoreX	170m	217meter
1	2	ScoreY	A-B	C
2	1	ScoreX	152m	167m

Table 4.3. Example for the structure of the scores table.

single row of a table. There are several features that need to be wrangled to match our desired structure. We aim to highlight the two most important ones.

At first, the focus lies on the diagnoses table. While each patient is associated with exactly one primary diagnosis, a patient may have from *zero* up to n secondary diagnoses. Each secondary diagnosis is stored in a separate row. The idea is to transform the information of the column *Type* that is currently distributed on rows, to a column information. This is easy for the primary diagnosis, as we face a *1:1* relation. We introduce a new column *Primary Diagnosis* that contains the ICD-10 primary diagnosis for each case[1]. A more complex solution is required for the *1:n* relation of the secondary diagnoses. In statistics this can be compared to categorical variables with multiple response options. The general approach to encode these variables is to represent the response options as columns of Booleans[2]. After this process is completed, we can remove the column specifying the diagnosis type. The result of all applied operations can be seen in Table 4.4.

Patient Id	Case ID	Primary[1]	Secondary_I67[2]	Secondary_E11.30[2]
1	2	J44	1	1
2	1	M17.9	0	0

Table 4.4. Structure of the diagnoses after data wrangling is performed.

Secondly, the state of the scores tables is quite similar. Each row represents a single health assessment performed for a patient. The *Type* column determines the kind of measurement that has been performed. *Admission Value* and *Discharge Value* columns contain the performance of the patients at the respective points in time of their rehabilitation process. As with the diagnoses, we face the case of a categorical variable with multiple response options, but this time we face other columns that are associated with it. For this reason, we create an admission (postfixed A) and discharge (postfixed D) column for each score. After this process we delete the columns *Type*, *Admission Value* and *Discharge Value*.

Two important observations are to be mentioned here. The introduced strategy can lead to a dramatic increase of columns in the data structure, while the number of rows is reduced. As only a subset of assessments is performed for a patient (largely depending on primary diagnosis and discipline) a major part of each score column will contain *not available (n/a)*. Encoding non-performed assessments as 0 is restricted in these columns, as it can be interpreted as "this test was performed with 0 as result" instead of "this test was not performed". The downside of this approach is that scores that should have been performed, but are missing for some reason can not be represented. In the original dataset, these values were encoded as rows with *n/a*. The scores after data wrangling can be seen in Table 4.5.

Patient ID	Case ID	ScoreX_A	ScoreX_D	ScoreY_A	ScoreY_D
1	2	170m	217meter	A-B	C
2	1	152m	167m	n/a	n/a

Table 4.5. Structure of the scores after data wrangling is performed.

After we performed all of the above described operations, we are now facing datasets that are reduced to a single case per row. Though, the number of columns has increased, as already mentioned. The final step of data wrangling is to join all available data into a single tabular structure with one case per row. A join operation associates rows of separate tables with each other by a specified criterion. In our case this criterion is that the case ID and patient ID of the tables to be joined must match. Table 4.6 displays the final data structure. For layout reasons, secondary diagnoses are abbreviated as *Sec* and scores as *Sco*. It can be observed that the case with case id 1 and patient id 1 is not present, as there are neither diagnostic nor score data exist for this case. This case may occur if patients have just started their rehabilitation and only demographic data is available. Finally, we want to point out again, that the descriptions above are just a subset serving as examples for all wrangling operations performed.

Data Cleansing

Data analysis is only as good as the quality of the data in the underlying dataset [GAM+14]. In order to improve this quality, data cleansing can be applied as its aim is to repair or remove dirty data. In the course of our interview sessions, one engineer stated that *"80% of effort in data analytics is spent on data cleaning"*, a figure that is also mentioned in the literature [McK12]. Without verifying this statement, we can presume that data cleansing is an important part of the data quality control performed for this thesis. Furthermore it is the final step before the dataset can be used in preha. Müller et al. [MF05] suggest to include a domain expert in the process of data cleansing as the correction of anomalies

Patient ID	Case ID	Age	Sex	Primary	Sec_I67	Sec_E11.30	ScoX_A	ScoX_D	ScoY_A	ScoY_D
1	2	67	male	J44	1	1	170m	217meter	A-B	C
2	1	85	female	M17.9	0	0	152m	167m	n/a	n/a

Table 4.6. Final structure of the data after joining.

requires detailed domain knowledge. Again we face the challenge of balancing automation level and the gold-standard, i.e., domain knowledge, like in the data profiling process.

To the best of our knowledge, there are no generic approaches for data cleaning. Dirty data can have multiple causes, so there is no standardized treatment for it. What we know for our dataset is how the correct data should look like for each column. The idea of our cleaning program is to perform several cleaning steps that are implemented for each type on the scale of measurement (as introduced in Section 2.2.3) individually. In each step, a rule is applied to the data, e.g., "all characters must be upper case". Furthermore, each cleaning program features boundaries for further validation (e.g., a value of the Barthel index must be in the interval [0, 100]). A feedback loop for refinement is provided via a result after the cleaning process that displays the percentage of successfully validated values in the dataset. If the value is too little, new rules may be introduced.

4.4 Storage

Preha's storage is where the entirety of data used in the application is persisted. All data stored here has been preprocessed and fulfills preha's data quality requirements. As already stated in the description of data wrangling in Section 4.3, all data stored for the use in the Visual Analytics dashboard or the data analytics engine is in a single table structure.

In order to determine a technology for the data storage, we will first highlight the scope it is being used in. McKinney [McK12] suggests to consider a data storage based on performance, data integrity, and scalability needs. Initially, we will highlight the *performance* requirements. The total size of the dataset used in preha is about 150 MB. This is a relative small data size, so we can state that the needed disk size for the dataset is not a major requirement. Another important aspect to discuss, when it comes to deciding on the data storage, is the access to the data. We presume that our application and therefore its data storage is only accessed by one user at a time. As the data is used for Visual Analytics and data analytics, speed is a critical facet. In order not to interrupt the users thinking process, visual feedback for the described operations should be delivered within seconds [CRM91] or even less than 100 ms [Shn94].

In terms of *data integrity* we do not consider access control to the data storage. In a real-world setting there would be a lot of aspects to consider, especially in terms of data protection. An example for this is restricting the access to the data from the application to case id level. Such cases may occur if a domain expert of facility A wants to access a case from facility B. After the preprocessing stage is passed, all data in preha is read only.

We consider *scalability* as out of scope for this thesis, because the dimensions of the dataset are already known before the application-design stage. The data used in preha will not increase over time.

After considering all of the factors described in this section, we conclude that speed is the only real restriction for the architecture of our data storage. For this reason, we will store preha's data in a plain and simple CSV file. The simple structure of this format enables a high speed for read operations, as it does not comprise any overhead (e.g., a query language).

4.5 Machine Learning

The main goal of this diploma thesis is to enable precision rehabilitation by utilizing data from large and complex datasets. Data analytics operations can be performed on existing data in order to determine patterns or discover correlations between features in the dataset. To establish precision rehabilitation, we need to implement means that enable the prediction (approximation) of rehabilitation outcome values for individual patients, based on the characteristics of the clinical case (1) and calculation models based on statistical analysis of the dataset (2). The deliberations in this section relate to potential approaches for preha's machine learning module.

The task described above is located in the field of machine learning. The name machine learning was coined by the American scientist Arthur Samuel [Sam59], a pioneer in the field of artificial intelligence. Machine learning is about analyzing stored data and turning this into information a human is able to understand [Alp10]. What makes machine learning so special is that the computer performs such tasks based on patterns in the data without following rules that were specified beforehand. The approach of using machine-learning algorithms for problems in the medical domain is not new. Stein et al. [SBS+15] apply logistic regression on measurements such as the Barthel index to predict rehabilitation needs. Singh et al. [SNG+15] develop predictive models to predict risk of renal function deterioration by analyzing temporal EHR data. In the present case of rehabilitation, we face a large dataset with high-dimensional data. What we know about the data is that it is not completely random. There are certain patterns occurring between the features. This is where machine learning comes in. With the help of machine learning, we can construct a good and useful approximation of those patterns. Even though we can not identify all patterns with 100% accuracy, we can make use of them to create predictions for specific features in the future, under the assumption that the future data does not differ much from the data used for machine learning.

Regression

Prediction is a task with its origin in statistics, where it is called regression analysis. The idea of regression analysis is predicting an output value Y based on a number of given input attributes X [Alp10]. Y is called the dependent variable, X are called independent variables. Regression analysis can solve this task by fitting a function to a given dataset including X and Y. The process of fitting the function is called training. In simple linear regression, this function can now predict Y as a function of X for a given observation i:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Y_i = Predicted Y value for observation i
β_0 = Estimate of the regression intercept
β_1 = Estimate of the regression slope
X_i = Value of X for observation i
ε_i = Random error component for observation i

Figure 4.7 illustrates the linear regression model. The horizontal axis represents the independent variables X, the vertical axis the dependent variable Y. The blue data points are actual values of Y from the underlying dataset. The red line is the function of our linear regression model. β_0 is the intercept of the regression. β_1 is the slope of the regression. The green data point is the predicted value of Y_i for X_i. ε_i is the difference for observation i between the predicted value of Y_i and the actual value of Y_i.

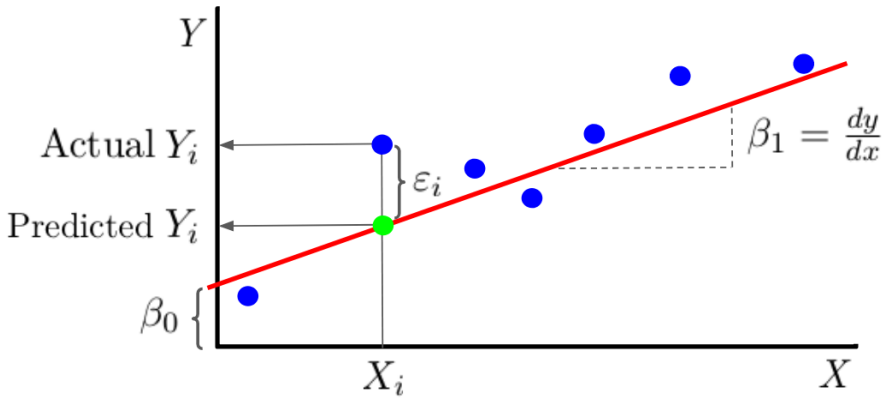

Figure 4.7. The linear regression model.

To give an example of the use of regression analysis within the scope of this thesis we will now discuss a concrete case. The goal is to predict the Barthel index [KEP12] at the discharge phase of the rehabilitation process. This is our dependent variable Y. Our independent variables X, on the other hand, are all other features in the dataset associated with the patient at the very beginning of the rehabilitation. Among these features are for example age, sex and the primary diagnosis of the patient. As the given example concerns a dataset with a high number of cases, we have numerous sample entries comprising both X and Y variables. Our regression model can now be trained on these samples. Eventually, the output algorithm can now make a prediction for the Barthel index at discharge. After the actual Barthel index is entered in the EHR, the delta between predicted and actual value can be calculated, representing our ε.

Classification

Besides regression, there is another important machine learning problem to mention in this context called classification [Alp10]. While regression aims to predict the value of a

dependent variable, classification predicts a discrete class of a given dependent variable. An example from the medical field would be the classification of a tumor as benign or malignant, based on independent attributes of the tumor. This example can be seen in Figure 4.8. Independent attributes in our example are the age of the patient and the size of the tumor. We encoded the real state of the tumors as follows: malignant tumors are visualized as red circles, while benign tumors are green crosses. The black curved line is the classifier trained by the machine learning algorithm. It can clearly be seen, that while most tumors got classified correctly, one tumor was misclassified as malignant, while it is actually benign.

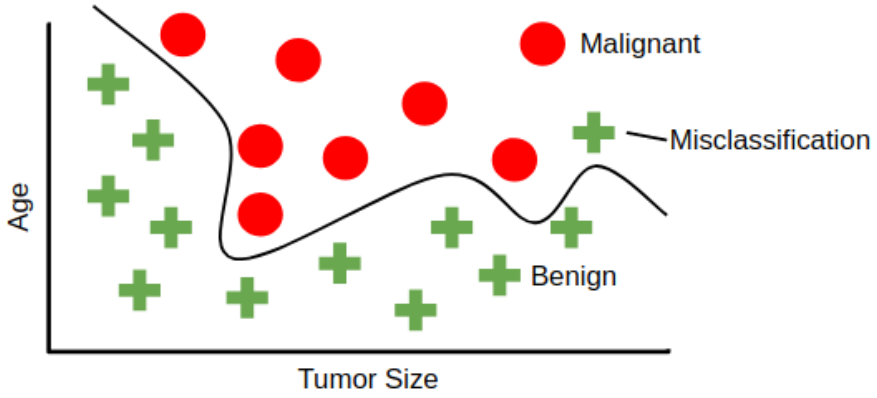

Figure 4.8. An example of a classification problem.

Classification and regression are considered as supervised machine learning [Alp10]. Supervised learning means, that an input is mapped to an output based on example input-output pairs, to train an algorithm that is aimed to solve a given classification or regression problem. If the algorithm is very precise, but highly complex with a low performance we speak of an *overfitting* algorithm [Alp10]. Overfitting algorithms perform very good on the data from the original dataset they were trained on, but may perform poorly on yet unseen data points. If the algorithm is rather simple and fast, but lacks in precision the algorithm is called *underfitting* [Alp10]. Obviously the goal is to achieve an optimum level between overfitting and underfitting. At this point we conclude that there are two kinds of prediction tasks we need to perform, namely classification and regression.

A multitude of machine learning algorithms can be applied to the proposed problem. Jensen et al. [JJB12] analyze the usage of data mining methods on EHR data. They highlight naive Bayes estimation [FTHW00], artificial neural networks [Yao99], support vector machines [Gun98], and random forests [Bre01] as suitable supervised machine learning algorithms. We consider the naive Bayes algorithm as unsuitable for our problem, as it performs poor on regression problems [FTHW00]. *Support vector machine* algorithms [Gun98] use hyperplanes to do classification and regression tasks in an N-dimensional space, where N is the number of features used. A hyperplane is a

subspace whose dimension is one less than that of its ambient space. An *artificial neural network* [Yao99] can be imagined as a pipeline with several stops, each of which is trained to make decisions based on a feature in the dataset. *random forests* [Bre01] use a randomized set of decision trees to solve classification and regression problems. All of the above described algorithms can be applied to our prediction problem. We select random forests, as we have previous experience with it.

Random Forests

The first proper introduction of random forests was made by Leo Breiman [Bre01]. As the name suggests, a random forest is composed of several decision trees. These decision trees are not all trained in the same way, but each one is provided with a random subset of the training data. This procedure is referred to as *bagging* and it enhances the accuracy and stability of the prediction. Furthermore, the bagging can already estimate an error for the subsets.

We will explain the random forest algorithm in detail with regards to its task in the course of this thesis. As all supervised machine learning algorithms, the random forest requires a dataset with independent variables X and the associated dependent variable Y. For the training of a supervised machine learning algorithm, the existing dataset is split up in two parts. The first part is the *training* dataset, holding the majority of the values in the original dataset. This part of the dataset is used to train the decision tree. The training dataset is now randomly distributed for each decision tree, whereby the number of trees is usually specified before the training. Now Y can be predicted, by using X as input parameters for the prediction of the random forest, whereas the algorithm predicts Y for each tree. The output of the algorithm is the combined results of each tree in the forest. For a classification the predicted class would be the majority of predicted classes (see Figure 4.9). For a regression it would be the average of the predicted values (see Figure 4.10).

After the model is trained, the remaining dataset, called *validation* dataset, is applied to the random forest, in order to validate its performance. A prediction is made for each entry in the validation set. Parameters used to assess the performance are the accuracy or the average error of the predictions. One of the biggest advantages of random forests is that they enable the determination of *variable importances*. The variable importance defines the contribution of an independent variable in X to the prediction of the dependent variable Y. Referring to our example from regression, we can derive the connection between age, sex, or the primary diagnosis of the patient at rehabilitation admission and the predicted Barthel index at discharge.

4.6 Visualization

We will now introduce the main part of our Visual Analytics application. The visualization component decides on how the users consume information, it is the user interface that connects the users with the underlying dataset. The visualization interface is a crucial

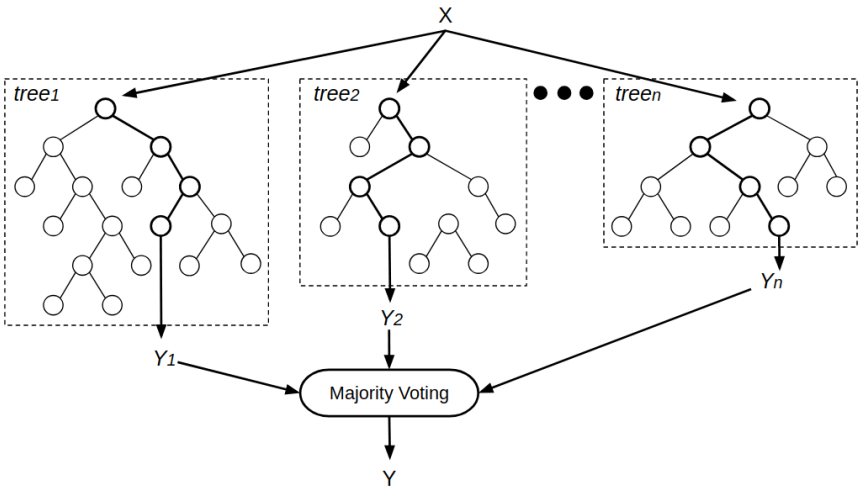

Figure 4.9. An example of a classification random forest.

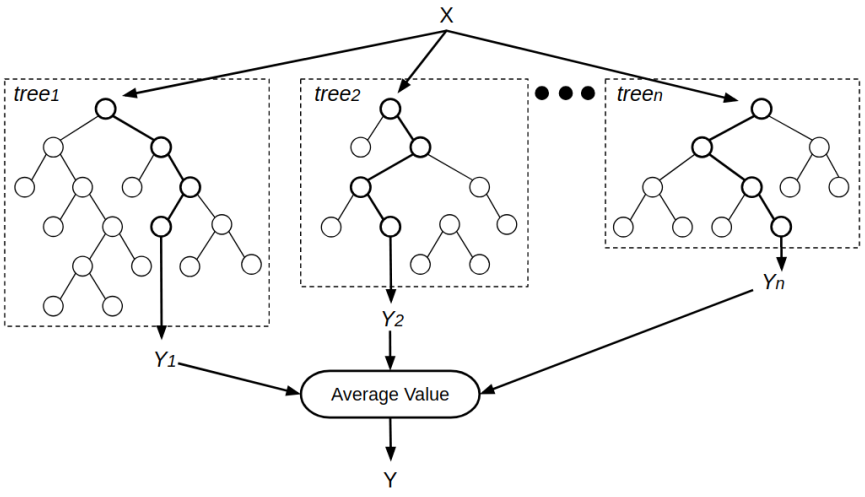

Figure 4.10. An example of a regression random forest.

component, therefore thorough examination of possible implementations needs to be carried out. Our domain experts are the key users to derive knowledge from the data, but they are not technologists. A rather simple minimalistic approach is needed to enable Visual Analytics for them. This is where analytics dashboards come in. One major design decision is taken here: the user interface is realized as a Visual Analytics dashboard.

As already mentioned in Section 3.1, dashboards exhibit a visual data representation in a tiled layout of simple charts. In health organizations, dashboards are in use for high-level (hospital management) and detailed patient-level purposes [SCB+19]. Our visualization component needs to be designed for multiple tasks, as introduced in Section 2.2.3. A possible issue in this context is that the tasks may change over time. This could be caused by new visualization technologies, changes in the workflow or issues with existing visualizations. In a *static environment* we would now have to update the visualization and introduce these changes to all users. Or, if this change is only demanded by parts of the users, a new software branch would be created which needs to be maintained from that point in time. However, a *dynamic dashboard* is highly adaptable and capable of supporting the needs of multiple users and multiple tasks. There is no need in maintaining multiple versions. Though, a possibility for creating dashboard templates is required. This minimizes the effort for creating new dashboards that slightly differ from existing ones.

Another important reason for using dashboards is the interchangeability of visualizations. It is not our aim to design a static visualization that fulfills a single purpose for a long period of time. The flexibility of dashboards enables changing of visualizations, while preserving the overall functionality of the dashboard. Interchangeable visualizations are a dynamic way of enabling data analytics from different points of view. Our goal is to utilize a Visual Analytics technology that can be integrated in dashboards.

4.6.1 General Design and Interaction

Generally, we aim to stick to the basic principles and techniques of Visual Analytics, as described in Section 3.1.1. These principles and techniques enable our Visual Analytics dashboard to fulfill the three requirements of the Data–Users–Tasks triangle by Miksch et al. [MA14]—*Expressiveness*, *Effectiveness*, and *Appropriateness*—in the best possible way. Ratwani et al. [RF14] and Sarikaya et al. [SCB+19] describe common design principles used in dashboard design. The first is *visual information seeking mantra* [Shn96], i.e., overview first, zoom and filter, then details on demand. We propose to fulfill overview first by setting default views that are very aggregated. Zooming in a dashboards visualization enables the dashboard interpreter to gain more information, without changing the context. Filtering on the other hand applies a constraint to the underlying data, that changes the data basis for the visualization. The *visual analytics seeking mantra* [KAF+08] suggests to perform an initial analysis that shows the important aspects of the visualization. All dashboards used in our visualization will have an initial state that marks the start for all analysis actions. Using *Multiple (Coordinated) Views* [WBWK00] is an aspect, that can easily be achieved with dashboards, as they already consist of multiple views.

In the implementation of the dashboard, we need to assure that all visualizations are permanently based on the same data. *Overview + Detail* [CKB09] can be realized by combining various types of visualizations with different levels of detail in a dashboard. Additionally, the interaction techniques of information visualization, as described by Yi et al. [YKS07], are considered for the design of the dashboard:

Select enables the user to mark something of interest and track it when changes in the underlying data occur. We design our dashboard to realize selecting by implementing a hover function on the visualization. When the cursor is hovered over a specific segment of a visualization, this segment is highlighted and additional information is displayed.

Explore is a technique that makes it possible to focus on different subsets of the data, for example by adding or removing data items from a view. It is our aim to realize the explore technique by using filter elements that can be toggled on and off by a single click.

Reconfigure provides different perspectives to the user, as the spatial arrangement of visualizations is changed. We realize reconfigure by providing means to arrange data in a specific order (e.g., alphabetic, by value, ...) for multiple visualization types. Moreover, the arrangement of visualizations as such in the dashboard shall be easily changeable by simple drag and drop mechanisms.

Encode means to modify the fundamental visual representation, including its visual appearance (e.g., size, shape, color, ...). All visualizations used in the dashboard must enable these modifications before and after they are added to the dashboard.

The technique *Abstract/Elaborate* is used to modify the level of detail visualizations employ for data representation. This technique shall be applied to all visualizations that are capable of presenting various levels of details, like maps or hierarchical visualizations.

Filter is used to set conditions on the underlying data, enabling the users to only view a particular subset of interest. This is probably the most important technique in the context of this thesis, because we aim to use it for creating subcohorts of data. There shall be two methods of creating filters on the data. The first method creates filters on the dataset by selecting segments of interest in a visualization, which is more focused on the needs of the domain experts. This eliminates the need for textual filter queries that requires a significant amount of training for inexperienced users [RF14]. The second method provides the possibility of issuing such queries for the engineers, as this is a rather fast way of filtering data for experienced users.

4.6.2 Task Design Implications

In order to take into account the tasks defined earlier in this thesis in Section 2.2.3, we will now analyze the design implications of each task. As each task is different, it relates to an individual dashboard. In the following section we will highlight the requirements task by task and how we will address them in the design of our application. The specifications on how the tasks are implemented will be given in Section 5.4.

Eng1: Meaningful partitioning

The aim of this task is to generate meaningful partitions of the entire dataset. These partitions can be used by the domain experts for clinical research and may even be exported to other tools. We propose to use the CSV format to do so, as it is a commonly used standard for exporting tabular data. The input for this task is the entire underlying data and instructions on how the resulting subcohort is defined.

As stated by a domain expert in the interviews, the data is filtered according to common characteristics like age or geographical location of the patient. As already mentioned in the section above, the possibility for textual queries is present in the dashboard for experienced users. These could be used for creating the corresponding subcohort. The major advantage of textual queries is that they can easily be saved and reused with a single click. In order to provide visual feedback to the engineers while creating the subcohort, we add visual representations of the current subcohort. A hierarchical visualization of diseases grouped on multiple levels, represents how the primary diagnoses of the corresponding subcohorts are structured. We will further elaborate on the visualization used for hierarchical structures in Section 4.6.3. As the output of this task will be a tabular structure, a data table with a preview of the output will be included. Detailed information on the encodings and implementations of data tables will be introduced in Section 4.6.7. Another important feature used for filtering that was mentioned in the interviews is the geographical location of the patients. In order to provide an appropriate presentation of this, we use map visualizations that show the location of the corresponding patients. Details on map visualizations may be found in Section 4.6.8. Finally we provide a chart that shows the individual values of the categorical variable sex. Our design decisions on charts for categorical variables can be seen in Section 4.6.4.

Eng2: Assessment templates

In order to inform the patients on therapy progress, their assessment scores are presented to them. It is the engineers job to provide dashboard templates for this purpose, that the clinicians interact with. The actual results of the assessments are displayed to the patients as simple metrics. The metric visualization is described in Section 4.6.9. In order to give the patients a sense of how they compare to other patients with similar characteristics, a distribution chart is used for the assessment data, as described in Section 4.6.6. The way the visualizations are prepared is critical for this task, as patients are not used to interpret complex charts. All used visualizations need to be annotated (e.g., Barthel index at admission), so it becomes clear to the patient what data they are shown. Furthermore, all visualizations need to be arranged so that admission and discharge visualizations are clearly separable.

Eng3: Benchmarking templates

Again, it is the engineers' task to prepare dashboard templates for domain experts. The purpose of this dashboard is to provide data on clinical efficiency. Four visualizations are included in this dashboard. A metric visualization, as shown in Section 4.6.9, that displays the total number of patients in the current selection. A distribution chart, as

shown in Section 4.6.6, that displays the development of patient admissions over time, grouped by each facility individually. Another distribution chart displays the development of characteristic patient assessments in the very same time frame. Finally, the dashboard includes a categorical visualization, as shown in Section 4.6.4, of the top ten payers of the rehabilitation, ordered by number of patients.

Eng4: Outcome predictions

The purpose of this task is to predict the discharge value of an assessment for a specific subcohort of patients. The engineers need to write textual queries that are used to define this subcohort. If the query is changed, a new machine learning algorithm is exexuted and its results are displayed in a dedicated visualization in the dashboard. This visualization includes metrics, as shown in Section 4.6.9, for the predicted value, the accuracy of the prediction, and the prediction error. Additionally, the importance of the variables for the prediction are displayed in a categorical visualization, as shown in Section 4.6.4.

Exp1: Outcome presentation

It is the domain experts' task to present the rehabilitation outcome to the patients. The dashboard created by the engineers in Task Eng2 is the basis for this task. The interaction of the domain experts with the presented visualizations is limited to setting the filters to a subcohort that corresponds to the respective patient. For example, a neurological male patient at the age of 75 is shown the typical results of his corresponding subcohort.

Exp2: Clinical benchmarking

Similar to Exp1, Exp2 is also based on a dashboard prepared by the engineers in Eng3. The domain experts can apply certain filters to the data and monitor the corresponding results. For example, how the developments of specific assessments differ among the rehabilitation facilities can be evaluated by selecting the corresponding segment from the distribution chart. Navigating through the time frame can also reveal additional details on the development of assessment or admission figures.

Exp3: Clinical exploration

Compared to the other tasks, this task is not defined very strictly. The aim of Exp3 is to provide the clinicians with the tools to explore the dataset. This allows the clinicians to utilize all possibilities of the Visual Analytics dashboard without any constraints. Possible actions include encoding data in visualizations, creating personal dashboards and defining custom queries.

Exp4: Clinical analysis

This task is aimed at discovering the dataset of the subcohort created by the visual queries defined by the engineers in Eng1. The domain experts are mainly supposed to interpret the data extracted in the tabular structure. This mainly means locating measures of interest in the dataset and comparing them across the subcohort. Filtering actions can be performed to view results more individually or to refine the subcohort.

Exp5: Intervention planning

This task is where the domain experts interact with the machine-learning module. The domain experts validate the response of the machine-learning module to changes in the dataset (e.g., a reduction of age). This further enables them to adjust the treatment plan of specific subcohorts that lead to a better rehabilitation progress. For this task, the dashboard template created by the engineers in Eng1 is reused. The machine learning visualization is added to the visualization. Furthermore, a categorical visualization that displays the number of patients per facility is added. We further modify the visualization to also include sub-categories for each facility, i.e., the corresponding medical discipline of the rehabilitation. We refer to this type of visualization as detailed categorical visualization, as described in Section 4.6.5.

To address all tasks that have been defined, a multitude of visualization types and combinations thereof are needed. In the upcoming subsections we will introduce all visualization types that are used for the introduced tasks.

4.6.3 Hierarchical Visualizations

As already discussed in Section 3.3, visualizing hierarchical structures is a common task when it comes to analyzing data stored in electronic health records [ZGP15]. There are multiple approaches for visualizing this kind of data. The two most important approaches are the sunburst diagram and the treemap.

The *sunburst diagram* was introduced by Stasko et al. [SCGM00]. The parent-child relation of hierarchical structures is displayed on the base of a radial chart. At the center of the diagram is the highest parent of the hierarchy as a circle. All children of this parent are aligned as a ring around this inner circle, like pie slices. The size of each child, or the angle each child takes up, determines its weight (e.g., the number of children of this child). Then again, each child has its own children that are aligned as the next outer layer. Figure 4.11 displays an example for a sunburst diagram. It is easily perceivable that the red child has the biggest share compared to its siblings. The size of a sunburst diagram depends on the depth of the hierarchy it represents.

Contrary, the *treemap* is built on a rectangular base. It was introduced by Shneiderman [Shn92] and Johnson [JS91]. A parent-child relationship in a treemap is displayed as a rectangle (parent) that contains the children as nested rectangles. Again, each element in the hierarchy has a weight (e.g., the number of children of a child), which determines the area of the rectangle. Usually, the children are assigned to a parent in the order of their weight, starting with the largest child on the top left and ending with the smallest child on the bottom right. There are several tiling algorithms, the squarify algorithm is commonly used to keep the rectangles as close to squares as possible. If the children of a parent are imbalanced in terms of size, it is also possible to represent the smaller child in a combined rectangle as “others”, e.g., as done by Krause et al. [KPS16]. It is important to mention that the area of a parent is always occupied by its children to 100 percent, so

Figure 4.11. Hierarchy as sunburst diagram [Rib].

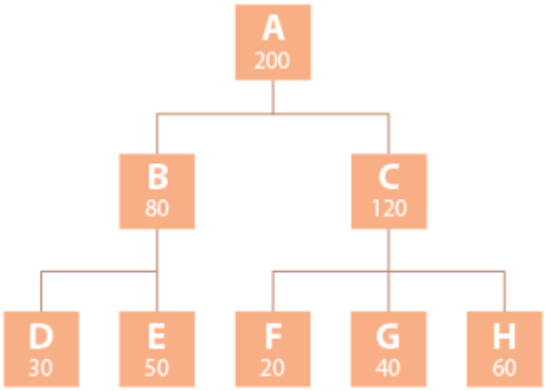

Figure 4.12. Hierarchy as tree [Rib].

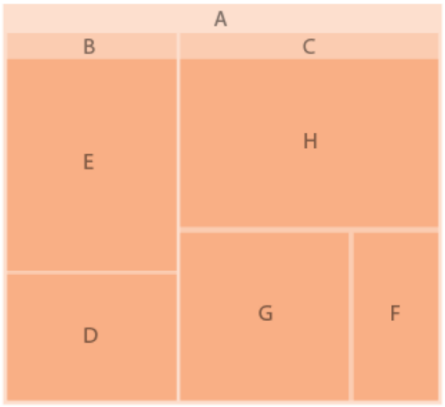

Figure 4.13. Hierarchy as treemap [Rib].

there is no free space. Figure 4.13 shows a treemap representation of the tree structure that is displayed in Figure 4.12.

Stasko et al. [SCGM00] evaluated the performance of sunburst diagrams and treemaps based on two experiments. Showing that the sunburst diagram outperformed the treemap in terms of accuracy and in the temporal domain. The participants in their study preferred the circular technique. Opposingly, Munzner [Mun14] states two issues with radial diagrams, such as the sunburst diagram. The first is, that the projection of information to an angular channel is not as accurately perceivable as information in a rectilinear spatial position channel. Secondly, the start and endpoint in the angle channel are always the same, which may be misleading for nonperiodic data. Munzner still recommends using sunburst diagrams for periodic information.

Both visualizations described have a common issue. The deeper a hierarchy is, the more space they use. This is an issue for using these visualizations in dashboards, as dashboards comprise of an arranged set of visualizations. If such a diagram would be used in a

dashboard, it would be necessary to adjust the size of the visualization with each change in the underlying dataset (e.g., by filtering a subset of data with less children). Blanch and Lecolinet [BL07] address this issue by introducing the zoomable treemap. This type of treemap visualization has a fixed size. Being an interactive visualization, the zoomable treemap enables the user to select a child of a treemap, subsequently changing its focus, so that the selected child is the new parent. All rectangles shown in the treemap are now children of this new parent. Again, these children can be selected until the bottom of the tree is reached. This approach perfectly aligns with certain fundamental techniques of Visual Analytics: the *visual information seeking mantra*, the *visual analytics seeking mantra*, and *Focus + Context*. Therefore, we include the zoomable treemap in preha for the visualization of hierarchical structures.

4.6.4 Categorical Visualization

Sometimes it is a data analyst's task to interpret very simple data [RSN+19, KPS16], like the average weight of different categories of animals. This task is very well suited for *bar charts*, as they are good in providing information appropriately for the human perceptual system. Two different kinds of sensory modalities are perceived by the system [Mun14]. The first is the *identity channel*, telling us what something is or where it is. On the other hand, there is the *magnitude channel*, which gives us information about quantity. A bar chart handles the identity channel by presenting the categories of the analyzed data in separate regions that are associated with a key. There is one key for every category. In simple, two dimensional, bar charts these regions are arranged in a one-dimensional way, either horizontally or vertically. The other dimension, the height of the bar, is used to display the magnitude channel.

Figures 4.14 and 4.15 show two examples of bar charts that utilize the animal example mentioned above. The horizontal axis aligns the type of animal, each of which is assigned a separate bar. All bars are positioned within a common frame. The height of the bar depends on the average weight of the animal in lbs. The width of each bar in a bar chart is independent of the magnitude that is being displayed, therefore all bars are of equal width. Hence, both the bar height as wall as the bar area can be perceived, which further improves the humans ability to interpret the data.

It is easily perceivable that the categories of animals are ordered differently in the two figures. Figure 4.14 arranges the position of bars alphabetically. This design choice makes it easy to look up the weight of the animals by their name, but the context of how the weight is aligned within all animals can get lost just like other meaningful patterns in the data. Figure 4.15 on the other hand displays the animals sorted by their average weight. Trends can easily be seen in this data-driven encoding style. Comparative tasks are also easier, while finding an animal among a large number of bars can become more difficult.

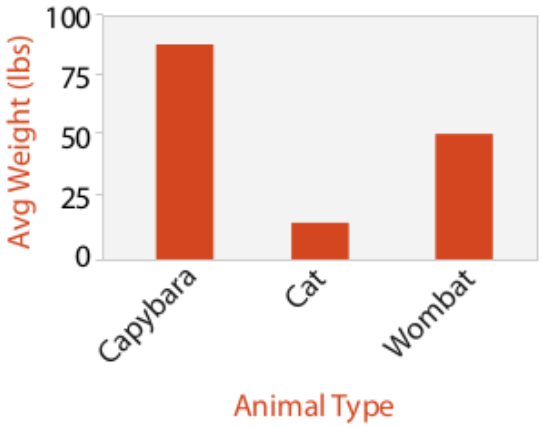

Figure 4.14. Bars sorted by label [Mun14].

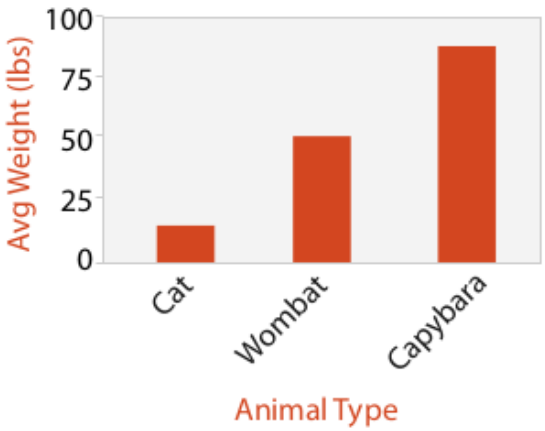

Figure 4.15. Bars sorted by size [Mun14].

Generally a bar chart is well suited for comparing values, but there are limitations:

- *Similar size:* This is especially an issue if the bars are not sorted by size. Bars that are of similar size can become hard to distinguish. For this reason, it may be helpful to use lines that are perpendicular to the magnitude dimension.
- *Scalability:* Bar chart data is usually analyzed on a two dimensional surface of limited size such as computer screens or paper sheets. Several or even dozens of bars can be handled, hundreds on the other hand can cause problems in terms of perception.
- *White space:* There must be a sufficient amount of white space between the bars in the chart in order to differentiate them.

Up to this point, we were talking about bar charts with two dimensions: a category and a magnitude of data. However, there are additional ways of imparting information through a bar chart. One possibility is to assign colors to additional categories of the data. In the animal example, we could add information on animal classes by coloring the bars (e.g., mammals in yellow, fish in blue, ...). Of course, there must be a legend present that contains this information.

4.6.5 Detailed Categorical Visualization

Another option to display multi dimensional data are *stacked bar charts* [Mun14, KAF+08, ANI+17]. The idea behind the stacked bar chart is to encode information by vertically stacking multiple sub-bars to one bar in the chart. The overall length of the bar still encodes a value, just like in any other bar chart. Therefore, the information of two keys can be encoded. The first one determines the bar where the information can be found. The second key addresses the sub-component of the bar and is color encoded for

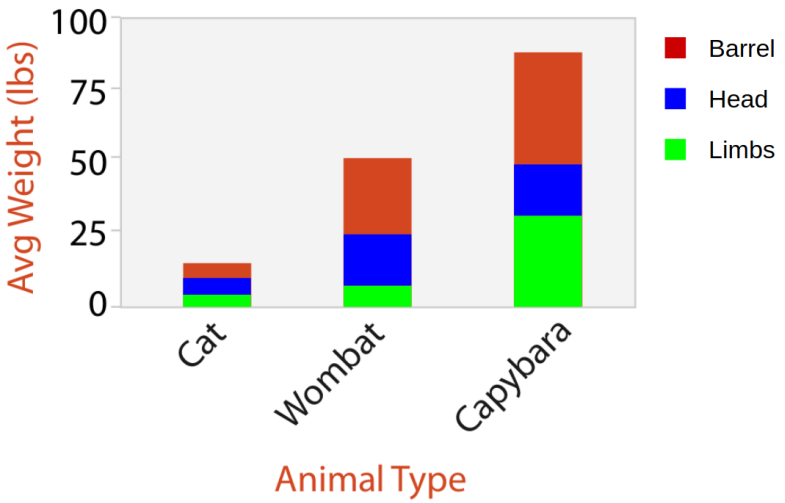

Figure 4.16. Example for a stacked bar chart. Average weight of animals divided by anatomical structure.

each category it represents. The color encoding, enables the user to identify the sub-components. Furthermore, the colors enable the user to look up the same sub-component in a different bar. Figure 4.16 displays an example for a stacked bar chart. The stacked bar chart is based on the simple bar chart of Figure 4.15. Contrary to the simple bar chart, which just provided the information of the average weight of the corresponding animal, the stacked bar chart of Figure 4.16 shows us the average weight partitioned by average weight of the barrel, head and limbs. Please note that these partitions are made up and do not correspond to the proper average weight of the respective anatomical structures.

The stacked bar chart is able to give the interpreter a good sense of the part-to-whole relation of a sub-compartment. Basic functionalities, such as looking up values, comparing bar heights or finding trends, remain untouched. Obviously, the order of the sub-compartments needs to be the same for all bars. A negative aspect of the stacked bar chart is that while the bottom sub-compartments can be easily compared to each other, the next one is harder to compare, as it has a different starting point in each bar. Moreover, the decision on the second key must be made carefully, as not all first key entities may include all sub-compartments. Referring to our example figure, it might not be a good idea to add snakes to the comparison, as they do not have a comparable anatomical structure.

4.6.6 Distribution Visualization

Dot charts display a quantitative attribute against a categorical attribute using point marks. A dot chart can be seen as a bar chart where the quantitative value is encoded as a dot instead a of bar. A *line chart* on the other hand also visualizes the connections between dots through line segments. Line charts are quite frequently used to encode a

Year	Weight (lbs)
2004	5
2005	7
2006	9
2007	10
2008	12
2009	16
2010	16
2011	12

Table 4.7. Weight development of a cat over the years.

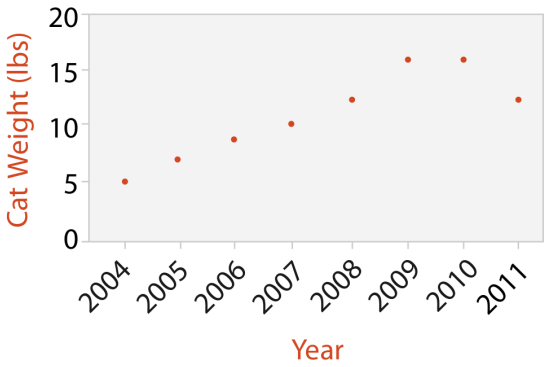

Figure 4.17. Simple dot chart [Mun14].

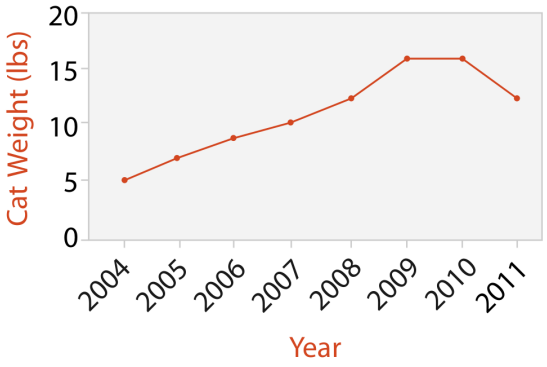

Figure 4.18. Simple line chart [Mun14].

quantitative information that is encoded on the Y-axis by connecting data points aligned to corresponding values on the X-axis, which is often temporal. Data shown in dot charts and line charts can also be visualized as a tabular structure. Additionally, it is possible to encode the quantitative values of multiple instances in a single dot or line chart, by using coloring to indicate different entities. As in the previous subsection, we will use animal weight, specifically cat weight, as an example for our chart. The data can be seen in Table 4.7.

Figure 4.17 shows the data from Table 4.7 as dot chart. Each dot can be projected on two axes, the Y-axis displays the cats weight, while the X-axis corresponds to the year of the measurement. The dot chart displays the data points as isolated measurements at specific points in time. No information on the weight between the measurements is implied. Contrary, a line chart for the same data can be seen in Figure 4.18. The measurements from the table are projected as dots in the same ways as they are in the dot chart. The difference in the line chart are the line segments that connect the dots. Both charts allow us to track the development of the cats weight. There is a constant increase from 2004 to 2009, followed by stagnation to 2010 and a decrease until 2011. It is harder to interpret the extent of the weight difference in the dot chart, as it is easier for the human perception to interpret the angle of a line. What needs to be considered

n the design of a chart is the ratio of width to height, as telling angles apart depends on how steep they are. Munzner [Mun14] mentions that telling 45° from 43° is way easier than telling 20° from 22°. Furthermore, the information from a line chart needs to be handled with care, especially as the linear connection between two data points may be interpreted as real values instead of interpolated ones. To refer to our example, we probably would assume that the cat's weight between 2004 and 2005 was about six lbs. This is an assumption that may not correspond to reality. It could also be possible that the cat discovered its temporary passion for Lasagna for a period in that year, which caused a huge weight impact that is not covered by the measurement points.

We finish our explanations on dot charts and line charts with their suitability in the context of this thesis. In medicine, it is usual to monitor certain values over a period of time, e.g., weight, height, pulse, or blood sugar, as this information can be used for adjusting treatment or clinical studies. Rajabiyazdi et al. [RPOC18] show the usage of line charts for monitoring blood sugar levels over time. Rincon et al. [RYS16] use line charts to visualize the development of an EMG signal. In preha, we will use line charts for all data interpretation tasks that show the development of a value over time, e.g. patient admissions, and discharges.

4.6.7 Data Tables

Visualization is the clear focus of the research performed within the scope of this diploma thesis. Up to now, we frequently highlighted the advantages of using visualizations for data representations. We referred on how their level of abstraction makes use of human perception to enable better insight into the data. At this point, we refactor this concept by adding a new item to our extended understanding of visualizations: the *data table*.

A table is a rectangular structure, separated in rows and columns, like a grid [Est14]. When visualizing data, columns usually contain the information of which attributes are shown, while rows determine the identity of unique elements. The element at a specific row and column is referred to as cell. Typically, the very first row of the tabular structure contains the so-called headers. A header is the first element of each column, it is a text that describes the content of a column. A row in the table body is referred to as entry. Figure 4.19 shows an example of a data table containing information about four cats. The header row is highlighted in grey, the table body has a white background. Columns display information on the sex, age, weight, and color of the cats. The first column contains the name of the cats. A table should always be provided with a proper title that communicates its main idea.

Data tables are well suited for visualization tasks that require raw data presentation. They are not used as visually appealing eye catcher or to make use of human perception [Est14]. An important aspect of tables is sorting of the entries by one of their attributes. This is of special interest when searching for specific values of attributes. Considering our example, it is hard to find the oldest cat or the cat with the most weight, as the data interpreter is required to manually search through all entries. By sorting the entries by

Main characteristics of my cats

	Sex	Age	Weight (lbs)	Color
Garfield	Male	6	7	Red
Tofu	Male	10	10	Brown
Poppy	Female	4	5	Grey
Gwenny	Female	3	6	Grey

Figure 4.19. Example of a data table, displaying cats and four associated attributes: sex, age, weight, and color.

Average characteristics of my cats grouped by color

Color	Sex	Ø Age	Ø Weight (lbs)	Count
Red	Male	6	7	1
Brown	Male	10	10	1
Grey	Female	3.5	5.5	2

Figure 4.20. Example for a grouped data table, displaying the cats from Figure 4.19, grouped by color.

the attribute age or weight, this task becomes rather simple. Another common feature in data tables is the ability to filter the table by values of attributes. This is useful if a subset of the data is explored. Referring to our example, a filter to the sex column with value "female" could be applied, resulting in a table with female cats only.

Furthermore, data tables support the representation of grouped data. In this case, entries are grouped together by common characteristics. If this is the case, all numeric values must be merged together, e.g., by using the average value. The resulting entries do not have an identity, even if there is just a single entry in a group. A count column provides information on how many original entries are in the group. Figure 4.20 shows an example of a grouped data table. It can be observed that there are two grey cats, with an average weight of 5.5 lbs and an average age of 3.5 years.

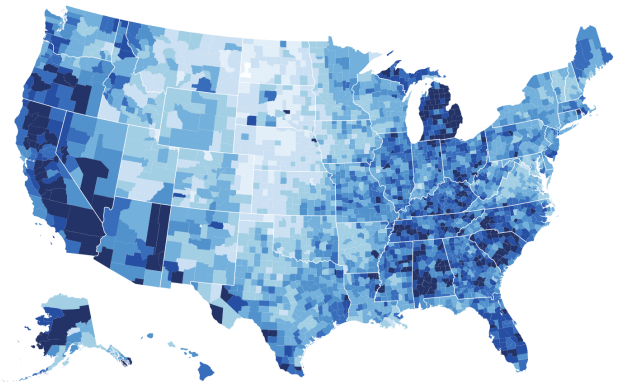

Figure 4.21. Example of a choropleth map from Munzner [Mun14]. This figure shows the US unemployment rate from 2008 on a segmented sequential colormap (blue to white). The color of a region corresponds to the unemployment rate of this region.

4.6.8 Geographic Visualization

The previously introduced visualization methods were mainly focused on providing information on quantitative aspects of the data. The task of displaying geographic information is rather difficult to fulfill with those methods. Still, it would be possible to display a bar chart with one bar per geographic region of interest, but this might not be as intuitive as using a map. A map can be used to encode quantitative information based on geographic positions. So called choropleth maps [Mun14] show regions as area marks based on their geometry, with a color that encodes a corresponding quantitative value. An example of a choropleth map can be seen in Figure 4.21.

In the example above, it is only possible to encode the values if they correspond to a specific region and not an exact geographic position. Though, cases may occur where the visualized location does not correspond to a specific region or this correspondence is not relevant. A quite popular example here is the visualization of the frequency of an event at a location, e.g., a map of lightning strikes after a thunderstorm. For this reason, geographic heatmaps are used. These maps also use a sequential colormap for displaying quantitative values, but they are not grouped by regions. An example for geographic heatmaps can be seen in Figure 4.22. Whether choropleth maps or heatmaps are used, it is critical to provide a legend showing information on the color mapping.

4.6.9 Metric Visualization

In certain cases it is required to present very simple elements, for instance the total number of patients in the cohort or the average age of the patients in the cohort. It would not add information if such an element is the only component of a categorical or distribution visualization. Important figures should be prominently placed in a dashboard

Figure 4.22. Geographic heatmap showing the popularity of locations in a Microsoft map service [Fis07]. Locations clicked more frequently by users are brighter.

to stand out immediately. For this reason, we use the so called metric visualization, which basically is just a large text displaying the corresponding element or number.

CHAPTER 5

Implementation

In Chapter 4 our focus was the design of preha, there we introduced the idea of our Visual Analytics application. We analyzed all parts of the architecture in separate sections and discussed possible solutions and design patterns. All of those sections were concluded with design choices. The aim of this chapter is to highlight the implementational details of our design choices. In the upcoming sections we will present technologies, programming languages, and patterns for the implementation of preha. In this chapter we will describe the implementation details module by module, with a structure analogous to Chapter 4.

5.1 Workflow

We will describe the overview of our implementation by tracing the path of the data, from the raw, unprocessed storage in the EHR up to its transformation to knowledge in the Visual Analytics dashboard. At first, the data is exported from the *EHR* as CSV files, at least one file per table. The *preprocessing* module, implemented in python is called via command line. We receive a single clean and well-structured CSV file at the output of the preprocessing module. It serves as the *primary data storage* of preha. Furthermore, this file is the data source for the kibana *Visual Analytics dashboard* [Gup15] and the *predictive analytics engine*, implemented in python. The data is transmitted to the Visual Analytics dashboard via the Representational State Transfer (REST) protocol. The predictive analytics engine loads the CSV file directly. Communication between the Visual Analytics dashboard and the predictive analytics engine is also implemented via REST interfaces.

5.2 Preprocessing Module

The preprocessing module is responsible for validating all data before it is used in preha. In order to determine the approach used for preha's data preprocessing, we will compare

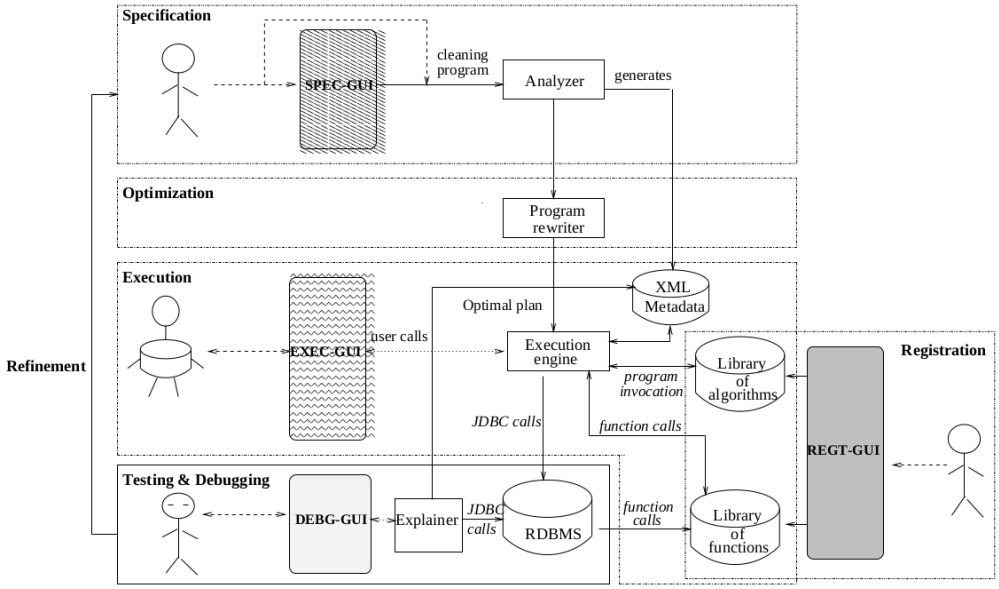

Figure 5.1. The architecture of AJAX as described by Galhardas et al. [GFSS00].

two frameworks for data preprocessing. The first approach we take a closer look at is AJAX [GFSS00]. Further we will investigate the potential usage of `pandas` [McK12].

5.2.1 AJAX

AJAX is a system that was designed to specify and execute data cleaning programs on one or multiple data sources. Four types of transformations are supported by AJAX. *Mapping* is used to standardize data formats if possible (e.g., date format) or reformat data by performing splitting and merging. *Matching* is used to compare data records that may refer to the same real object by a given matching criterion. *Clustering* can be used to group pairs of data with a high similarity value. *Merging* eliminates duplicates or produces new records for a dataset resulting from integrating data sources. Another major contribution of AJAX is an expressive and declarative language for specifying the so-called cleaning programs. This language is based on SQL statements. A big advantage of AJAX is the opportunity to enable the intervention of human experts in exceptional cases, where the given rules do not apply.

Figure 5.1 shows the architecture of AJAX. The start of the cleaning process is to write a cleaning program in the AJAX specification language. The subsequent optimizer rewrites the program for improved performance. The execution engine schedules the tasks of the elected cleaning plan. In the registration unit, external functions and programs can be included in AJAX. Furthermore AJAX features a Testing & Debugging module. Data cleansing is described as a sequential task with loops [GAM+14]. AJAX also follows this design principle by highlighting a refinement loop in the architecture.

5.2.2 pandas

The second approach we consider for data cleaning is a python library called pandas. Nowadays, python is one of the most important languages for machine learning or data science [McK12]. pandas is a library that provides high-level data structures and functions, used for working with tabular or other structured data. Among others, the main focuses of pandas are manipulation, preparation and cleaning of data. All transformations described by AJAX (mapping, clustering, merging and matching) are available in pandas. Probably the biggest advantage of pandas is that it is a python module. This fact enables all kinds of python operations to be performed. If pandas does not contain an operation, it is possible to use an external library or write a custom function.

pandas is lacking to be a full framework, though. While AJAX includes a structure for specifying cleaning programs, registering functions, or performing tasks [GFSS00], pandas is just a library. pandas covers the specification phase and python the execution phase in direct comparison to AJAX. In contrast, the simplicity of pandas' architecture makes it easier to use.

For this diploma thesis, we decide that using AJAX would require too much effort in terms of implementation. A simple and dynamic approach like pandas seems to be a better fit for our requirements. In order to overcome the issue of using just a primitive library, we aim to implement the cleaning by using pandas functions with elements from the AJAX world. The specification of cleaning programs, depending on the given dirty data, will give our cleaning approach the required structure. Furthermore, we provide a feedback loop that enables refinement of the respective cleaning program.

```
{
    "name": "DIAGNOSIS_CSV_IMPORT_PROFILE",
    "import_module": "diagnosis_importer",
    "clean_method": "diagnosis_importer.
       clean_diagnosis_dataframe(initial_dataframe.copy())",
    "features": [
        "facility",
        "case_id",
        "patient_id",
        "icd10_code",
        "type"
    ]
}
```

Listing 5.1. Example profile for diagnostic information.

Eventually, we developed a python application using pandas to deal with all types of tabular data that are exported from the EHR as CSV files. Among these types are diagnostic information, demographic information, therapy information, and score

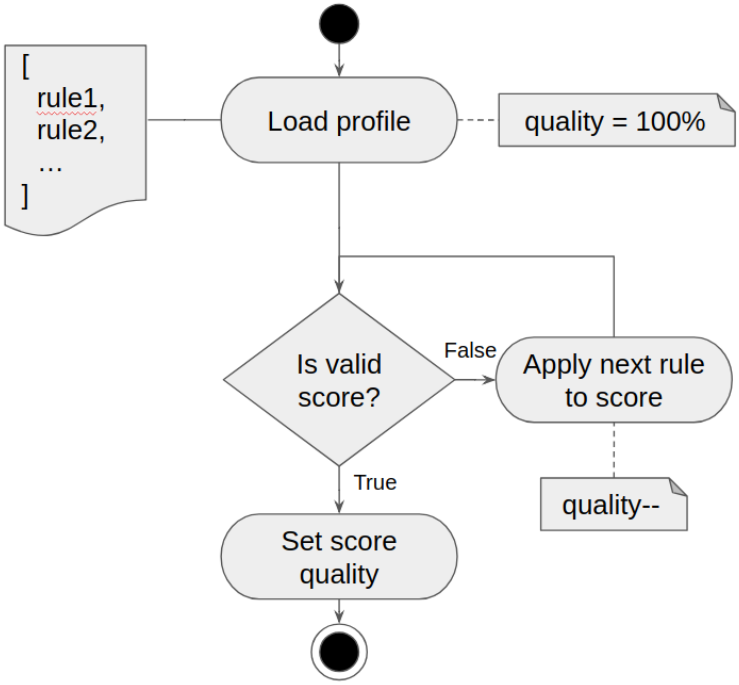

Figure 5.2. The logic of the score_importer module as UML flowchart.

information. When running the preprocessing module, the type must be specified, as each type of data has its own logic. In the processing module we refer to this logic as profile, each of which is covered by a separate python module. This profile is double-checked by validating the column names of the specified file. Preprocessing is only executed if the file is structured as expected by the given profile. Listing 5.1 shows an example of a predefined profile for diagnostic information.

Only if the features in the above Listing 5.1 match the columns of the given data, the specific module (diagnosis_importer, in this case) is executed. Each module has its own logic applied. The logic of the diagnosis and demographic module is rather simple. They involve just basic data wrangling tasks, such as renaming or deleting columns. The wrangling tasks in the preprocessing module are implemented using the `pandas` library [McK12]. The logic of the score_importer module is more advanced and will therefore be elaborated in more detail.

Figure 5.2 shows a unified modeling language (UML) flowchart of the score_importer process. When a new score (e.g., Barthel Index 66.0) is imported, an associated profile is loaded. This profile contains a set of rules for each score (e.g., replace "," with "."). These rules have been created in cooperation with VAMED engineers, as domain knowledge is highly important for such tasks [GAM+14, KCH+03, GGAM12]. A quality property is assigned to the score, which is intended to track the quality of the score measurements over time. After the profile is loaded, the score is validated. If it matches all criteria, it

Patient Id	Sex_male	Sex_female
1	1	0
2	0	1

Table 5.1. One hot encoding for the Sex column.

is returned including the quality. If not, the next rule is applied to the score and the quality is decremented until it matches all criteria. If all rules have been applied and the score is still not valid, it is considered as missing data.

5.3 Predictive Analytics Engine

The predictive analytics engine is the implementation of the machine learning part of preha in python. python is one of the most popular languages for scientific computing [PVG+11], especially in terms of machine learning and data science [McK12]. As we already use python for prehas preprocessing we see no reason in switching to a different framework. For the sake of completeness, we mention R, MATLAB, and SAS as other programming languages and tools suiting our needs.

We implemented our predictive analytics engine with two different python libraries. The first is `scikit-learn` [McK12], a very popular machine learning toolkit for python. We did not face any technical issues in performing our analytics with `scikit-learn`. A drawback of using `scikit-learn` is that it requires to encode categorical variables. This is because machine learning algorithms, such as random forest regression (as discussed in Section 4.5) are unable to deal with categorical variables, especially strings. The most popular method to solve this issue is "one-hot encoding". One-hot encoding takes a categorical variable and introduces a new column for every category. The category that corresponds to the categorical value is set to "1" in all rows that pertain. A simple example would be the Sex column of our dataset. In this case, only "male" and "female" are present. Table 5.1 displays the result of one-hot encoding this column. If a new category, e.g., "diverse", is introduced into the dataset, a new column "Sex_diverse" is added.

As our dataset is very high-dimensional and complex, dealing with categorical variables becomes rather difficult. In the course of development we ended up facing hundreds of columns, which results in increased complexity. This brought us to the second, very similar library called `h2o`. Very little changes were required in our source code in order to implement our predictive analytics engine in `h2o`. The result is a reduced effort for dealing with categorical variables. Both, `h2o` [NZL+16] and `scikit-learn` have been discussed in research as libraries for machine learning in the field of precision medicine. Williams et al. [WLR+18] focus on general applications of machine learning in precision medicine. Nezhad et al. [NZL+16] use `h2o` to determine risk factors for hypertension in the vulnerable demographic subgroup of African Americans.

It is the task of the predictive analytics engine to predict a rehabilitation score based on a set of filters (e.g., "What is the expected Barthel Index outcome for a patient between the age of 40 and 60."). This filters are the result of the visual query that has been applied to the Visual Analytics dashboard. As we defined a clean REST interface as input for this filters, it is also possible to use the predictive analytics engine without the Visual Analytics dashboard, or with a different application. This is a best practise approach from software engineering, which makes the code more reusable and versatile. The input parameters consisting of dataset filters and the prediction target can be seen in Listing 5.2.

```
filters[query][bool][must][1][range][age][gte]: 40
filters[query][bool][must][1][range][age][lt]: 60
filters[query][bool][should][0][match][rehabilitation_center.
   keyword]: Example Facility
predictionTarget: barthel_index_score_discharge
```

Listing 5.2. Incoming JSON request to the predictive analtiycs engine.

A translation of the above input would be as follows: Predict the value of the*Barthel Index at discharge*, for patients that

- are *40* years or older
- are younger than *60* years
- are in the rehabilitation center *Example Facility*

The preha predictive analytics engine now starts training a new random forest regression for the given dependent variable Y as prediction target. For memory optimization reasons, there is always just one random forest loaded in the current state of the application, corresponding to a prediction target. If a new prediction is requested, the application checks if there already exists a random forest for this prediction target. If this is not the case, a new random forest is trained.

For the training of this supervised machine learning algorithm, the existing dataset is split up into two parts. The first part is the *training* dataset, holding 75% of the values from the original dataset. As the name suggests, this dataset is used for training the algorithm. After the random forest is created, the second part of the dataset, the validation dataset is applied to the algorithm in order to determine its characteristics. The most important characteristic of the random forest is its accuracy [Alp10]. As we know the actual values of the dependent variable in the validation dataset, we calculate the accuracy based on the predicted and actual value. A measure that can be used for predicting the accuracy of forecasts is the symmetric mean absolute percentage error (sMAPE) [Flo86]. The sMAPE is an advancement of the standard MAPE, which is based

on dividing the predicted by the actual value, potentially causing division by zero. The sMAPE overcomes this issue and is defined as follows:

$$\text{sMAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|P_i - A_i|}{|A_i| + |P_i|}$$

$$\begin{aligned} P_i &= \text{Predicted value for observation i} \\ A_i &= \text{Actual value for observation i} \\ n &= \text{Number of observations} \end{aligned}$$

The accuracy is defined as 100% - sMAPE. Another important measure for the precision of our prediction is the mean absolute error (MAE), which is defined as follows [Flo86]:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |P_i - A_i|}{n}$$

$$\begin{aligned} P_i &= \text{Predicted value for observation i} \\ A_i &= \text{Actual value for observation i} \\ n &= \text{Number of observations} \end{aligned}$$

Another highly important characteristic is the variable importance, which is directly provided by `h2o`'s random forest regressor object. The variable importance is a percentage value that determines the influence of a variable on the prediction result. The mean absolute error, which is the mean value of the difference between predicted and actual values, is also returned for each prediction.

```
{
    "result": "87.5",
    "meanAbsoluteError": "3",
    "topTenImportances": [
        {
            "label": "barthel_index_score_admission",
            "importance": 84.7
        },
        ...
        {
            "label": "age",
            "importance": 2.7
        }
    ],
    "accuracy": "0.9"
}
```

Listing 5.3. Outgoing JSON response from the predictive analytics engine.

In order to predict the value for the prediction target in the request, the independent variable needs to be set up. First, we apply the filters from the request to the dataset. Every row from the remaining subset of the data is now applied to the random forest algorithm. The output of this is a vector of the predicted values for the prediction target. We return the average value of this vector as result of the prediction. The result for the Barthel Index at discharge is returned in the JSON structure visible in Listing 5.3.

In the response, it is visible that the result of our prediction is a Barthel Index of 87.5 with a mean absolute error of 3. The top ten variable importances of the prediction are represented as an array, with corresponding label and importance in % for each entry. Also the accuracy of the prediction, 90%, is included in the response.

5.4 Visual Analytics Dashboard

In Section 4.6 of Chapter 4 we introduced the visualization component of preha and highlighted the techniques this component needs to support. There are several Visual Analytics toolkits available that would possibly fulfill the needs of our Visual Analytics dashboard. However, in the course of our interviews we determined that there are two technologies VAMED's engineers are familiar with: R and kibana. We consider this a major requirement, as introducing a new technology would lead to less acceptance in the evaluation phase of our application. Therefore, we select R and kibana as potential frameworks for our Visual Analytics dashboard. Despite this, we want to mention that Harger and Crossno [HC12] provide a comparison of over 20 open source Visual Analytics toolkits. The comparison is the result of a two-stage evaluation. The first stage is a feature comparison of the toolkits, including visualization functions, analysis capabilities, and development environments. The second stage comprises a deeper analysis of a subset of the toolkits, including an implementation of a simple Visual Analytics application. The remaining part of this section deals with the comparison of two selected Visual Analytics toolkits, R and kibana.

R

R is a free and open programming language and environment set in the field of statistics. In science, R has become not only a common and well accepted language for statistics [Rah17], it is also popular for Visual Analytics applications [HC12]. The R core comprises a range of statistical methods: linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and also visualization techniques [R f]. The architecture of R is modular, the core can easily be extended by packages from the comprehensive R archive network (CRAN) including additional functionality. Over 10,000 packages can be found in CRAN, a lot of them are visualization components. The packages from CRAN enable R to load data from different sources, e.g., CSV or JSON.

Basically, visualizations in R are programmed in a script. While this fact makes it harder for inexperienced users to create visualizations, it enables better error tracing and code versioning. R includes well known visualizations such as bar charts, line charts or pie

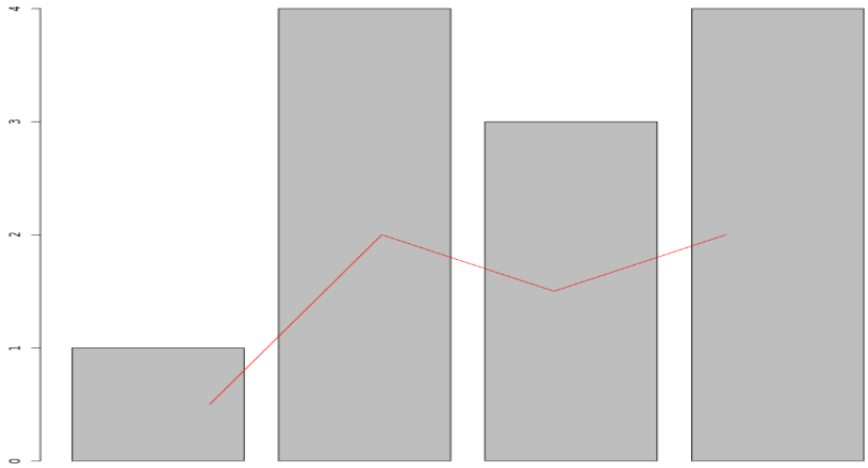

Figure 5.3. A basic visualization in R: Combination of a bar chart and a line chart.

charts out of the box. They can be created with a single-line command. Moreover, visualizations in R can be changed in terms of color, shape, and size by simple code modifications. Annotations, such as text, symbols, or arrows are also supported, just like the combination of different diagrams. We will now take a look at a rather simple example of R code. We create a bar chart showing the content of an array with four values (1,4,3,4). Further, we create a red polyline that represents each value divided by two. Subsequently, we plot the bar chart and add the red polyline. The result can be seen in Figure 5.3. The R code we used for creating the visualization is given in Listing 5.4.

```
par(bg="white")
bar<-c(1,4,3,4)
line<-bar/2
barplot(bar)
lines(line, col="red")
```

Listing 5.4. Code snippet for a bar chart with a red polyline in R.

However, designing a Visual Analytics dashboard for preha, basic static graphics are not sophisticated enough because of lack of dynamics and flexibility. For this purpose, R provides connectors that other visualization libraries (e.g., D3.js [BOH11]) can interface with, enabling interactive visualizations. As there are multiple concepts in Visual Analytics that are used frequently, there is a need for a framework systematically providing these concepts. In R this framework is called *Shiny* and is provided as a package of the CRAN. Shiny is intended to deliver interactive data summaries and queries to the user through a modern web browser [Bee13]. It comprises a variety of widgets that can be put together to build interactive user interfaces. Additionally, Shiny is free and can be extended by applying standard web technologies, such as HTML, CSS, JavaScript, and jQuery. An example for an interactive Shiny user interface can be seen in Figure 5.4. Conway et al. [CLG17] use it as a framework for a new visualization type

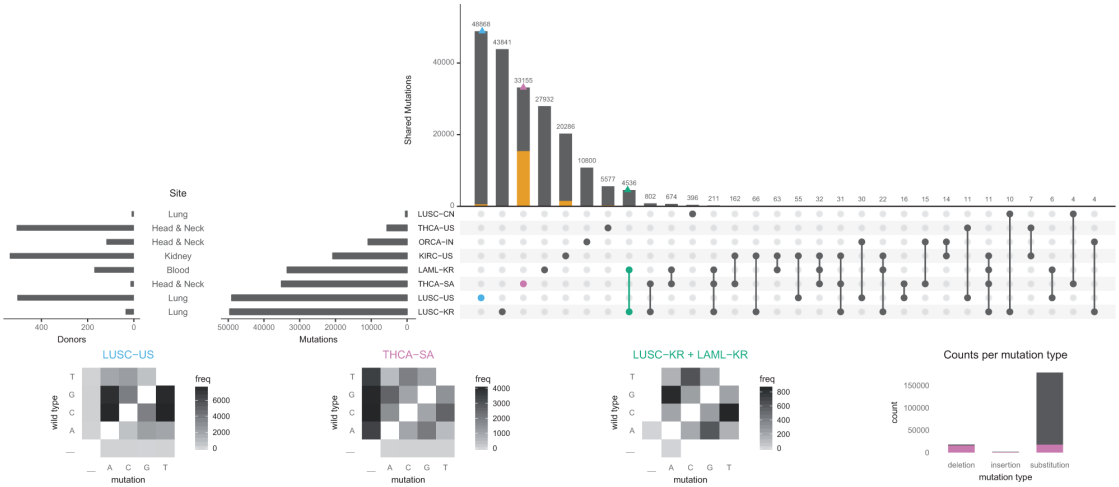

Figure 5.4. Shiny app used by Conway et al. [CLG17].

employed for plotting variants across eight cancer studies. Still, there are concepts Shiny is lacking. Chelaru et al. [CCB15] state that brushing and linking is not supported out of the box. Furthermore, custom visualizations can not be built in R directly, they require knowledge of a third party library, such as D3.js [BOH11].

kibana

Unlike R, kibana is not a programming language, neither an environment. kibana is a simple, yet powerful, UI for elasticsearch [Gup15]. As elasticsearch is the fundamental data source for kibana, we will now introduce basic information about it. Elasticsearch is a search engine built around Apache Lucene, a full-text search-engine library [GT15]. The main advantage is its speed when it comes to searching information stored in documents. Basically, a document is a JSON structure with keys and values. The example document in Listing 5.5 contains the same data as our example data in Table 4.6.

```
{
    "patient_id":       1,
    "case_id":          2,
    "age":              67,
    "sex":              "male",
    "primary_diagnose": "J44"
}
```

Listing 5.5. Example for indexed data in elasticsearch.

elasticsearch is a free and open source technology used for a near real-time analysis of large datasets. An example processing data from electronic health records can be seen in the work done by Chen et al. [CCB+17]. The American health care provider Mayo Clinic generates more than one million HL7 messages—a standardized format for the

transport of healthcare information—per day. These messages are stored in elasticsearch and can be searched among millions of other messages within milliseconds.

kibana provides a modern web browser user interface to elasticsearch [Gup15]. A query language can be used to define filters that are applied to the data. For instance, "sex:male AND age:<50" returns all documents with male patients under the age of 50. Furthermore, kibana comprises a variety of visualizations that can be used for gaining insights from the data, including but not limited to bar charts, line charts, data tables or simple metrics. There is also a range of third party visualizations that can be used in kibana. Contrary to R, visualizations in kibana are not created by writing code, but in the user interface of kibana itself. This is a major advantage, as standard visualizations such as bar charts can be created in a rather simple way. Additionally, this makes it easier for inexperienced users to create visualizations. Versioning of the visualizations is possible by creating a JSON object that contains all information of the visualization and can be imported and exported. Unlike in R, it is not possible to combine different visualizations.

For the purpose of this thesis, the biggest advantage of kibana is that it is built around interactive dashboards as core components. Visualizations can easily be added to a dashboard with simple drag-and-drop mechanisms. All visualizations in a dashboard are multiple coordinated views by default. Filters can be applied either by using queries or by selecting the corresponding items in the visualizations. The query "sex:male AND age:<50" would have the same effect as clicking on the "male" bar in a bar chart and selecting all values below 50 in a histogram of age. Additional visualizations that are not included in kibana can be provided in two ways. The first is programming a new visualization in HTML, CSS, JavaScript, e.g., D3.js [BOH11] and jQuery. The second is using the built-in vega [SWH14] module of kibana to create all visualizations supported by vega. vega is a declarative language for creating, saving, and sharing interactive visualization designs [SWH14]. It comprises all kinds of visualizations, also advanced ones, such as treemaps or sunburst diagrams.

As elasticsearch is designed to store large datasets, kibana is capable of providing visualizations for those. Kibana is built to visualize aggregated data, which can be an issue if users want to visualize individual data points. Additionally, kibana contains a built-in machine learning tool. However, there are two reasons this module does not fit the requirements of preha. The first reason is the lack of transparency of the machine learning process. It is not possible for the user to modify parameters of the algorithm. The second reason is that the machine learning tool is not licensed as open source product.

In order to make an objective decision on which technology to use, we first need to compare how the two introduced technologies fit our needs. Unfortunately kibana is not included in the Visual Analytics framework comparison by Harger et al. [HC12], because it did not exist in 2012. So therefore, we start by providing a brief overview on the main advantages and disadvantages of R (incl. Shiny, see Table 5.2) and kibana (see Table 5.3).

Both considered toolkits enable the fundamental techniques of Visual Analytics, as described in Section 3.1. Some of them seem to be easier to realize in kibana, especially

R - Summary

Advantage	Disadvantage
Popular environment for Visual Analytics	No dedicated visualization software
Packages for all kinds of data sources available (CSV, JSON, ...)	Building basic visualizations (bar chart, pie chart, table, ...) can become rather complicated
Visualizing single data points (e.g., scatter plots) is not an issue	Dashboards need to be implemented manually
The components required for preha are free and open source	Multiple coordinated views need to be implemented manually

Table 5.2. Summary of the main advantages and disadvantages of R.

kibana - Summary

Advantage	Disadvantage
Built for visualization of large and complex datasets	Machine learning module does not match the needs of precision rehabilitation
Building basic visualizations (bar chart, pie chart, table, ...) is simple	Data needs to be stored in elasticsearch
Dashboards are the main GUI components	Visualizing individual data points (e.g., scatter plots) is rather complicated compared to visualizing aggregated data
All visualizations in a dashboard are multiple coordinated views	Combining different visualization types is not possible (e.g., line in a bar chart)
The components required for preha are free and open source	

Table 5.3. Summary of the main advantages and disadvantages of kibana.

multiple coordinated views. To sum up, we conclude that kibana seems to be a better choice for the tasks introduced in this thesis. The rather simple tools for creating visualizations and dashboards in kibana have a bigger potential in establishing a direct link between the users and the data. Especially, tasks that are liberal regarding the dependency on specific visualizations, e.g., *Exp5*, benefit from this feature. A schematic overview of kibanas visual analytic dashboards can be seen in Figure 5.5.

The visual analytics dashboard is realized with the software kibana. Two of kibana's components are used to implement the desired dashboards. The first are the visualizations. kibana already supports a wide range of visualizations. There is no effort necessary in implementing basic visualizations, such as bar charts or line charts. Though, there is a visualization that is not supported by kibana from scratch, namely the treemap (as already introduced in Section 4.6.3). Furthermore, there is no existing visualization we can use for communicating with our external machine learning instance.

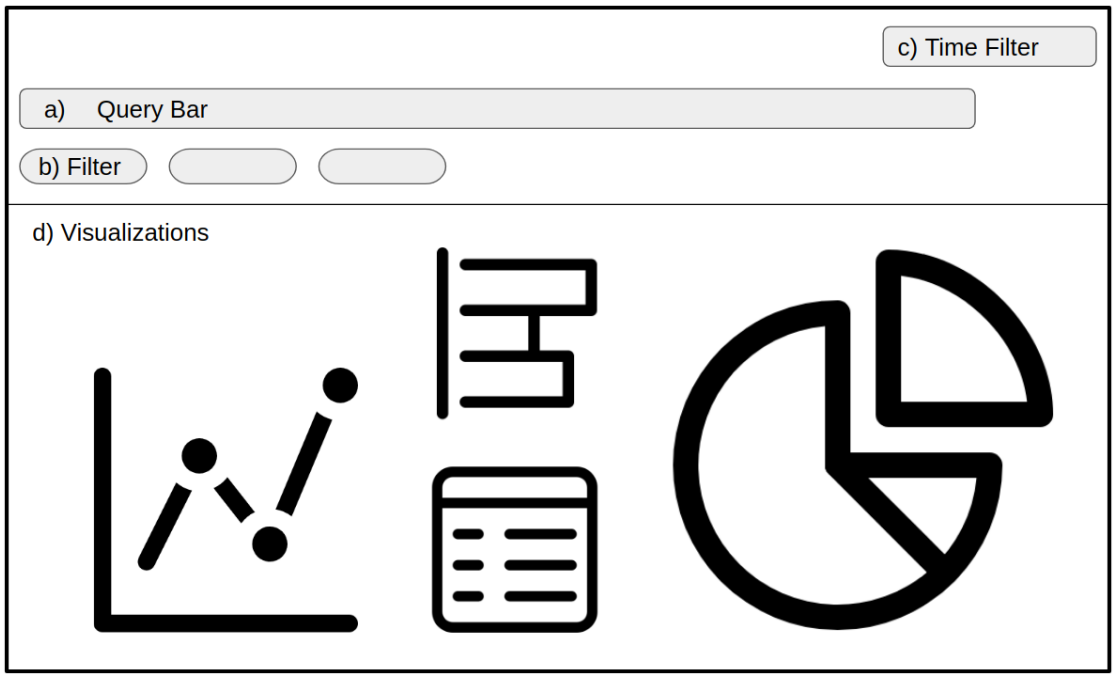

Figure 5.5. A schematic overview of kibana's dashboard user interface. *a) Query Bar*: Here data can be filtered by a criterion that is specified in a special query language. *b) Filter*: A visual filter element is created for every filter applied to the data, here existing filters can be modified or deleted. *c) Time Filter*: Here time filters can be applied to the data (e.g., show only data from the last year). *d) Visualizations*: The most space of the dashboard is occupied by all kinds of interactive visualizations. Each visualization reflects the data according to the applied filters.

5.4.1 General Interaction Implementation

As already described in Section 4.6, a range of interaction techniques must be supported by the Visual Analytics dashboard. In this section we will explain how kibana implements these interaction techniques. We start by an introduction of the kibana visualization panel. A schematic overview of the visualization panel can be seen in Figure 5.6. The *Query Bar* (a), *Filter* (b) and *Time Filter* (c) are exactly the same as in Figure 5.5 and therefore not explained in further detail. Additionally, the visualization panel contains a *Settings Bar* (d) and the *Visualization Preview* (e). The *settings bar* is where all visualization settings are applied and it is divided into three sections. In the settings in the *Data* tab the user can determine what is shown on the X and Y axis, how the data is sorted, how many items are shown, how the data is aggregated, or what labels are applied to the data. In the *Metric & Axis* tab all settings on the used metric and axis are specified, including how visualization elements, e.g., lines, are drawn or axis settings such as which scales are applied to the axis (linear or logarithmic). The settings in the

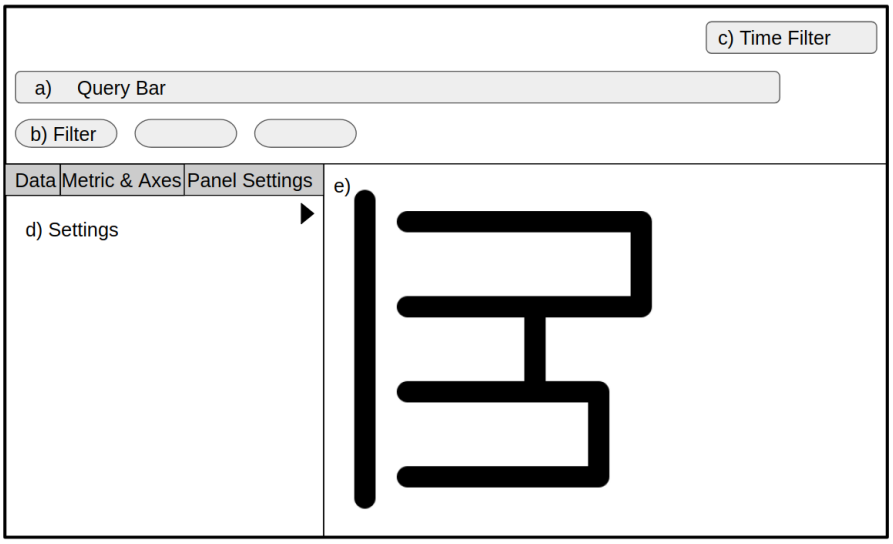

Figure 5.6. A schematic overview of the visualization panel in kibana. *a) Query Bar*: Here data can be filtered by a criterion that is specified in a special query language. *b) Filter*: A visual filter element is created for every filter applied to the data, here existing filters can be modified or deleted. *c) Time Filter*: Here time filters can be applied to the data (e.g., show only data from the last year). *d) Settings*: Here all possible configurations on Data, Metrics & Axes or Panel Settings can be set. *e) Preview*: A preview of the visualization with the current settings applied.

Panel tab are used for settings that are only related to the container of the visualization. They comprise the position of the legend, whether or not a toolkit is shown, and whether or not orientation lines are shown. The play button in the settings bar compiles the visualization with the current settings. If a visualization is added to a dashboard, only the part from in Figure 5.6 is visible.

Select interactions are triggered when a segment of the visualization is hovered by the cursor. Figure 5.7 exemplarily shows a pie chart that depicts different types of primary diagnoses. The cursor is hovered over the blue segment, which instantly highlights this segment in the visualization and the legend on the right. Also, a tooltip showing the name, value, and ratio of the segment, as well as the total number of the corresponding type in the dataset is displayed. Whenever changes are applied to the data, this tooltip is updated and enables users to track changes in the data concerning their item of interest.

Explore enables the user to shift the focus between subsets in a dataset. In kibana this is realized by enabling or disabling filters. Filters can be thought of as constraints applied to the data, e.g., show only values bigger than X or show only values of category Y. When performing exploration actions, it is necessary for the user to quickly toggle these filters in order to keep the focus on the analyzed data. We realize this by simply toggling the applied filter elements. A filter for sex male in enabled and disabled state can be

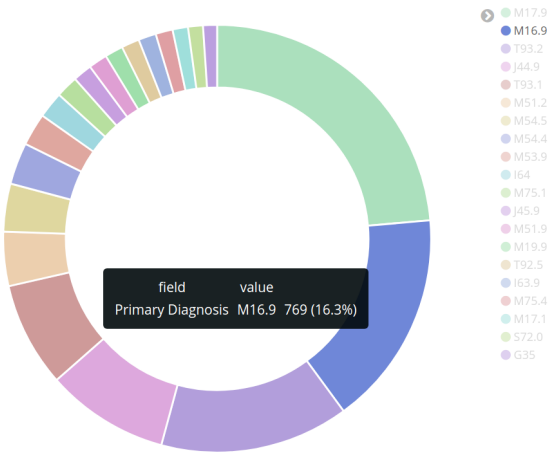

Figure 5.7. The select interaction in a kibana visualization.

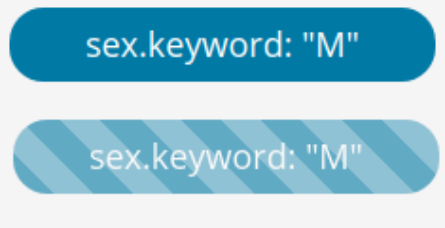

Figure 5.8. Filter for male in enabled state (top). Filter for male in disabled state (bottom).

seen in Figure 5.8. The filters are given in Figure 5.6b and Figure 5.5b as well.

Reconfigure is realized in our kibana dashboard by providing means to arrange data in visualizations in a specific order. This can be performed in the Data tab of the visualization panel. The visualization in Figure 5.9 displays a horizontal bar chart of the number of patients in the categories male and female. The top bar displays the female patients, the bottom bar displays the male patients. On the left side in the data tab the setting *Order* is set to *Descending*. Setting this to *Ascending* would change the order of the bars.

Encode actions can also be performed in the visualization panel of kibana. The color of the visualization can be changed in the top right corner, as it can be seen in Figure 5.10. The size of the visualization can be changed in the dashboard by adjusting the arrow in the bottom right corner with drag and drop.

Abstract/Elaborate is an interaction technique that is used to modify the levels of detail a visualization provides. In kibana this can be done in the visualization settings for map visualizations. The map visualization displayed in Figure 5.11 is an example of a visualization with few details, mainly displaying three circles where data points are located. Figure 5.12 shows the same data with more details. Numerous circles can be seen and the level of clustering is way lower, compared to the previous map. The

Figure 5.9. A bar chart ordered by the descending number of elements in a bar.

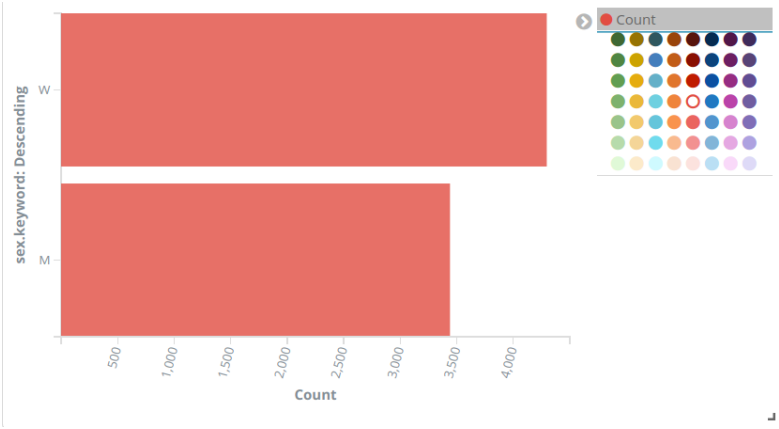

Figure 5.10. Encoding options in the visualization panel.

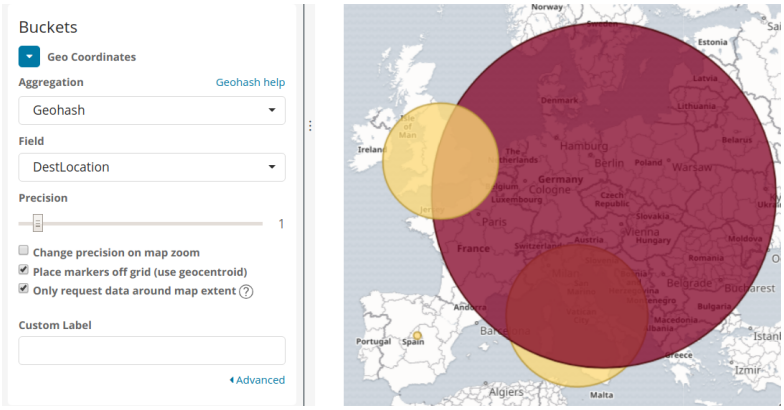

Figure 5.11. Map with precision set to 1.

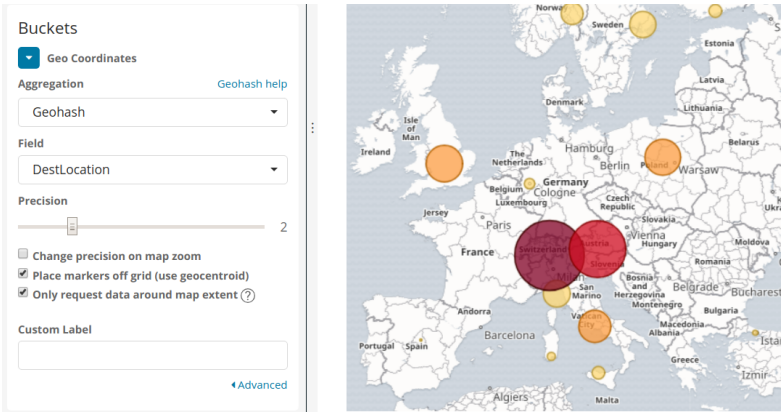

Figure 5.12. Map with precision set to 2.

corresponding precision setting in the visualization panel is visible at the left in both figures.

Some of the *Filter* possibilities of kibana have already been introduced as part of the *Explore* interaction. There is another important possibility to define filters, namely by selecting a segment in a visualization with the cursor. This will create a filter for the corresponding segment across all visualizations in the dashboard. Independent of how a filter has been added to the dashboard, it is always attached to the filter panel as shown in Figure 5.5b. Once a filter is added to the panel, it can be enabled, disabled, deleted, negated, or provided with a label. Furthermore, it is possible to change the operator of the filter, as can be seen in Figure 5.13.

5.4.2 Task Implementation

Section 5.4.1 focused on the general interaction methods and how they are used in kibana. In Section 4.6.2 we introduced how the tasks are designed and which visualizations are used. This section focuses on how the visualizations used for the tasks are implemented.

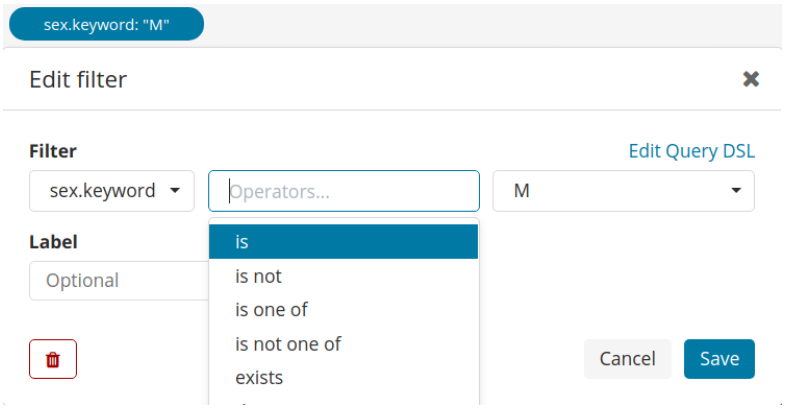

Figure 5.13. Edit filter menu of an element in the filter panel.

Firstly, we will concentrate on how visualizations are created in kibana. kibana provides a wide range of predefined visualizations, such as bar charts, pie charts, and line charts. Instead of explaining how every single one of those is created, we will describe how visualizations are created in kibana in general, based on the bar chart example, as displayed in Figure 5.9. When a new visualization is created in kibana, the first step is to set a so-called metric in the data tab of the visualization panel. Here the quantitative dimension of the data to be shown in the visualization can be set, for example average values, a simple count, a minimum, maximum, or a median. After this step is performed, the visualization would comprise a single segment, like a bar, a pie chart segment, or a dot, showing the selected quantitative value.

In order to split the data according to some characteristics, so-called buckets are used. Each bucket represents a subset of the data. This subset may, for example be a histogram of the data, count of data points by age groups of 20 years, a date histogram showing the distribution of a value over time, or the so-called "term", which splits a categorical feature with buckets for all characteristics. Moreover, the bucket options include various possibilities such as for sorting or setting a maximum number of buckets. It is also possible to create sub-buckets, which divide each bucket further. This would enable stacked bar charts by dividing each bar or sunburst diagram by adding an outer layer.

A very important functionality of all visualizations is the possibility to export the settings. This functionality produces a JSON object of the visualization, including all options, such as color and bucket settings. We will now continue our elaborations on the task implementation with the custom visualizations we developed.

Treemap Visualization

There are already existing approaches of implementing treemaps in the vega language, but we consider these as inappropriate, as none of them provide interactively zoomable treemaps as described by Blanch and Lecolinet [BL07]. We decided to develop a custom treemap visualization for kibana. In the development process, we opted to stick with the main design objectives of Johnson and Shneiderman [JS91]:

A1. *Efficient Space Utilization*: As large information structures are presented.

A2. *Interactivity*: The visualization must provide real-time feedback and support interactive control.

A3. *Comprehension*: It is desirable to minimize the loads on perception and cognition.

A4. *Esthetics*: The visualization is required to be visually appealing.

Furthermore, we considered the design guidelines that are the result of a series of experiments by Kong et al. [KHA10]:

B1. *Use treemap layouts that avoid extreme aspect ratios.*

B2. *Use luminance to encode secondary values in treemaps.*

A kibana visualization is based on javascript, Html, and CSS code. There are numerous examples of zoomable treemaps available. We decided to base our visualization on a zoomable treemap by Mike Bostock [Bos12], who made significant contributions to the development of the D3 javascript library [BOH11]. This is a suitable decision, as the treemap already covers several of the above mentioned design guidelines. Also, D3 is a library that is already used in kibana.

The treemap visualization always shows two levels of a tree structure. The sub-rectangles on the first level are highlighted with thick borders, we refer to this rectangles as parents. On the second level, the visualization shows sub-rectangles less emphasized for each parent. We refer to these as children. A1 is fulfilled by the principle of a zoomable treemap: The outer dimensions of the visualization stay the same, regardless of its content. A2 is also implemented from scratch. Though, slight modifications were necessary to match kibanas behavior. If an item is clicked in the treemap (see Figure 5.14), a new filter is applied to the data and a new treemap is generated with the selected item as top level (see Figure 5.15). A3 is accomplished through soft and minimal transactions, when a zooming action is performed. A4 requires our attention again, as Bostock's treemap does not use colors. In order to fit the visualization into kibana's design, we randomly assign a color from kibanas color palette to all parent elements. B1 is fulfilled by using a squarified layout—all rectangles are drawn as square as possible. In order to fulfill B2, we start with the colors assigned to the parents for A4. We set the same basic color of the parent to each child, but specify the brightness of the color depending on the area of the child. The larger the area of a child, the darker it is colored.

Machine Learning Visualization

The machine-learning visualization is the second visualization that we developed. Its purpose is to handle all communication with the predictive analytics engine. This comprises issuing requests with filters and prediction targets and handling responses with prediction characteristics. A prediction target is set in the machine-learning visualization

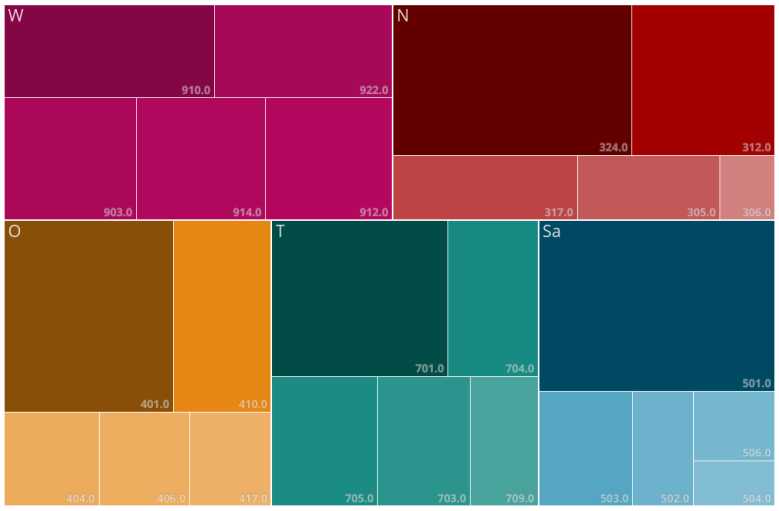

Figure 5.14. Top level of the treemap, showing the share of patients to Austrian regions. The second level shows the share of patients in the districts of a region. The label of a parent is always on the top left. The label of a child is always on the bottom right.

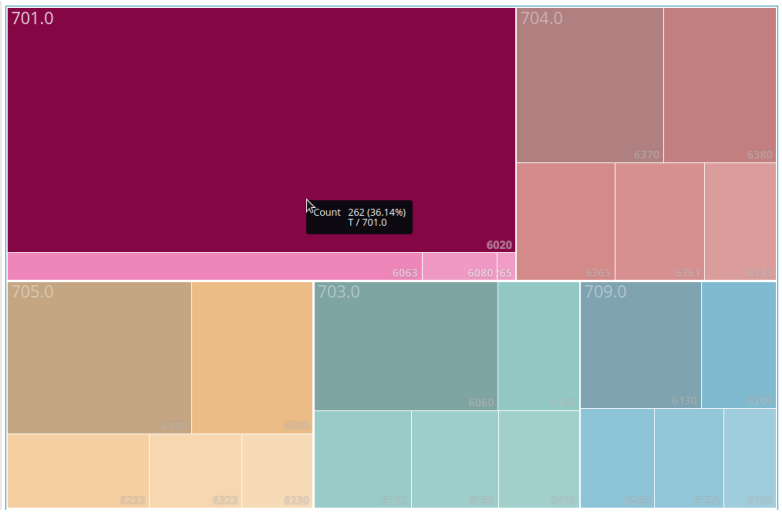

Figure 5.15. The treemap from Figure 5.14, after the region T (Tyrol) is selected. Now the children of T become parents. The highlighting can clearly be seen in this figure. If the cursor is moved on a parent, all other parents are occluded, which makes clear, which parent is currently focused. A tooltip shows additional information on the parent element, including the path through the tree so far and the share of this parent of the total data.

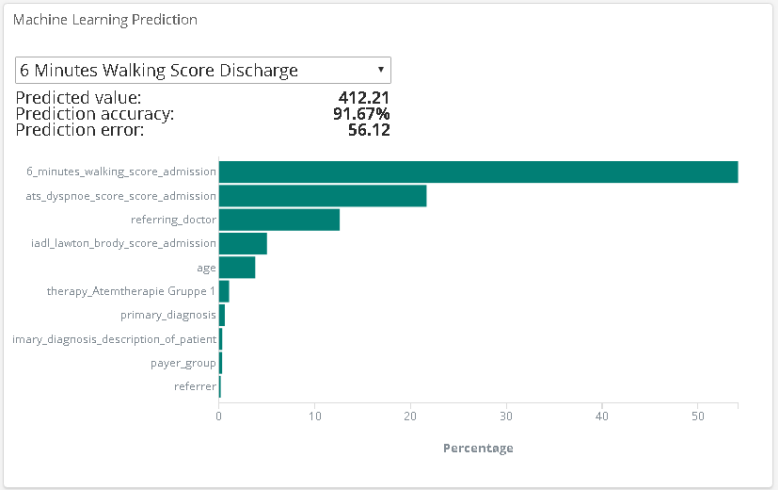

Figure 5.16. The Machine-Learning Visualization.

settings. Whenever a new filter is added to kibana, e.g., by selecting a part of a visualization, or removed from kibana, a new request is issued. This request extracts all filters currently applied to kibana, regardless if they were created by a visualization or manually. After the predictive analytics engine responds to the request, the machine-learning visualization handles the response and updates the data in the visualization. The machine learning visualization can be seen in Figure 5.16. At the top of the visualization, three textual lines are placed. The first line provides the predicted value, the second line gives the accuracy of the prediction, the third line shows the mean absolute error of the prediction. Below the text, we find a horizontal bar chart displaying the variable importances. The X-axis of the diagram shows the influence of the variables on the prediction in percent. The Y-axis lists the variables. A tooltip is displayed if the cursor is hovered over a bar, showing the variable name and its importance.

CHAPTER 6

Results

In this chapter we show the results achieved with respect to the completion of each task. For each task we provide usage scenarios, a short evaluation and meaningful representations. As a result of the definition of our tasks, the design of our application, and its implementation, we aim to analyze all described tasks by providing a usage scenario for every single one of them. At this point we want to highlight, that all of the data mentioned in this section, or visible on screenshots of the application, is based on concealed subsets of the dataset. This way, we ensure the confidentiality of VAMED's business information as well as the privacy of the patients. For the evaluation of each task, we will follow Munzner's nested model [Mun09]. We include the third stage of the model, that is focused on the encoding and interaction technique and design as well as the fourth stage that is about the algorithm design and performance in our evaluation. The third stage will be tested in the course of brief evaluation sessions with the users that were initially interviewed for this thesis. We document the overall impression of the users about the designed application, no metrics are recorded. In order to perform the evaluations related to the fourth stage of Munzners model, we perform measure the algorithmic performance of the machine-learning module. We conclude this chapter with a critical reflection of our results and outline potential improvements. More details are given in the sections below.

6.1 Usage Scenarios

Usage scenarios are common practice in the field of visualization evaluation [IIC+13]. The aim of a usage scenario is to demonstrate the potential applicability of a visualization from the perspective of a user, as opposed to case studies which are conducted by the experts. The visualization designer or developer takes on the role of the user. In the present case, we will now present the usage scenarios task by task, acting as an engineer or domain expert respectively. Having conducted an initial task analysis, presented in

Section 2.2.3, we are able to provide an objective and non-biased perspective on the application. A clear focus is upon the feasibility of the tasks using the proposed tool. We will introduce screenshots of the respective dashboards and show more details for the more complex visualizations. Other aspects besides feasibility, such as usability, user experience, and performance will be covered by the user evaluation section later in this chapter.

6.1.1 Usage Scenario of Eng1: Meaningful partitioning - Provide meaningful subsets of data

Research is carried out by rehabilitation facilities in order to improve the quality of care in the long term. For this reason retrospective studies are carried out based on population data, which the engineers provide for the domain experts. Generating meaningful partitions of the total dataset for those studies is the aim of this task. We present the dashboard that was created for this task in Figure 6.1. The aim of the task is to get the results of WOMAC ATL health assessment scores for female orthopedic patients from western Austria. The data table (Section 4.6.7) is the clear focus of this task, as it generates the output values. We can use the data table visualization as a preview on the subset. Each column can be used as sorting criterion for the data table. Usually a subset of data is comprised of a larger number of rows that exceeds the space provided for the visualization in the dashboard. For this reason, only the maximum number of entries is visible in the visualization. It is possible to highlight a cell in the data table and use it as filter (e.g., select "female" from the sex column, as it can be seen in Figure 6.1C. On the bottom left of the visualization, we can select the download button, that triggers the export of the data subset.

In order to visualize information on the distribution of the patients in the selected subset, a map visualization (Section 4.6.8) is used. It is clearly visible, that the majority of patients is situated in Vienna, which can be explained by the high population density. Zooming on this visualization shows additional details and gives the user a clearer sense on where exactly the patients are from. By drawing either rectangular or polygonal shapes in the map, we can filter the corresponding locations in the data. Assuming the research to be performed is focused on western Austria, we can now draw a rectangle on the map and filter these values. This rectangle can be seen in Figure 6.1B. The remaining filtered data is visible in Figure 6.2B, in the filter bar at the top, the applied latitude and longitude filter is added as filter button. At this point the location is zip code based, so the visualization does not reflect actual addresses of the patients with more detail than city level.

The treemap (Section 4.6.3) in the dashboard is a hierarchical visualization of the ICD10 diagnoses and the related groups. In Figure 6.1 we see the top level of the visualization, that corresponds to the chapters of the diagnoses. The hierarchical structure of the treemap is described in Table 3.1, along with the corresponding hierarchy of the ICD10 catalogue [Wor04]. Each level displays a maximum number of the tree structure's nodes that is set in the visualization panel. The treemap gives us a good overview

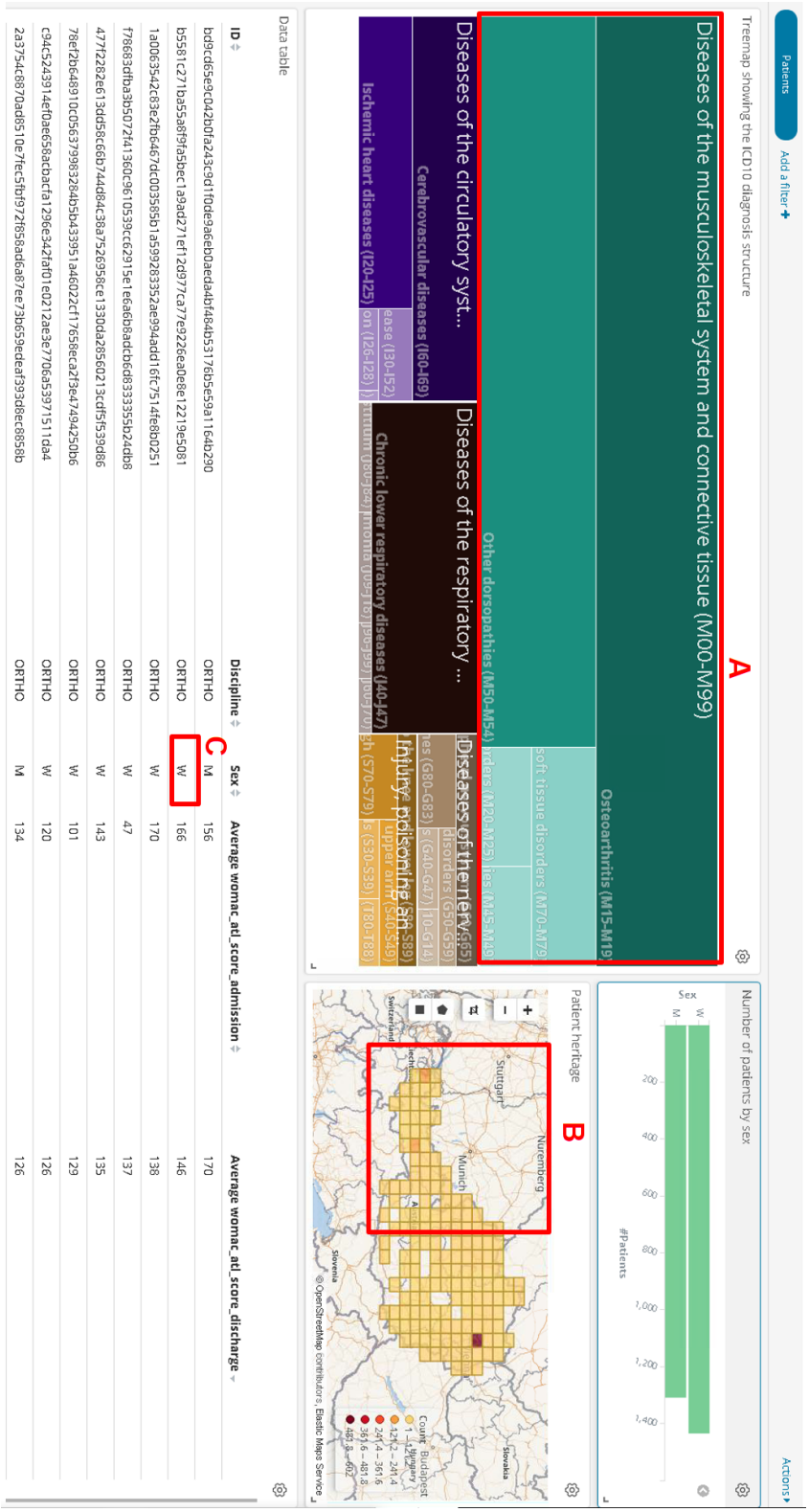

Figure 6.1. The dashboard of Task Eng1 before the filters are applied. A: The segment of the treemap we select to filter for ICD10 chapter. B: Shape of the area we filter in the geographic visualization. C: Value we select from the table visualization to filter for female patients.

of the proportions of the corresponding diagnoses in the dataset. As we only want to include orthopedic patients in our research, we select the ICD10 chapter *Diseases of the musculoskeletal system and connective tissue* from the treemap, as visible in Figure 6.1A. As the texts of ICD10 diagnoses are often quite long, a disadvantage of the treemap becomes visible. The available space for displaying the label of each parent is limited by the width that is determined by its proportion compared to the siblings. We handle this by cutting the text at the corresponding width, while making it available in full length through a tooltip.

6.1.2 Usage Scenario of Eng2: Assessment templates - Prepare templates for patient health assessment

It is inevitable to involve the patient when it comes to the discussion of the outcome measurements, in order to achieve the best possible rehabilitation outcome. Visualizing the rehabilitation outcome for the patients, helps them to understand their rehabilitation progress, according to statements of the domain experts in the interviews. It is the engineers task to provide the data in the desired format to the domain experts. Contrary to Task Eng1, it is not the engineers objective to provide data for analysis, but rather a dashboard template for presenting the data to the patient. The purpose of the dashboard is to depict simple presentations of health assessment outcomes that are presented to the patients. In the design of this dashboard, we decided to horizontally split it in half, as shown in Figure 6.3. The left side represents scores at the time of admission. The right side is analogous for discharge values. Also, we split the dashboard vertically in half. The average value of the current cohort is shown on the top, a distribution chart (Section 4.6.6) is given at the bottom. In order to solve this task, we prepare four visualizations for our dashboard.

1. We create a new metric visualization (Section 4.6.9). We set the aggregation of the metric to average and select the field containing the desired admission score.

2. In order to create the second metric visualization, we select the corresponding discharge score and save the visualization with a different name. This procedure is quite efficient, when it comes to creating a multitude of visualizations with similar settings.

3. As the distribution shall be presented as a line chart, we open the visualization panel of a line chart in order to create the third visualization. For the quantitative value on the Y-axis, we set the count of elements in the corresponding bucket on the X-axis. On the X-axis we define the corresponding admission score as the field for our bucket. Finally, we need to define the bucket size. The result is a line chart, with the points set to the extent of each corresponding bucket.

4. For the fourth and last visualization we again simply change the field name and save the chart under a different name.

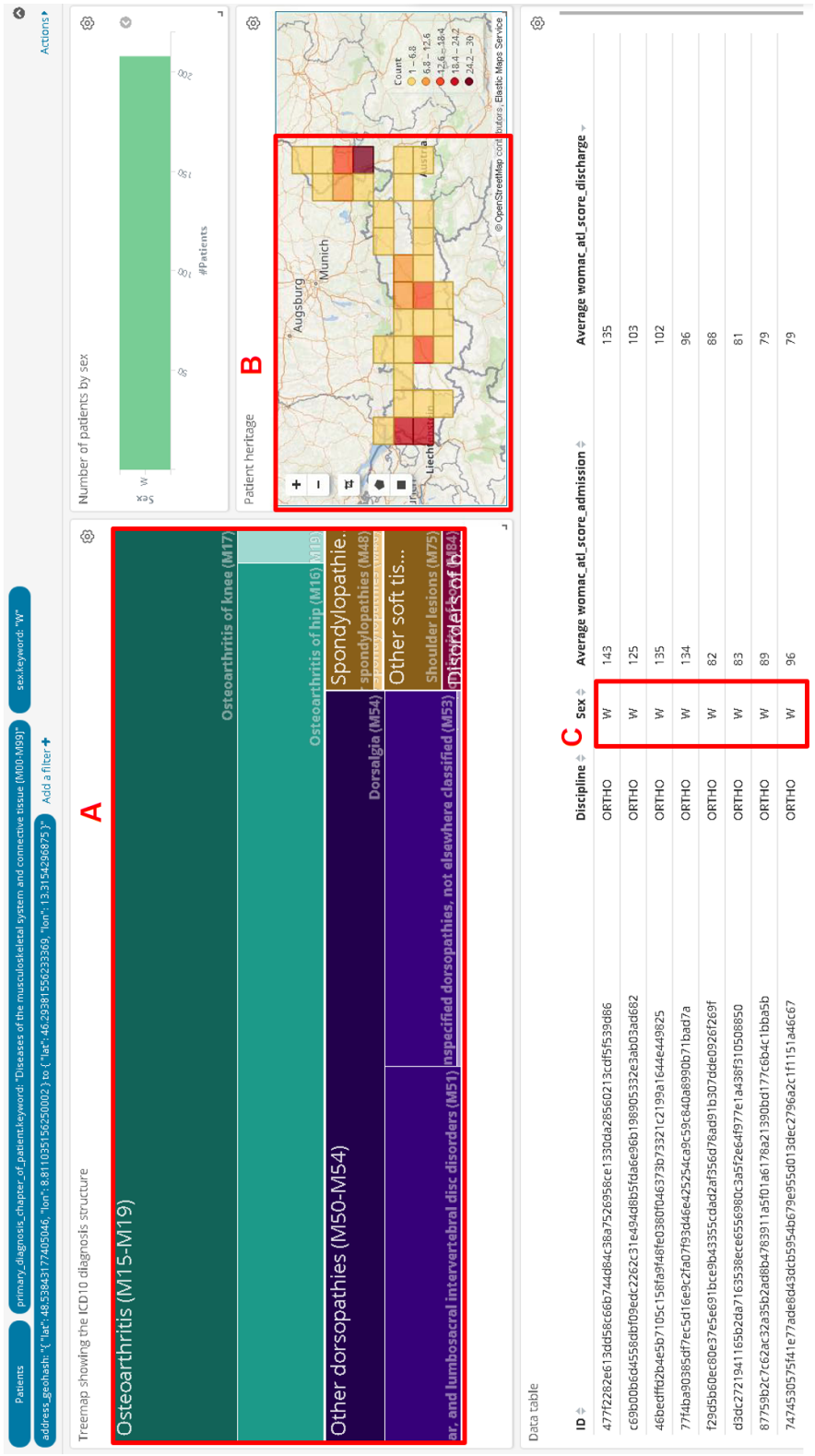

Figure 6.2. The dashboard of Task Eng1 after the filters have been applied. A: The former selected chapter of the ICD10 diagnoses is now the parent of the treemap. B: The selected area of the map now determines the geographic boundaries. C: Only female patients are present in the table now.

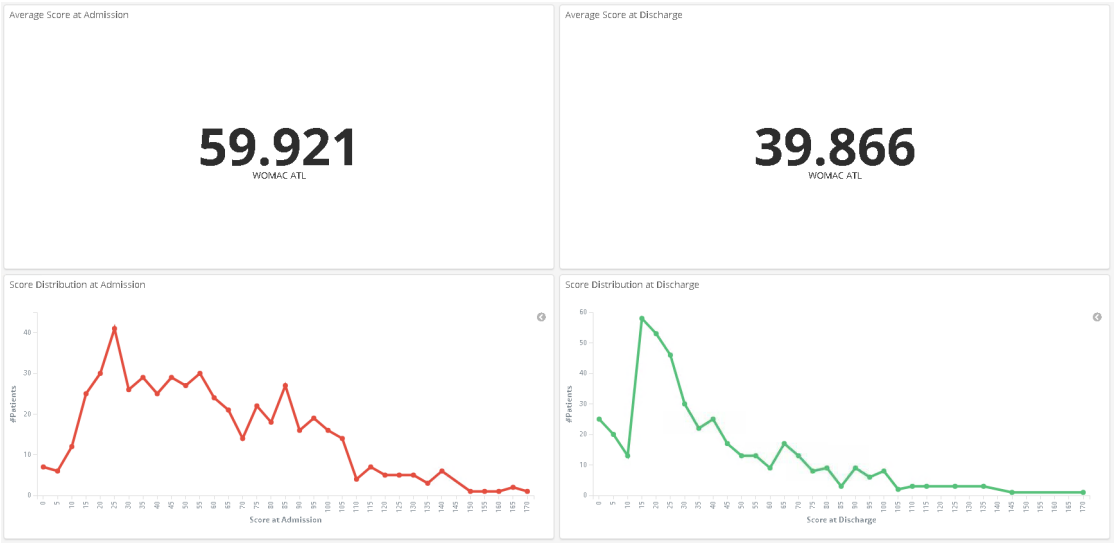

Figure 6.3. The dashboard of Task Eng2. Left: Visualizations for scores at admission. Right: Visualizations for scores at discharge. Top: Metric visualizations showing the average scores. Bottom: Distribution charts showing the histogram of the scores.

The dashboard created for Task Eng2 can be seen in Figure 6.3. Here we visualized the WOMAC ATL score. We will mention additional details on how the dashboard is used in the usage scenario for Task Exp1, where another screenshot is given.

6.1.3 Usage Scenario of Eng3: Benchmarking templates - Prepare templates for clinical benchmarking

The quality and efficiency [CCCD12] of healthcare services can be monitored and improved by means of clinical benchmarking. Payers, government regulators, and affiliated healthcare delivery organizations [RMM01] often demand this instrument to monitor healthcare facilities. The fundamental data source for this kind of analysis are rehabilitation outcome measures as described in Section 2.2.2. The task of clinical benchmarking is described quite frequently in the interviews by engineers as well as domain experts.

From the engineers point of view, four visualizations have to be prepared, they are shown in Figure 6.4.

1. In order to display the number of patients in the current selection, we add a new metric visualization (Section 4.6.9). As metric, we set the count aggregation. This visualization can be seen in Figure 6.4A.

2. For the distribution of the patients over time per facility, we add a new line chart (Section 4.6.6) in kibana. As we want to count the number of admissions, we select the count aggregation as metric. In order to define the location of the dots on the X-axis, we set a per-week date histogram as bucket. Subsequently, we split the

line chart by the corresponding facility. This is done by adding a sub-bucket to the date histogram, with one sub-bucket per facility. This visualization can be seen in Figure 6.4D. Facilities A and B are represented by the yellow and blue line respectively.

3. Quite similar, we add a line chart (Section 4.6.6) that shows how a health assessment score develops over time. We add a new line chart and set the average value of this score as metric aggregation. As bucket, we again select the date histogram to show progress over time. This visualization is given in Figure 6.4C.

4. Finally, in order to create a categorical visualization (Section 4.6.4) of the rehabilitation's top five payers, we add a new horizontal bar chart. As metric we select the standard count aggregation. In order to get a single bar per payer, we split the bucket by "Term", where term refers to the unique features of the category. This visualization can be seen next to the metric visualization in Figure 6.4B. Each category corresponds to a payer.

We will mention additional details on how the dashboard is used and what information can be derived in the usage scenario of Task Exp2, where a screenshot is also given. Task Eng3 is limited to creating the dashboard template only.

6.1.4 Usage Scenario of Eng4: Outcome predictions - Use machine learning to predict rehabilitation outcome

Using machine learning to predict the rehabilitation outcomes is a desirable feature for the domain experts. As this is a highly technological task, tight interdisciplinary collaboration is mandatory. From an engineer's point of view, we need to thoroughly understand how the machine-learning visualization (Section 5.4.2) works in order to support the domain experts with their predictions.

It is possible to use the machine-learning visualization in any dashboard we created in kibana. In the design of the dashboard for the machine-learning visualization, we take care which visualizations to add. The purpose of all other visualizations in the dashboard is to make constraints for the prediction of the machine-learning visualization. For example if we want to predict a score that is assessed at the admission and discharge of the rehabilitation, we add a visualization that allows us to set boundaries on the admission scores. We implement this by adding a bar chart that displays the distribution of the admission score as it can be seen at the bottom right of Figure 6.6.

Another example would be a categorical visualization, where the patient age group can be selected. The joint constraints set by these visualizations determine the input parameters of the prediction. We now have the ability to explore the predicted outcome, based on a variety of feature combinations. We notice how the predicted outcome changes for each constraint applied. As an example we focus on the prediction of the WOMAC ATL score at discharge. The WOMAC score is used to evaluate the condition of patients

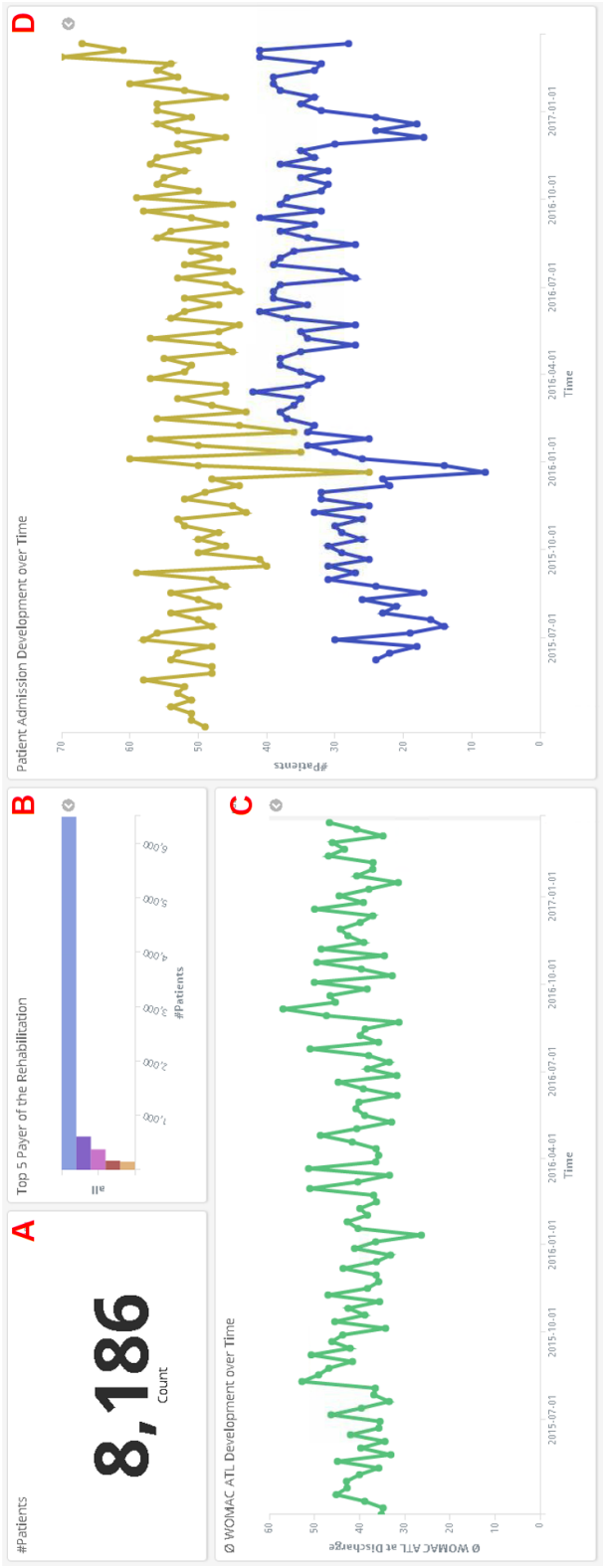

Figure 6.4. The dashboard of Task Eng3. A: Metric visualization showing number of patients in current selection. B: Categorical visualization showing distribution of payers. C: Line chart showing the development of admissions over time. D: Line chart showing development of average WOMAC ATL over time.

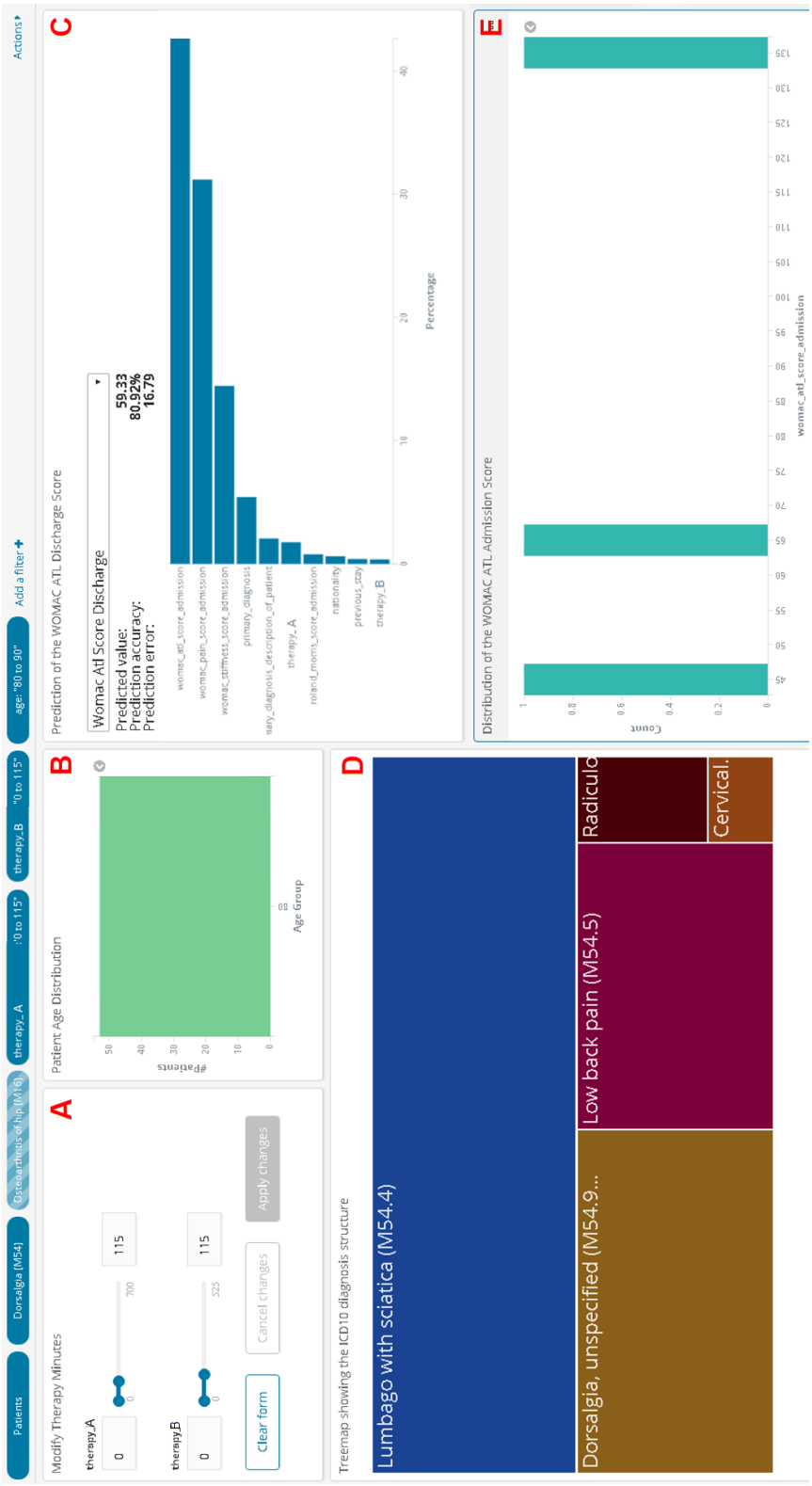

Figure 6.5. The dashboard of Task Eng4 showing the machine-learning visualization with the prediction of a high WOMAC ATL score. A: Range sliders for selecting the number of minutes per therapy. B: Categorical visualization showing the patients' age distribution. C: Machine-learning visualization. D: treemap showing the diagnosis group for Dorsalgia. E: Categorical visualization showing the distribution of the WOMAC ATL admission score.

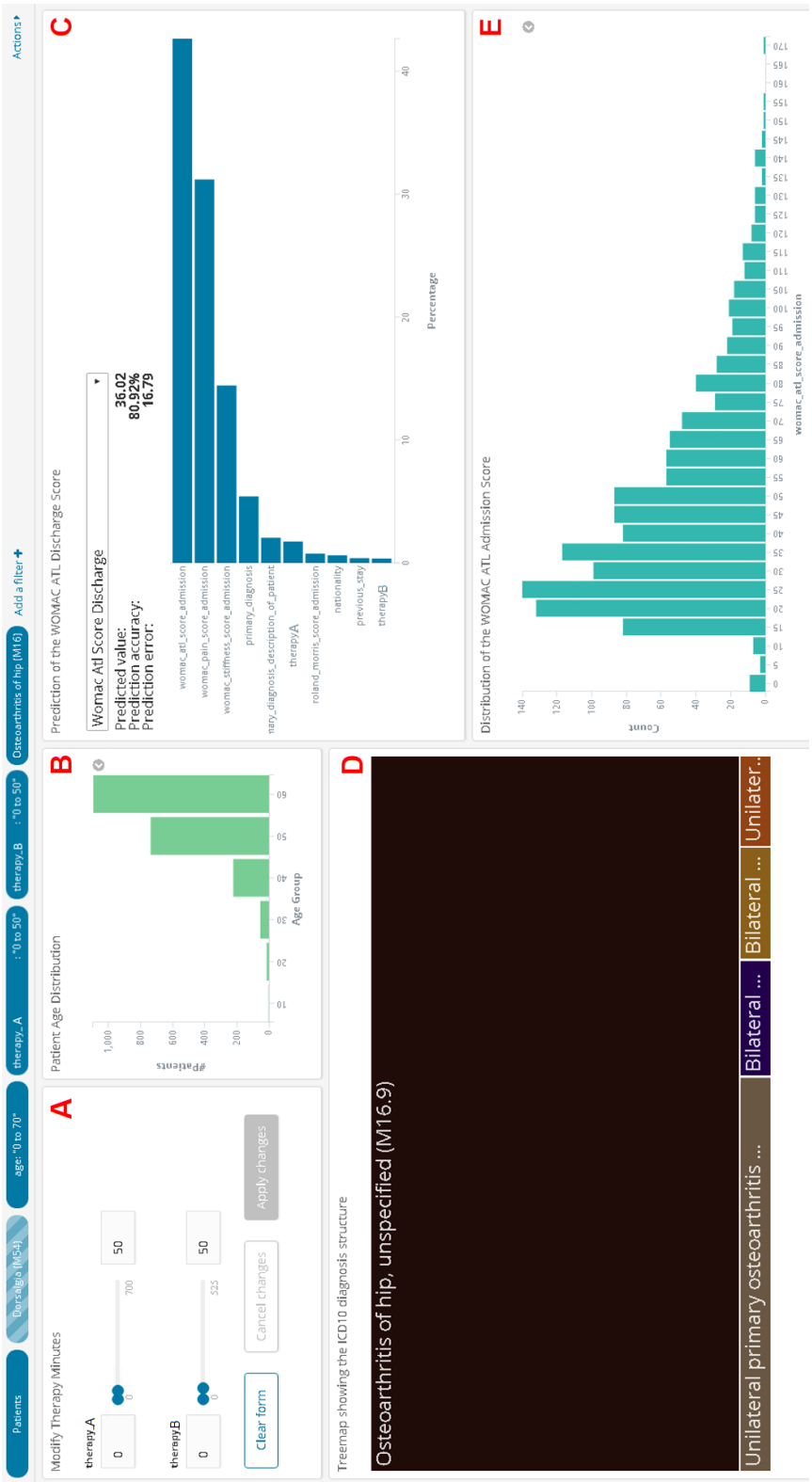

Figure 6.6. The dashboard of Task Eng4 showing the machine-learning visualization with the prediction of a low WOMAC ATL score. A: Range sliders for selecting the number of minutes per therapy. B: Categorical visualization showing the patients' age distribution. C: Machine-learning visualization. D: treemap showing the diagnosis group for Osteoarthritis of the Hip. E: Categorical visualization showing the distribution of the WOMAC ATL admission score.

ICD10 Diagnosis Group	Age	Therapy A [min]	Therapy B [min]	Predicted WOMAC
Dorsalgia (M54)	80-90	0 - 115	0-115	59.33
Dorsalgia (M54)	80-90	0 - 115	-	54.71
Dorsalgia (M54)	80-90	-	-	55.95
Dorsalgia (M54)	-	-	-	53.6
-	-	-	-	41.38
Osteoarthritis of Hip (M16)	-	-	-	38.51
Osteoarthritis of Hip (M16)	0-70	-	-	37.97
Osteoarthritis of Hip (M16)	0-70	0-50	-	36.52
Osteoarthritis of Hip (M16)	0-70	0-50	0-50	36.02

Table 6.1. Prediction results of the WOMAC score for different combinations of filters applied.

with osteoarthritis of the knee and hip. A low score means that the patient is in a good condition, so we want it to be as low as possible at the time of outcome. Table 6.1 shows the results of various predictions with different filters each. A detailed view of the machine-learning visualization can be seen in Figure 6.6. The machine-learning visualization includes a bar chart of the variable importances. We now add constraints on the variable importances and monitor their effect on the predicted score. In order to select an ICD10 diagnosis, we use a treemap. In the prediction performed, we compare the effects of Dorsalgia (M54) and Osteoarthritis of Hip (M16). We use range sliders to adjust the number of therapy minutes a patient has in the course of the rehabilitation, we notice that setting the range to 0-50 minutes has a significant impact on the predicted outcome score, as can be seen in Figure 6.5. Another variable that has an impact on the prediction is the patients age, which may be set via a bar chart. By varying these variables, we face results ranging from 59.33 to 36.02. A detailed overview on the prediction results of our filtering interaction can be seen in Table 6.1. The prediction features an accuracy of 80.92% and a mean absolute error of 16.79. We refer the reader to Section 6.2.6 for further details on the machine-learning performance.

At this point, we shift our focus to the domain experts' usage scenarios. Contrary to the usage scenarios of the engineers, which were rather focused on building visualizations, it is the domain experts' task to use our application in clinical routine to support decisions and present data.

6.1.5 Usage Scenario of Exp1: Outcome presentation - Show rehabilitation outcome to patients

The domain experts present the rehabilitation outcome to the patients. This task is based on the dashboard templates created by the engineers in Task Eng2. This task is designed to support a standardized procedure, in opposite to more exploration-focused tasks such as Exp3. Presenting the rehabilitation outcome is a frequent task that is embedded in the clinical routine. Therefore the possibilities for interaction are quite minimal.

Figure 6.7 shows the dashboard for Task Exp1. Again we decide to show the WOMAC

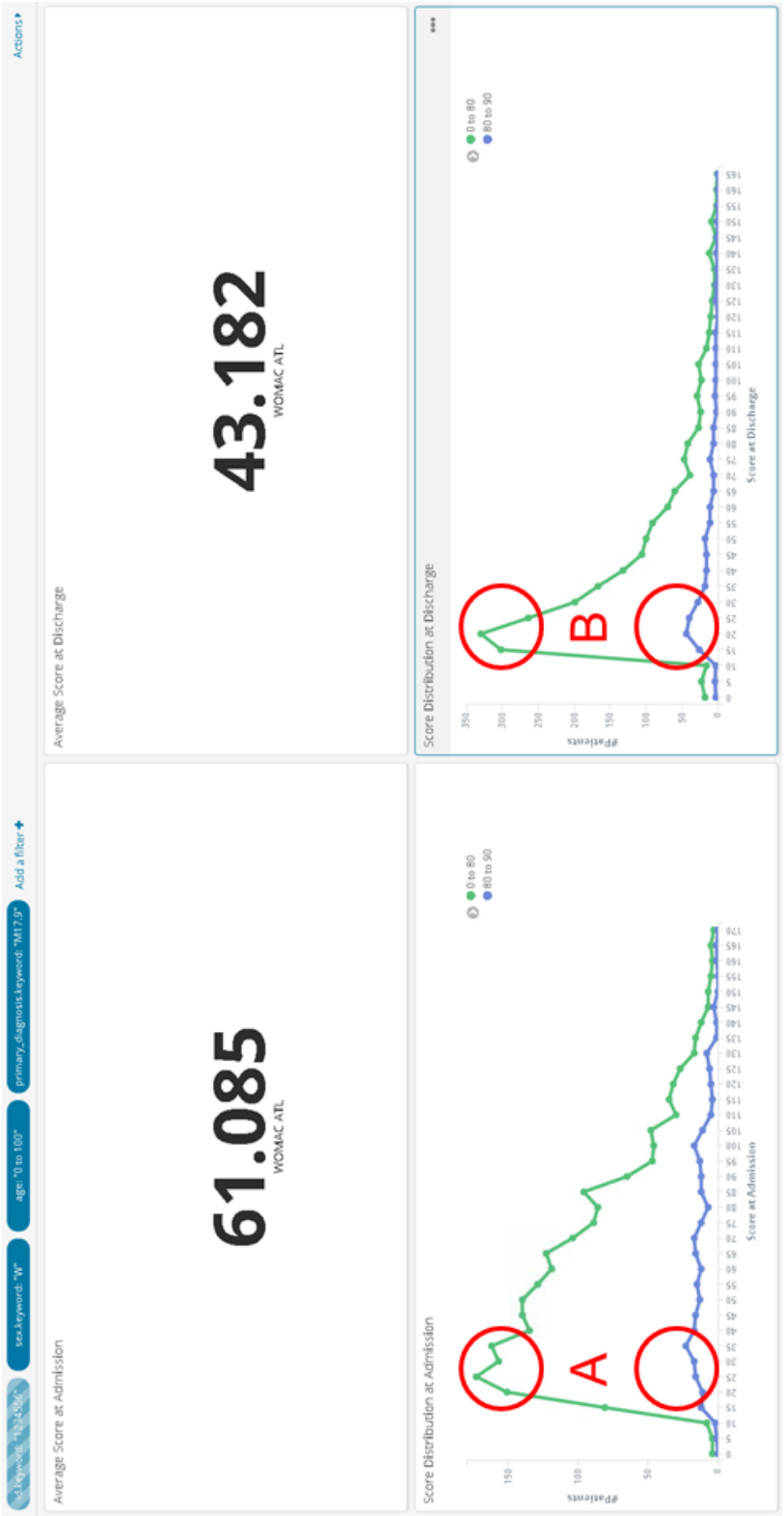

Figure 6.7. The dashboard of Task Exp1. A: The histogram distribution accumulation of the WOMAC ATL admission score. B: The histogram distribution accumulation of the WOMAC ATL discharge score.

ATL score. At first we present the individual assessment outcome to the patient in a metric (Section 4.6.9). Therefore we prepare a filter on the patients ID, which can be seen in the filter bar at the top. Enabling and disabling this filter allows us to quickly switch between the average score of all patients and the average score of the presented patient. Though, comparing a patient to the total average of all patients is a bit short-sighted. Therefore, we prepare filters that enable us to compare the patient to a more suitable sub-cohort. Options for these filters include age, sex, and primary diagnosis. We filter for female patients between the age of 0 and 100 with primary diagnosis M17.9. The same settings can be used for the comparison of the patients' results in the score histogram (Section 4.6.6). In this chart the distribution of the score results becomes visible. This allows the domain experts to explain to the patients how their score performance is aligned among the others. An issue that needs to be considered here is the time range of the comparison. It has to be determined, if a patient that was admitted in 2018 is comparable to patients that were admitted for example 10 year ago. New treatment strategies may have been developed that shift the rehabilitation outcome and scores are therefore not comparable. It is up to the domain expert to decide how to take care of this. In the dashboard we can address this issue by setting the time range of the sub-cohort as desired. It is visible in Figure 6.7A that the histogram of WOMAC scores at the time of admission is more equally distributed than at the time of discharge (Figure 6.7B). An interesting observation we make here is the comparison of age groups. While all age groups of 10 year ranges behave the same, the 80 to 90 year olds show a different distribution in the admission chart. For a better visibility we put all remaining patients in a single group, leaving us with two lines in the bar chart. The 80 to 90 year old patients do not show an accumulation on the lower end of the WOMAC ATL scale at the time of admission. At the time of discharge the accumulation is present.

6.1.6 Usage Scenario of Exp2: Clinical benchmarking - Perform clinical benchmarking

Similar to Task Exp1, Exp2 is also based on a dashboard prepared by the engineers in Eng3. The dashboard for Exp2 can be seen in Figure 6.8. From the domain experts' point of view, we apply certain filters to the data and monitor the corresponding results. A focus of this dashboard is on two line-chart visualizations (Section 4.6.6). The first displays the development of admissions over time. In this visualization we can observe information on recurring patterns.

An example for this is the so-called Christmas kink, a significant decrease of admissions around Christmas time. This can easily be explained, as a large part of patients want to return home to their families for the holidays. Therefore an interruption of the rehabilitation is requested. In Figure 6.8 the Christmas kink is highlighted by red circles. In the line chart showing the admissions over time we separated the development of 80 to 90 year olds and all other patients. An interesting observation is that the number of patients in the older age group does not increase as significantly as it is the case for the other age groups. Furthermore we observe, that at the mid of 2015 and end of 2016 there

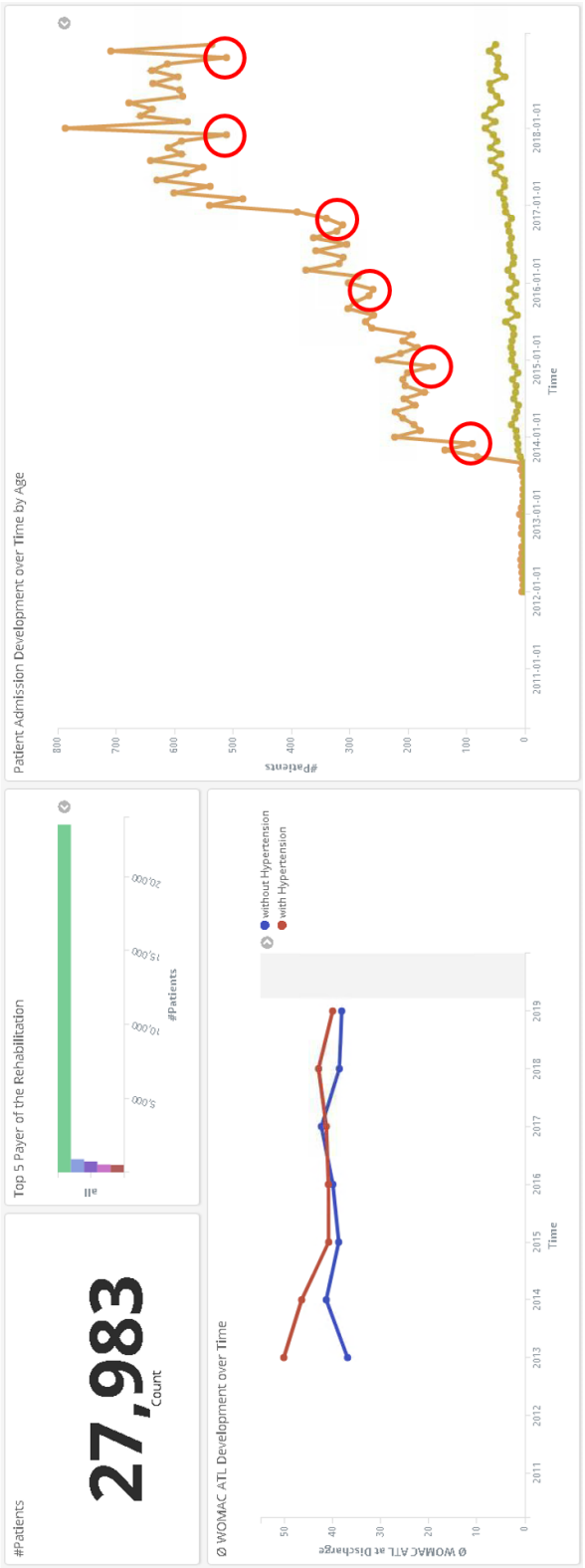

Figure 6.8. The dashboard of Task Exp2.

is a significant increase of admissions which is also not reflected by the group of 80 to 90 years olds. This increase may be caused by a new rehabilitation facility using the EHR or an existing rehabilitation facility expanding its services.

In the other line chart, the development of a rehabilitation performance indicator can be seen. This indicator, a discharge score, has to be set for each discipline individually, as there is no general measure that can be applied to all patients. In combination with the categorical chart (Section 4.6.4) that displays the rehabilitation payer, the domain experts can observe potential differences in the rehabilitation performance. Furthermore, deviations in the admission development can be observed for a payer. This can be done by interactively selecting the bar of the payer of interest from the bar chart. No major differences could be perceived among individual payers—which was expected—so we do not filter it in the presented screenshot. An interesting observation can be made when comparing the WOMAC ATL score among patients with and without hypertension. In Figure 6.8, the line chart at the bottom left displays this comparison. It is clearly visible that patients without hypertension have lower scores. Though, this observation may not be caused by the hypertension itself, but by other factors among this subcohort.

6.1.7 Usage Scenario of Exp3: Clinical exploration - Explore clinical dataset

The exploration of the data for the reason of pure enjoyment and curiosity is a task that was mentioned by all domain experts in the interviews. New ideas for scientific research may possibly be generated as a result of this very unique way of looking into the data. A hands-on usage scenario can not be provided for this task, as there is no clear target set for it. We will present some possible visualizations that the domain experts may use for their analysis. Figure 6.9 shows the dashboard for Task Exp3 for patients from western Austria with hypertension. Figure 6.10 shows the same dashboard with a filter applied to patients with diseases of the circulatory system (I00-I99).

As we recognized an interesting development of patients with hypertension in Task Exp2 we want to further investigate this subcohort. To explore the dataset in all its facets, we have to apply different kinds of visualizations, each of them highlighting a different aspect. The domain experts stated in the interviews that is is desirable to integrate the local heritage of the patient in the analytics process. In order to achieve this, we introduce a geographic map (Section 4.6.8) to the dashboard, as already used in Eng1. This map can be seen in the dashboard in Figure 6.9D. A clear accumulation of distribution can be seen at the eastern and western ends of the map. Another frequent request is to explore the data in relation to the patients' age or age group. We use a detailed categorical visualization (Section 4.6.5) of sex by age in order to enable this feature. An age our age group filter can be set from this chart by selecting the corresponding age range on the X-axis. As the distribution of the patients sex is also of interest in this context, we split each bar by sex. The blue bars correspond to female patients, the green bars correspond to male patients. A distinct difference in terms of sex distribution can be observed when

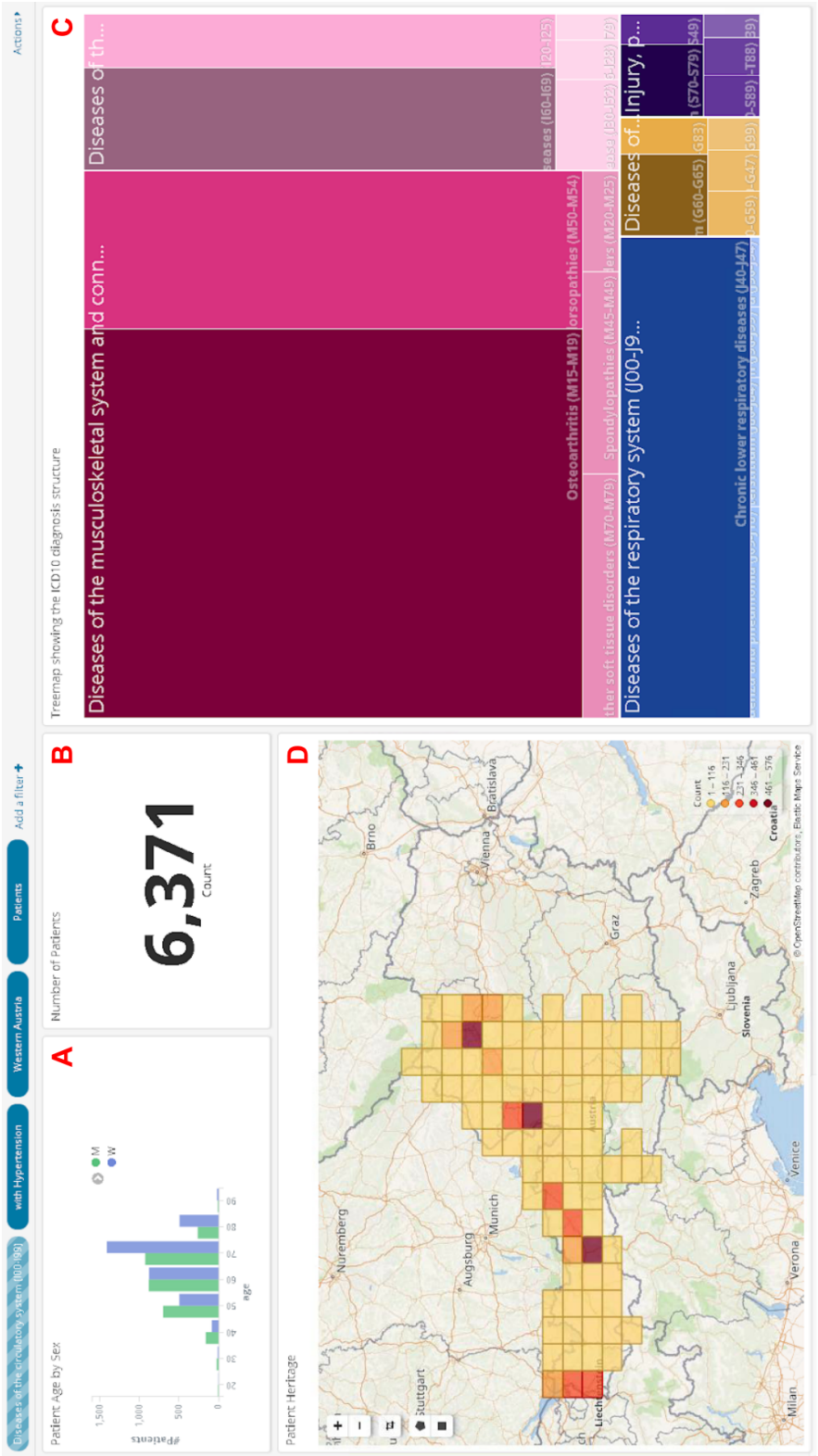

Figure 6.9. The dashboard of Task Exp3 before diseases of the circulatory system are selected. A: Detailed categorical visualization showing patient sex by age. B: Number of patients in selection. C: treemap showing all diagnosis chapters. D: Geographic visualization showing the patient distribution.

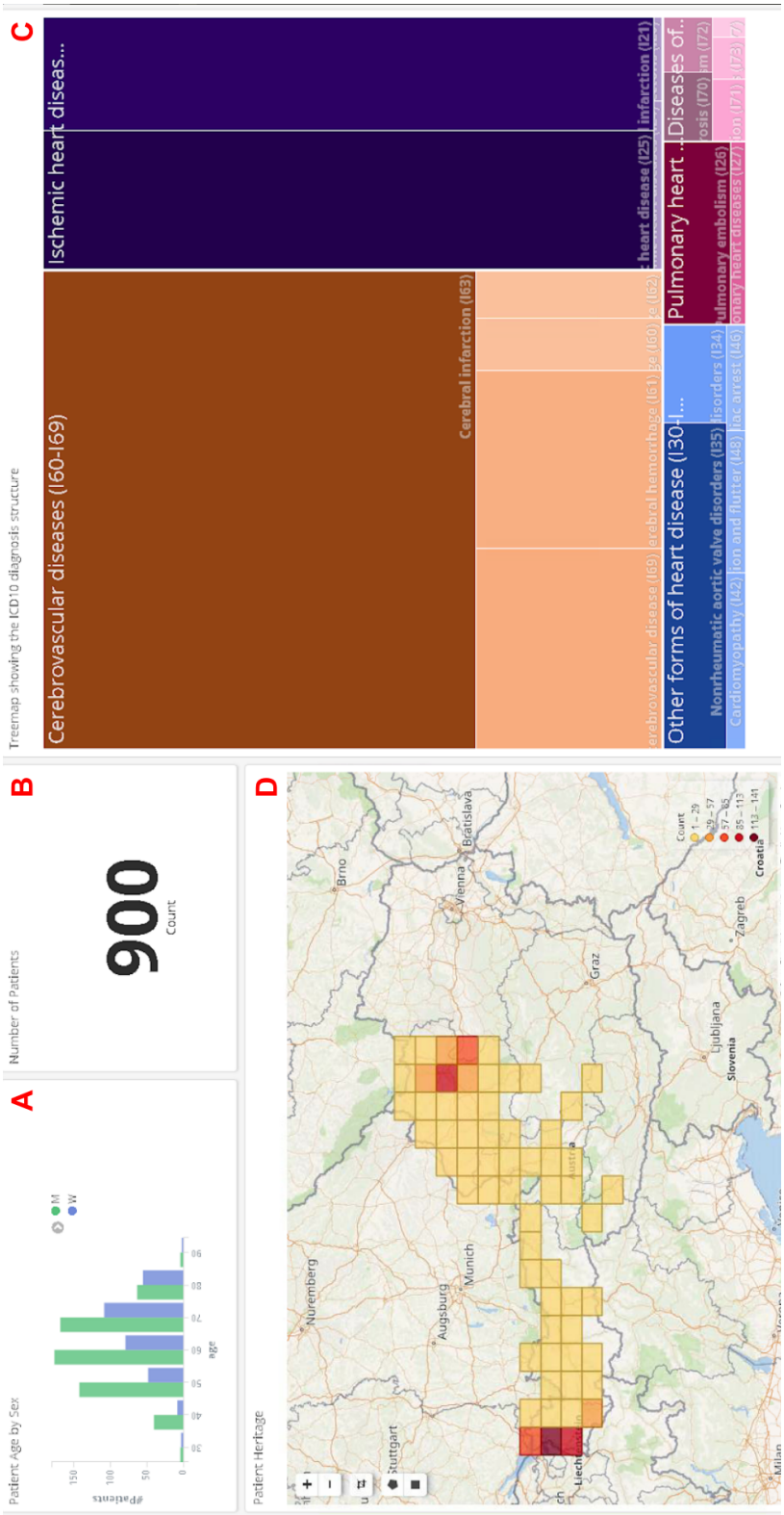

Figure 6.10. The dashboard of Task Exp3 after diseases of the circulatory system are selected. A: Detailed categorical visualization showing patient sex by age. B: Number of patients in selection. C: treemap showing the diagnosis chapter diseases of the circulatory system. D: Geographic visualization showing the patient distribution.

comparing Figure 6.9A to Figure 6.10A. If diseases of the circulatory system are filtered, a clear male dominance becomes visible.

Additionally, the treemap (Section 4.6.3) that displays the hierarchical structure of the ICD10 diagnoses as used in Eng1 is added to the dashboard, as it can be seen in Figure 6.9C. This treemap enables the domain expert to explore how the clinical condition of the patients compares to the rest of the data. After the chapter with diseases of the circulatory system is selected, the next level of the tree comes into focus, comprising the sections of this chapter only. We see that cerebrovascular diseases, which include for example hemorrhagic stroke, are the most frequent section of this disease chapter.

6.1.8 Usage Scenario of Exp4: Clinical analysis - Analyse data for clinical studies

The domain experts clinical work in the rehabilitation facilities also includes performing scientific research. The enforcement of a constant improvement of treatment strategies and assessment tools is the overall goal of this research. From the perspective of the domain experts we now want to see the data of all patients with diseases of the circulatory system and hypertension for the time of X to determine a set of correlating features in the cohort. The dashboard created for this purpose is given in Figure 6.11.

For tasks like these and others, the domain experts use the dashboard prepared by the engineers in task Eng1. The data table (Section 4.6.7) at the bottom includes the two minutes walking score. This score measures how many meters a patient can walk in two minutes. The score at admission and at discharge are given in separate columns each. We will not describe further details on the usability of the interactive visualizations, as they are already described in Task Eng1. We just mention here the ability to directly download a CSV structure from the data table, which can be seen at the bottom left of the visualization in Figure 6.11D. This feature can be used to export the dataset for further analysis in an external software component.

6.1.9 Usage Scenario of Exp5: Intervention planning - Use machine learning for intervention planning

In order to maximize the rehabilitation outcome of the patients, it is up to the clinicians to modify the clinical intervention. Machine learning will be an important tool for *"calculating and extracting the main factors for a successful rehabilitation"*, according to a domain expert. There is need to make the machine-learning functionality available to the domain experts, with respect to their a-priori knowledge in the field. From the domain experts' point of view, it is desirable to plan the clinical intervention with the highest possible outcome for the patient. The dashboard we use for the usage scenario of Task Exp5 can be seen in Figure 6.12.

In order to support the clinical intervention planning, we use the visualizations in the dashboard to create possible subcohorts of patients. We can use the detailed categorical

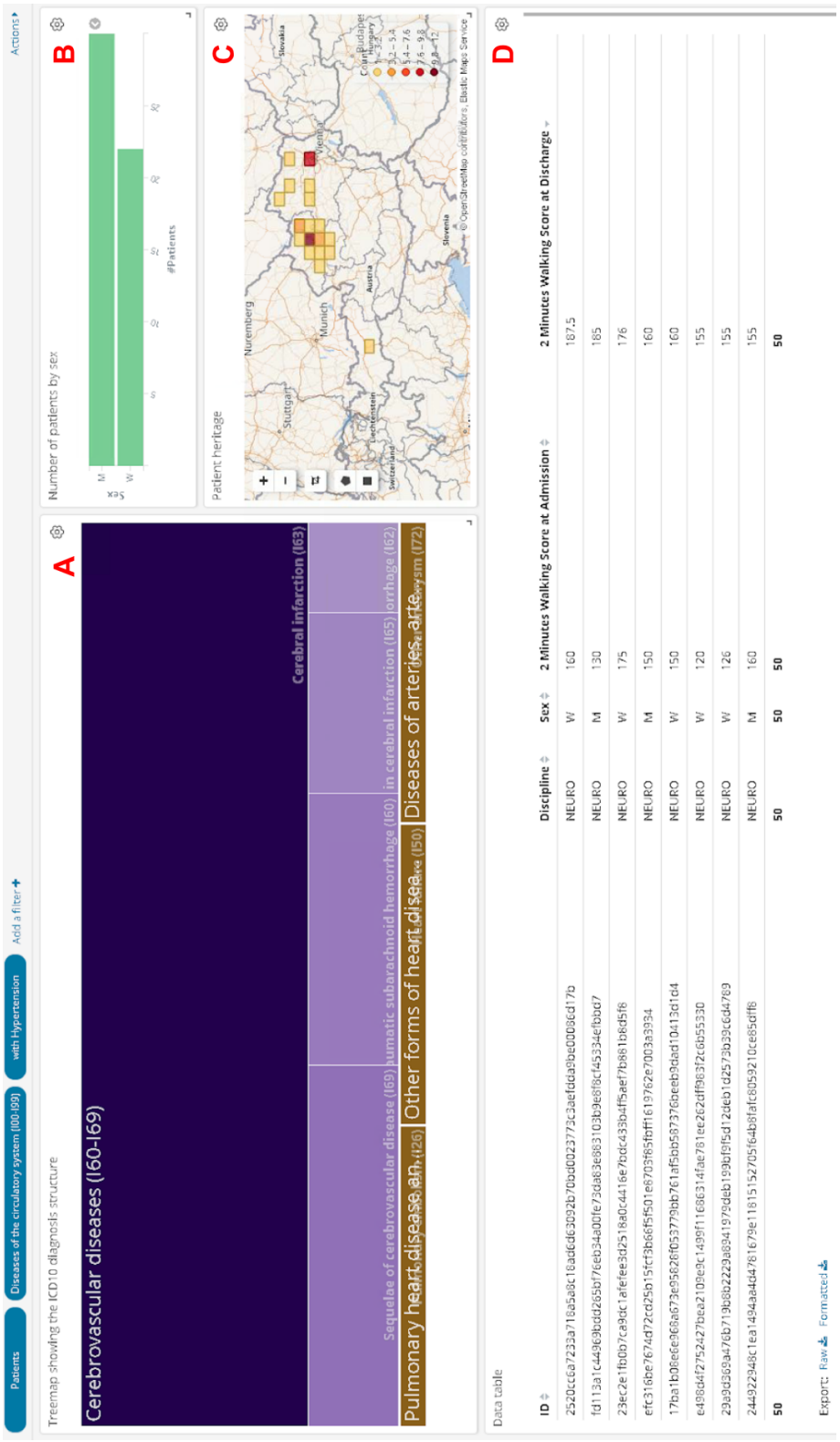

Figure 6.11. The dashboard of Task Exp4. A: treemap showing the diagnosis chapter on diseases of the circulatory system. B: Categorical visualization showing the patients' sex. C: Geographic visualization showing the patient distribution. D: Data table with the patients' score results.

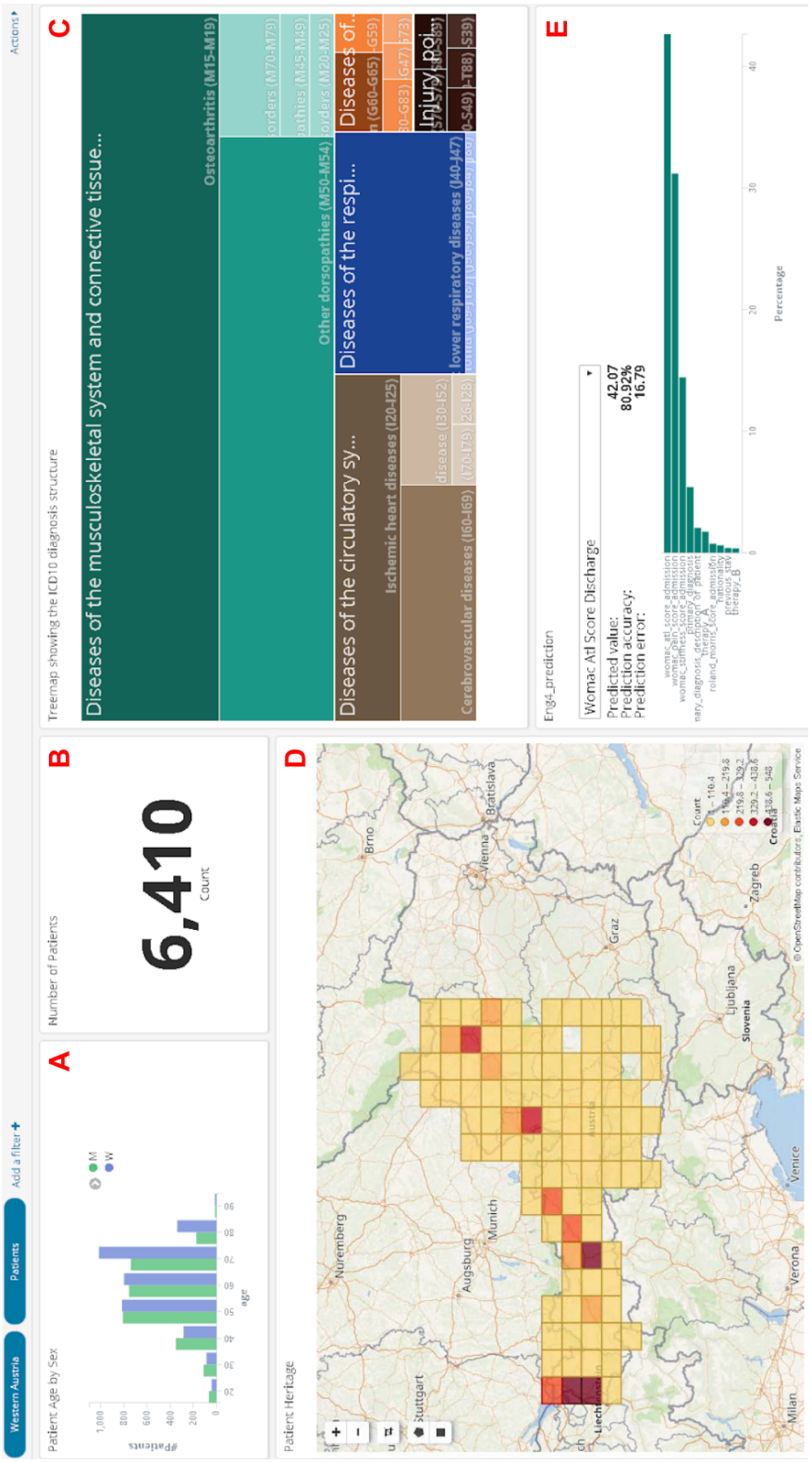

Figure 6.12. The dashboard of Task Exp5. A: Detailed categorical visualization showing patient sex by age. B: Number of patients in selection. C: Treemap showing the diagnosis chapter on diseases of the circulatory system. D: Geographic visualization showing the patient distribution. E: Machine-learning visualization showing the value and performance metrics of the current prediction.

visualization (Section 4.6.5) to set the age and sex for the subcohort. This can be achieved by selecting a bar that reflects the sex from the bucket that reflects the age group. With this interaction, two filters are created for the visual query. Also, we may use the treemap to refine the patients' clinical condition. Every time a subcohort is created, we monitor the results of the machine-learning prediction for the discharge scores of the patients in this subcohort. The machine-learning visualization (Section 5.4.2) displays a predicted WOMAC ATL outcome score of 42.07, at a precision of 80.92%, and a mean absolute error of 16.79. At this point, we can make minor adjustments in our query and inspect the result of the corresponding prediction. This results in a series of *What if?* queries. What if the age is changed to a range from 60 to 65? What if we add 60 minutes of this therapy? We face an iterative process, with possible adaptions of the clinical intervention at its end. Also we can detect features that have negative effects on the rehabilitation outcome in the dataset.

6.2 Pilot Study

In the evaluation section, we even go one step further and evaluate preha with our potential users. As a disclaimer, we have to mention here that we exclude the domain expert tasks from the evaluation. This is due to the fact that our application is in a very early stage. As for the technical know-how of the engineers, we propose to gather their comments and concerns first. For this reason we conduct an early pilot study with them. The results from the evaluation with the engineers will provide the necessary feedback to revise our approach.

We stick to the usage scenarios described before for the evaluation. While the usage scenarios were rather focused on a generic view of the many features of preha, we find it more suitable that the users stick to well defined assignments. Therefore we provide a fictive but realistic assignment to be accomplished by the user. After the users have completed the assignment, we ask them to provide statements on how the application helped them to accomplish the task. The results of this statements are reported for each task. We do not record quantitative features (e.g., time to complete a task) of the user evaluation, as the sample (n=4) is too small for statistically relevant results.

The nested model as proposed by Munzner [Mun09] is used for the evaluation, as it can be seen in Figure 4.1. The first level of the model is dedicated to learning about the tasks and data that the potential users of the visualization are dealing with. As tasks and data are already defined in the description of the evaluation assignment, we exclude them from the evaluation. In the second level, attention is paid to mapping the problems and data of the real world into the abstract tasks of information visualization. Again, we exclude this level from our evaluation as this mapping has already been performed for the assignments. The third stage is about designing the visual encoding and interaction. Here we focus the evaluation on the effectiveness of the encoding and interaction techniques for the proposed task. In the final level of the nested model, the algorithm for the visual encoding and interactions are in our focus. In our evaluation we examine the performance

of the used algorithms.

Each evaluation session held with the users is started by an explanation of the system. This explanation starts with an overall introduction of the system and what the intended tasks are. An important aspect of our explanations are the dashboards as the centerpiece of our system. We explain the relationship between visualizations and dashboards as well as which different types of visualizations are supported by preha. Another important point is which filters are available and which visualizations in a dashboard are affected when filters are applied. Also we explain the time filter and how it is related to the observed data. We end our explanations with a showcase visualization, a simple bar chart that shows the age distribution in groups of ten. In the remaining section we present the results of the pilot evaluation of Task Eng1-Eng4. It is not possible to do the evaluation of Tasks Exp1-Exp5 with the engineers, as we may face possible abstraction dangers. For each task in the following section, we describe the assignment and summarize the users' statements on the task.

6.2.1 Pilot Study of Eng1: Meaningful partitioning - Provide meaningful subsets of data

Assignment

Fictional domain expert Dr. Stephen Strange works at a rehabilitation clinic with a focus on orthopedic patients. Part of Dr. Strange's work is to perform scientific research on methods in orthopedic rehabilitation. Dr. Strange is interested in performing a retrospective study on knee diseases and asks you to prepare a dashboard in preha, with a data table as central point of attention. Dr. Strange wants to perform research on knee injuries. For this reason data for a retrospective study needs to be conducted, on the following subcohort:

- The age of the patients shall be above 50
- The primary diagnosis of the patients shall be M17, Gonarthrosis.
- Only female patients shall be present.
- Only patients treated in the second half of 2016 (1st of July to 31st of December).

The result of this task shall be a data table with a row for each patient in the cohort, with the columns: ID, age, primary diagnosis, WOMAC ATL score at admission and WOMAC ATL score at discharge.

The following questions should be answered:

- How many patients are present in the cohort?
- Visualize the distribution of the WOMAC ATL admission score. Create a bar chart that shows the average value for age groups of ten years.

Findings

As this is the first task each user has to accomplish, we predefine the encoding techniques to be used. In this case this is the data table. We observe that it takes all users a few minutes to find out how to add a new visualization. This could have possibly been prevented by additional explanations, but one user states: *"I prefer to find out some aspects on my own, this helps me to memorize it my own way."*. Also it is not clear to all users whether or not a dashboard must be added before creating the corresponding visualizations. A general observation we made in the first assignment is that the users struggle with thinking in dashboards and visualizations—prehas central GUI structure. Creating the subcohort did not rise any difficulties among the users, especially adding filter buttons is one of the users' preferred solution for filtering the subcohort. A significant problem we observed is that the users aim to change the time frame of the dashboard via normal filtering means, instead of using the time filter in the top right corner of the dashboard. Two users reported on having issues with what a metric and what a bucket (both are described in Section 5.4.2) is when creating visualizations. A brief introduction to these two elements does not seem to be sufficient.

6.2.2 Pilot Study of Eng2: Assessment templates - Prepare templates for patient health assessment

Assignment

In rehabilitation, it is important to include the patient as much as possible in the treatment process. Part of this process is to discuss the progress of the treatment at the time of discharge. The fictional domain expert Dr. Hank Pym asks you to provide a dashboard that supports this process. Dr. Pym wants to present the results of their two minutes walking test to individual patients. Furthermore it is of interest for the domain experts to show the patients how they compare to other patients of the same age group (10 years range), sex, and diagnosis. For this reason a dashboard with the following elements needs to be prepared:

- A visualization that shows the distribution of the two minutes walking-test at admission time and two minutes walking test discharge time at each.
- A metric that shows the results of the individual patient for admission and discharge each.
- A filter button that enables fast changing between individual results and all results.
- A filter button to set the sex of the patient.
- A filter button to set the age of the patient.
- A filter button to set the primary diagnosis of a patient.

The following questions should be answered:

- What are the peaks of the distributions of the two-minutes walking-test in the observed subgroup?
- What is the difference between the admission and discharge value of the individual patient?

Findings

The most important aspect of this assignment was to determine a good way to present the distribution of the two-minutes walking-test. Two users decided to use a line chart, two users decided to use a vertical bar chart. For creating the buckets of the distributions, the easiest way is using the histogram that only requires a bucket size to be specified. Some users decided to take the range criterion for buckets, where the range of each bucket must be specified, which takes more time to complete, compared to the histogram. To create a metric visualization that shows the average value of a score, the users had some difficulties about how a single figure can be used to show the average value of single patients as well as patient groups. We aided the users by explaining how filtering according to the patients ID produces just one result, calculating the average does not affect the results. Generally we observe that the users are now way faster in creating visualizations, compared to Task Eng1, dashboards, and filters, which is no surprise as they start getting familiar with the system after having completed the first task. Still, the users tend to be overwhelmed in choosing visualizations. This may be caused by the fact that they are not used to freely choosing encoding techniques. The questions defined were answered easily by all participants of the evaluation.

6.2.3 Pilot Study of Eng3: Benchmarking templates - Prepare templates for clinical benchmarking

Assignment

Clinical benchmarking is a tool used by healthcare facilities to monitor and improve their quality and efficiency. Ms. Natasha Romanova, head of the administrative staff at rehabilitation clinic X, needs some key indicators on the clinical performance. For a specific timeframe it is of interest, how many patients received a treatment in total, how the patient admissions developed, how the WOMAC ATL discharge score developed, and how the ratios of payers are partitioned. For this reason a dashboard with the following elements needs to be prepared:

- A metric that shows the total number of patients.
- A bar chart showing the portions of the individual payers.
- A line chart that shows the development of patient admissions.
- A line chart that shows the development of the WOMAC ATL discharge score.

The following questions should be answered:

- Describe the development of the WOMAC ATL discharge score in the course of 2017.
- Identify two outliers in the patient admission development line chart. What are possible reasons for these?

Findings

For Eng3 we again experience an improvement on how confident the users are with using preha. One user states that the learning curve is *"a bit steep first, but once you are familiar with the basic interactions, the system feels very intuitive"*. In this task we define, which visualizations are to be used by the participants. None of the users has troubles to create the metric visualization. Up to now, the users were supposed to use histograms for creating the buckets. The qualitative value to be selected here is the payer of the rehabilitation, which is a nominal variable. For this reason, no histogram can be employed for the qualitative aspect, instead the users have to select the "terms" bucket. Two users state that they would prefer using pie charts over bar charts for visualizing the parts of the individual payers. As the X-axis of the line charts to be created in this task corresponds to a timeline, using a histogram is not possible in this case either. Instead a specific date histogram has to be used, which takes all users some time to figure out. No participant pointed out any development of interest in the WOMAC ATL discharge-score line-chart. On the other hand, the users were able to clearly see the outliers in the patient admission development, which was explained by Christmas and Easter holidays.

6.2.4 Pilot Study of Eng4: Outcome predictions - Use machine learning to predict rehabilitation outcome

Assignment

For the treatment of osteoarthrosis, therapy A has been the state of the art for years. Rehabilitation Centre Y has been testing therapy B quite successfully concerning the same task for a while now, randomly assigning therapy minutes of both therapies to the patients. For both therapies, the WOMAC ATL score is the best feedback measure. Now it is of interest to determine the effectiveness of both therapies. For this reason, a prediction for the discharge score shall be defined that takes the therapy minutes for each therapy into account. Prepare the following dashboard:

- A visualization that enables specifying the minimum and maximum number of minutes for the two therapies affecting the prediction.
- A machine-learning visualization that predicts the score at discharge.

The following questions should be answered:

- What are therapy A and therapy B?
- How would you describe the impact of therapy A on the WOMAC ATL score at discharge?
- How would you describe the impact of therapy B on the WOMAC ATL score at discharge?

Findings

Eng4 is the last, but also the most complex task for the engineers. All users start with adding the machine-learning visualization to the dashboard. As this is the first time the users employ this visualization, they spend some time to analyze its capabilities. The machine-learning visualization is rather complex compared to the other visualizations, some users ask additional questions at this point in order to fully understand its capabilities. After the users are aware of the machine-learning visualizations' functionality, they have no difficulties learning what therapy A and therapy B are. The majority of users aims towards using filters for specifying the minimum and maximum number, which is an obvious solution. An alternative solution is using range sliders, which can be seen in Figure 6.6. Now the participants are able to modify these variables and monitor the differences in the predicted WOMAC ATL scores.

6.2.5 Pilot Study Outcome

After the four tasks have been evaluated with the users, we ask the users for general comments on preha. All users agree that preha is capable of realising the tasks that were worked out together. Furthermore the users highlight the multiple coordinated views [WBWK00] in preha as a central feature and main advantage. The flexibility of the dashboards, including rearranging and resizing visualizations, is a functionality especially liked by the engineers. In the course of this diploma thesis, we designed preha entirely in English, however the users suggest to provide preha in the users' native language. All users state that they would have required more knowledge to start with the first assignment. On the other hand, the users report that exploring the system on their own helped them getting to know preha in their own working style. Another general comment on preha is the time-range filter and its location at the top right, as it can be seen in Figure 5.5 at (c). The engineers further suggest to prepare extensive training material including a lot of examples, before approaching the domain experts. One engineer states *"the domain experts are not used to work with tools such as preha, they lack required technical knowledge"*.

Preha includes some highly specialised and complex features, as the first evaluation stage with the engineers demonstrated. In the course of this evaluation, a few particular aspects were pointed out by the engineers. For example, a lack of extensive documentation, a steep learning curve in the beginning, trouble distinguishing metrics and buckets, the placement of the time filter,

and the oversupply of visualization types. Using this feedback from the engineers, we can improve the functionality of the tool and design appropriately an evaluation that will include also clinical domain experts.

6.2.6 Machine Learning Performance

As already mentioned in the introduction of this chapter, our evaluation also includes the analysis of the used machine-learning algorithm. Prehas machine-learning module predicts values based on a set of input parameters. To do so, a random forest is created independent of the applied input parameters, which are only needed for the prediction. In order to evaluate the performance of the machine learning module, we use the REST interfaces it provides.

Performance Tuning

The first target of our evaluation is to determine the influence of various parameters used for the performance of the prediction. Even though, numerous parameters can used to tune the algorithm, we focus on the two most important ones in terms of the algorithm complexity: the number of trees and the maximal depth of each tree. The depth of a tree is determined by the number of layers where decision nodes are located. Referring to Figure 4.9, the depth of tree1 is six and the depth of tree2 is five. As score for the prediction, we decide for the six minutes walking test that is present in about 2,000 cases. This test requires a patients to walk for six minutes straight, the walked distance in meters is measured. A typical value for this score is 300 m to 600 m. About 650 features are taken into account for this score's random forests. For each prediction we get the calculation time, the accuracy, and the mean absolute error returned from the machine-learning module. All measurements for the evaluation were taken based on the full dataset.

```
for tree in {1..20}; do
    for max_depth in {1..20}; do
        for i in {1..3}; do
            metrics+=predict_value("6_minutes_walking",$tree,$max_depth)
        done
        metrics=average(metrics)
        print(metrics)
    done
done
```

Listing 6.1. Pseudo code for capturing the six minutes walking score performance metrics of the machine learning module. Various combinations of the number of trees and the tree depth from a range from 1 to 20 are applied. The performance metrics are saved in the variable metrics.

Listing 6.1 shows a pseudo code of the bash script used for the evaluation. This simple script is based on three nested for-loops. The outer two loops determine the values for the number of trees n_{trees} and the maximal tree depth $depth_{max}$. These parameters are the inputs for the random-forest regression, along with the dependent variable

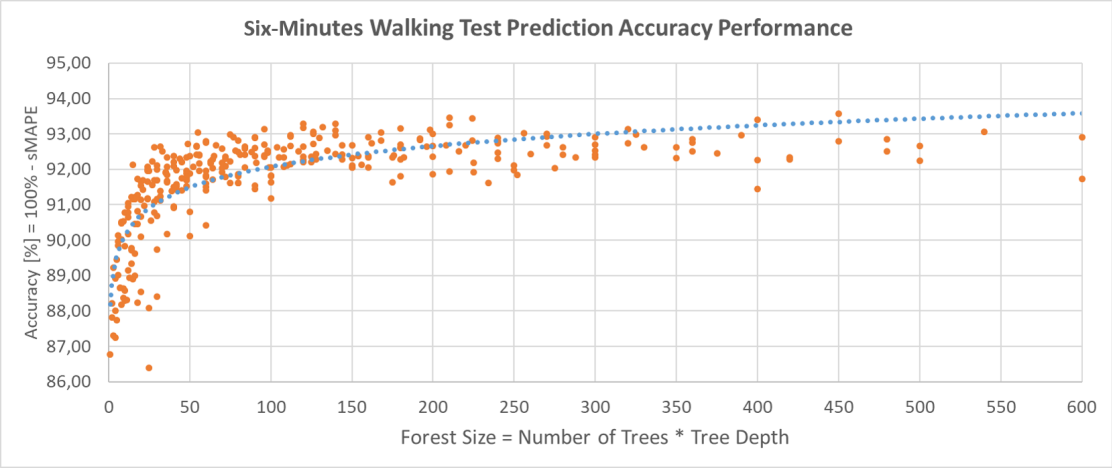

Figure 6.13. Accuracy performance of the six minutes walking test prediction.

6_minutes_walking. This operation is done three times and the average value of each performance metric is calculated, leading to less scattered results. For the analysis, we present each performance metric in relation to the forest size F that we define as $F = n_{trees} \cdot depth_{max}$. This enables the comparison of the performance metrics with respect to a single parameter and is easier than a two dimensional representation. We visualize each performance metric with a dot plot, where each dot corresponds to the average of the three measurements.

The initial performance metric we will focus on is the accuracy of the prediction that was already introduced in Section 5.3. Our analysis shows that the accuracy of the prediction grows logarithmically with the forest size, as displayed by the blue trendline in Figure 6.13. We document an initial accuracy of about 86% (with $n_{trees} = 1$ and $depth_{max}$=1). An interesting point to observe is the accuracy at F=100, where the logarithmic growth stagnates at an accuracy of 92%. Up to F=600, no further increase of the accuracy can be observed via means of F.

A similar behaviour can be observed for the mean absolute error (MAE), as is depicted in Figure 6.14. The red trendline of the MAE shows an exponential decay with increasing F, mirror-inverted to the accuracy. At F=1 the MAE is about 95. Again F=100 can be observed as the significant value, where no further decrease occurs. The MAE settles down at 45.

Finally, we analyze the calculation time of the prediction, which can be seen in Figure 6.15. The calculation-time trendline shows a linear behavior over F, whereby scattering is directly proportional to F. For random forests with little complexity, the time required for the calculation is about 15 seconds. The calculation time further increases even beyond F=100, from where the other performance metrics remain constant. Please remind, that a new random tree in preha is only generated when the prediction target is changed. A new prediction for an existing target is much fast and takes about two seconds. From

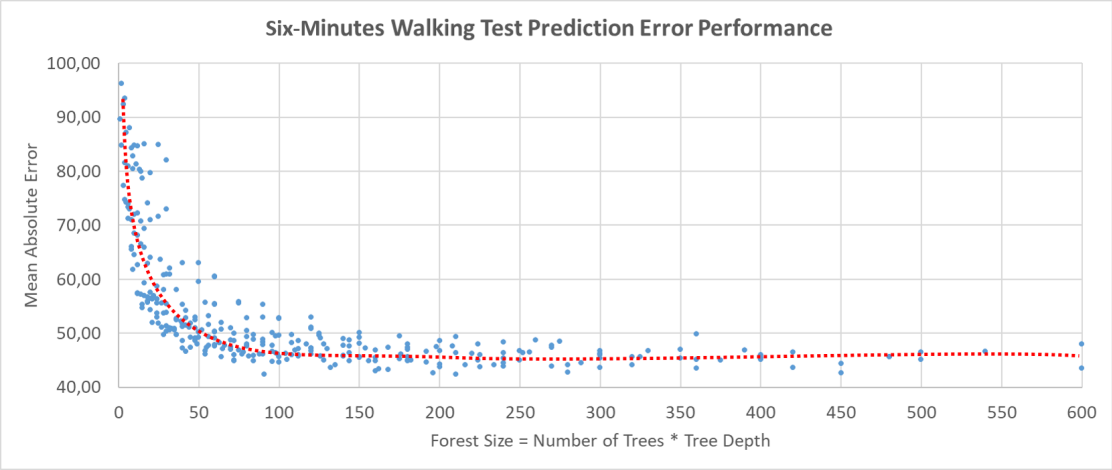

Figure 6.14. Mean absolute error performance of the six minutes walking test prediction.

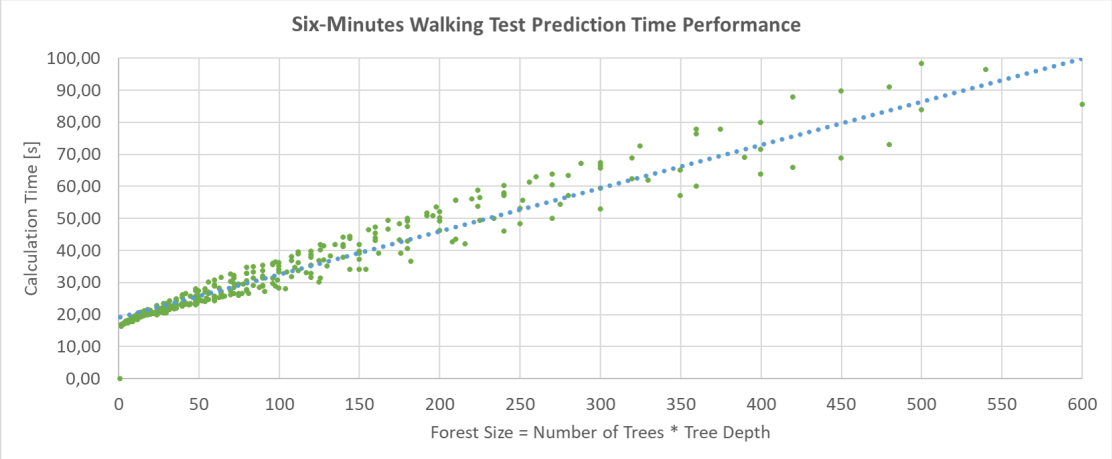

Figure 6.15. Temporal performance of the six minutes walking test prediction.

the detailed results of our evaluation, we observed F=33 as the sweet spot for further predictions. At this level a good accuracy and MAE can be achieved at a moderate calculation time of about 22 seconds. A forest with eleven trees and a max depth of three will further be prehas default setting. At this point we have to state that this setting may not be the sweet spot for all prediction targets, which have different data characteristics.

Sample Size of the dataset

The ultimate goal of precision rehabilitation is to base clinical intervention planning on predictions based on EHR data. It is of interest to determine whether or not the EHR dataset is "ready" for precision rehabilitation or not. As is the case for all machine-learning tasks, the level of precision depends on size of the underlying dataset. This is the reason, we performed our second analysis session. Now we observe the development of our three performance metrics over a random sample size of the dataset. The forest

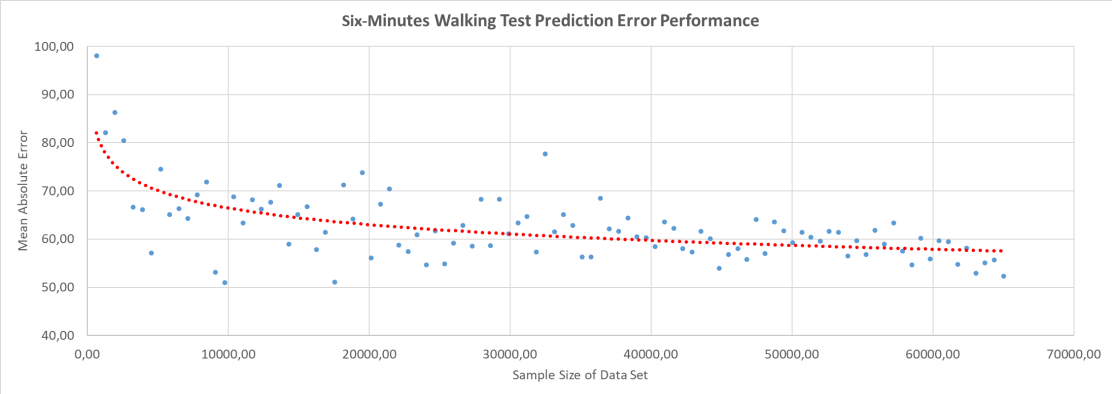

Figure 6.16. Mean absolute error performance of the six minutes walking test prediction.

size is fixed at F=33. Again, we visualize each performance metric with a dot plot, where each dot corresponds to the average of three measurements.

```
for sample_size in {1..100}; do
    for i in {1..3}; do
        metrics+=predict_value("6_minutes_walking",$sample_size)
    done
    metrics=average(metrics)
    print(metrics)
done
```

Listing 6.2. Pseudo code for capturing the six-minutes walking-score performance metrics of the machine-learning module. A varying sample size from 1% to 100% is applied. The performance metrics are saved in the variable metrics.

From the 46,000 cases present in our dataset, the requested score was absolved 2,000 times. This explains the scattered results of the measurements in the corresponding figures of the performance metrics. Again each measurement was performed three times and the average of the measurements was calculated. Listing 6.2 shows a pseudo code of the bash script used for the evaluation. Figure 6.16 shows the development of the MAE over the sample size. The MAE quickly settles at 60 for a sample size of 10,000 patients. The red trendline indicates exponential decay. With increasing sample size, the scattering decreases and a MAE of about 52 is the value achieved at a sample size of 100%.

A similar picture emerges for the accuracy-performance metric. Scattering is again present in the results of the measurements. An interesting observation is that the MAE seems to show higher convergence with increasing sample size, compared to the accuracy. This is not a significant observation, but makes the MAE appear more reliable as performance metric. At a sample size of 10,000 the accuracy of the prediction is about 90%. If the full dataset is used as sample the accuracy is about 92%. The blue trendline shows logarithmic growth.

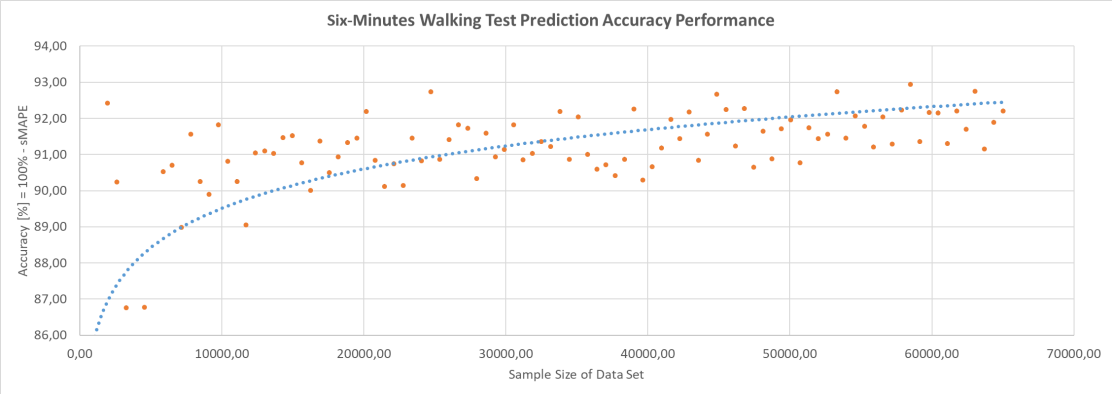

Figure 6.17. Accuracy performance of the six minutes walking test prediction.

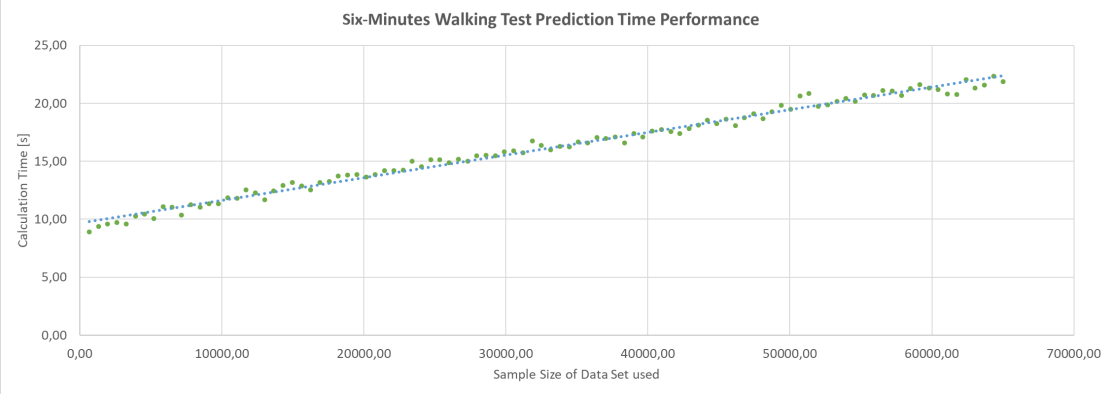

Figure 6.18. Temporal performance of the six minutes walking-test prediction.

Unsurprisingly, the calculation time again shows a linear behavior, as visible by the trendline in Figure 6.18. The calculation time starts at nine seconds for 1% sample size, rising to 22 seconds for the full dataset. This is a significant performance boost compared to the calculation time from the first part of our evaluation. The fixed forest size (F=33) enables the calculation time to stay within the limits.

At this point it is up to the users to decide how they want the accuracy and the calculation time of the predictions to be tuned. If it is the case that the same prediction target is called frequently or large computational resources are available, high sample and forest sizes may be chosen. In the analyzed dataset we reached the limits of the machine-learning module. A further increase of the prediction accuracy is only possible by storing additional features in the electronic health record.

6.3 Critical Reflection

Preha was designed to be capable of all functionalities that have been set by the task analysis. A series of design choices lead to the implemented system with all its modules.

Using kibana as the underlying Visual Analytics framework was the design choice that introduced the most restrictions to the overall system. In contrast to other systems presented in this thesis, such as S-ADVIsED by Alemzadeh et al. [AHN+17] or CAVE by Zhang et al. [ZGP15], this system was not designed for the targeted users from scratch. Possible usability issues will be pointed out in the future domain experts' evaluation with the revised preha version that includes the engineers input. Extensibility is also an interesting aspect to discuss at this point. As no specific software was used to build preha, we aim to highlight the possibility to use it for other tasks or domains. The most important aspect when doing so, is to decide which object is abstracted to be shown in the dashboard. In the present field of rehabilitation, we decided on using a clinical case for this purpose. Further possible use cases for our application will be discussed in Section 7.2.

Another aspect that needs to be addresses is also caused by the design choice to use kibana as the underlying framework. kibana's focus is on the visualization of aggregated data. Despite the fact that visualizing non-aggregated data was not required by the users in the interviews, this may be a future requirement. On the one hand, a way to realise this in kibana is to use custom vega visualizations that deal with non-aggregated data. On the other hand there is also the possibility to aggregate the data by ID, resulting in non-aggregated data.

The primary data storage in preha is realized as plain CSV file. Both, the Visual Analytics dashboard and the predictive-analytics engine access this storage as data source. Reasons for this design decisions have been described in Section 4.4. In practice, the data is persisted a second time, when it is loaded to elasticsearch for the visualization in kibana [GT15, Gup15]. In the presented case with a static dataset this is not an issue. Still, if preha were in productive operation, measures to keep both data sources consistent need to be applied. Alternatively, it would be possible to use elasticsearch as primary data storage that is also accessed by the predictive-analytics engine. For large subcohorts, using elasticsearch as primary storage may result in significant performance decrease due to larger network traffic.

As for scalability, there are multiple aspects to discuss. The first is the scalability of the machine learning algorithm. As the results of our performance evaluation show, the runtime of the prediction tasks increases linearly for both algorithmic complexity and dataset size. This behaviour can be seen in Figure 6.15 and Figure 6.18. The next aspect is scalability in terms of dataset size. Apart from providing sufficient disk storage, there is no issues to be expected. This is caused by the fact that a CSV file is used as primary data storage. Finally, the dashboard performance needs to be discussed. In the present dashboards, we noticed increasing loading times when querying large subcohorts. To counter this effect, sufficient computational resources need to be provided.

CHAPTER 7

Summary and Future Work

7.1 Summary

This thesis contributes with a novel framework called preha that enables the introduction of Visual Analytics to the field of rehabilitation. Preha may be used to analyze rehabilitation data and utilize the outcome to enable precision rehabilitation. This analysis may lead to improved rehabilitation outcomes and a deeper understanding of the rehabilitation process. In the design of preha, requirements that are specific to the field of rehabilitation were taken into account. Those requirements are the result of interview sessions conducted with engineers and domain experts from the rehabilitation field. In tight collaboration with these potential users, nine tasks have been identified. The engineers' tasks include the preparation of meaningful data subsets for clinical research, providing visualization templates that are used in the clinical process, and the application of machine learning algorithms to support the domain experts in clinical intervention planning. The domain experts' tasks involve presenting data to the patients, exploring the dataset to identify possible research targets, performing clinical benchmarking, or applying machine learning to determine new intervention strategies.

We solve these tasks by introducing preha, a solution that is built on top of multiple modules. A preprocessing module performs all necessary preparation steps before adding the data to the system. The primary data storage persists all data needed for the analysis. Prediction tasks are performed by a machine-learning module. All visualization tasks take place in the Visual Analytics dashboard. The dashboard provides a range of interactive visualizations that enable the users to interact with the underlying data. Multiple dashboards have been designed for the specific requirements of each task and user. With this solution, we answer our research questions:

"How can we identify the parameters that are responsible for a successful rehabilitation within a large set of high-dimensional data? Which are the most relevant and important

parameters for a successful rehabilitation?" - This question is addressed by the machine learning module. The applied machine learning algorithm—random forest regression—provides the variable importances for each prediction. This enables the identification of the features responsible for a successful rehabilitation.

"How can VA aid data analysts to extend their knowledge about rehabilitation outcome to improve the rehabilitation process as such?" - Preha provides a variety of visualizations such as tree maps, bar charts, line charts, or data tables. This enables the users to gain insight in multiple aspects of the data from different perspectives. Particularly the combination of those visualizations allows the engineers and domain experts to acquire additional knowledge on the rehabilitation process.

"How can the predicted and the actual outcome of the rehabilitation be visualized and analyzed comparatively?" - The interactive dashboards enable the integration of multiple visualizations in a single GUI. Aligning the visualizations of the predicted and the actual rehabilitation outcome with each other allows for comparative visualizations.

This diploma thesis supplies several contributions. The main contribution is the design and implementation of a Visual Analytics application that allows domain experts and engineers to fulfil their data analytics tasks in the rehabilitation context. Also, we performed a thorough user task analysis set in the rehabilitation field. Our visualization design chapter provides a collection of techniques and algorithms that can be used for the proposed problem. Furthermore, we introduce a comprehensive tool for preprocessing rehabilitation data, including cleaning non validated health assessment scores. Another contribution is the design and implementation of an interactive tree map visualization for kibana [1]. To the best of our knowledge, the proposed solution is the very first application of Visual Analytics in the field of precision rehabilitation.

7.2 Limitations and Future Work

In this diploma thesis we introduced a solution that brings various advantages to data analytics in the proposed setting. Flexible dashboards, interactive visualizations and machine learning integration, are a few. However, there is need to discuss limitations and possible future improvements of our approach.

In the design of our application, we decided to implement random forest regression as algorithm for the prediction of rehabilitation outcomes. Even though the results we achieved with this algorithm are satisfying, other algorithms could lead to more precise results or improved performance. By designing our machine-learning algorithm as separate module with clean REST interfaces, we assure easy changeability of the algorithm. Bennett and Doub [BD11] compare the performance of multiple machine learning algorithms for analyzing EHR data. Possible future work would be to evaluate the same algorithms in our application and to compare them to the results of Bennett and Doub.

[1] `https://github.com/bgeVam/kibana_treemap_visualization`

A topic that was already addressed in Section 6.3 is the potential usage of elasticsearch as primary storage [GT15]. As eventual future improvement of preha it would be possible to implement elasticsearch and evaluate the resulting changes. Interesting aspects in this context include compare the performance of prediction tasks in terms of response time. Other performance aspects such as network traffic, CPU utilization, RAM usage, or required disk space also need to be monitored.

Another aspect that should be considered in future work, is the integration of prediction history. This is a feature that has been suggested by the engineers in the course of the evaluation. There are two potential starting points for implementing this functionality. The first is caching results of past predictions in the machine-learning module. This improves the performance of the predictions. The second component that will benefit from a prediction history is the Visual Analytics dashboard. An overview of already performed predictions, including the applied filters, enables better monitoring of changes in the prediction output.

Additionally, a possible integration of the prediction engine in the rehabilitation process or even the EHR may be discussed. As soon as a new patient is registered along with all relevant data, a set of predictions is performed to determine the patients possible rehabilitation potential. This may be helpful to set goals for the rehabilitation. After the rehabilitation process, the domain experts can come back to the patients to compare and discuss the predicted and the actual rehabilitation outcome.

Moreover, we want to mention the future evaluation with the domain experts at this point. The engineers provided useful feedback on preha, including usability aspects such as the placement of the time filter and the oversupply of visualization types. A major suggestion made by the engineers is to reserve a sufficient amount of time and provide a training unit with the domain experts, before diving into the system. In addition, the engineers suggest to provide a handbook for the users.

A few particular aspects were pointed out by the engineers. Examples include a lack of extensive documentation, a steep learning curve in the beginning, trouble distinguishing metrics and buckets, the placement of the time filter, and the oversupply of visualization types. Problems such as those mentioned may be addressed by applying guided visual analytics strategies [CGM+17]. Guidance is a dynamic, iterative and forward oriented process, aiming to help users with their investigative work in Visual Analytics. Examples for guidance are providing cues or alternative options for solving Visual Analytics tasks.

The visual analytics dashboard used in preha, i.e., kibana, is not specifically developed for the usage in rehabilitation or medicine. It is a software that is not designed to solve a specific domain problem, but rather a generic solution with a focus on visualizing performance and log data. In retrospective to the approach presented in this diploma thesis, we conclude that preha can be used with any kind of rehabilitation data, there is nothing specific to orthopedic or neurological rehabilitation. We further see no issues with using kibana for general EHR visualization approaches, as long as just the information visualization aspect from Visual Analytics is applied. The application of kibana to other

medical disciplines such as biomedicine would be a highly interesting aspect to further investigate on. One interesting use case may be the visualization of the prevalence of a disease. Another possible application in biomedicine would be showing the genes of a genome. Finally we see no point in restricting potential use cases of kibana to the medical domain, as it may be helpful for all information-visualization approaches.

List of Figures

List of Tables

Listings

Glossary

Anschlussheilverfahren See subsequent therapy.. 137

cohort A cohort is a subset of a statistical population sharing common characteristics. In a population of all stroke-patients, a cohort could be "all male patients, above the age of 50, having their first stroke before the age of 40".. 35, 137

CSV Comma-separated values (CSV) is a text file with a special delimiter to separate values. CSV files store tabular data, each line of the file is a data entry. The default separator for CSV files is the comma, as the name suggests.. 44, 52, 137

electronic health record An electronic health record (EHR) is a "longitudinal collection of electronic health information about individual patients and populations" [GT05]. A variety of information is stored in the EHR: biometric information (age, height, weight, ...), clinical information (diagnose, patient state, ...) or demographic information (patient residence, ...).. 14, 28, 29, 32, 34, 44, 80, 137

federal pension fund This is the English term for the "Pensionsversicherungsanstalt" in Austria. Its main functions are paying of pensions, handling rehabilitation requests and take preventive health care measures.. 10, 137, 139

ICD-10 ICD-10 [Wor04] stands for "international classification of diseases" and is a constantly revised classification and coding system for diseases. It has been developed by the World Health Organization since 1983.. 13, 34, 137

ICF ICF [O$^+$01] stands for "international classification of functioning, disability and health". ICF was developed to provide a comprehensive classification to describe human functioning.. 13, 137

Pensionsversicherungsanstalt See federal pension fund.. 137

population Population is a statistical term, defining the total set of items that are of interest, e.g., for research. In medical studies, populations usually represent a number of patients sharing a common characteristic (e.g., a disease).. 35, 137, 139

range of motion The range of motion (ROM) is a measure that describes a distance or an angle that determines how moveable a patients joint is.. 33, 137

Rehab-Heilverfahren See rehabilitation therapy.. 137

rehabilitation therapy This is the English term for the "Rehab-Heilverfahren" in Austria. A rehabilitation therapy is a rehabilitation taking place after medical treatment by a facility without inpatient beds [GFI+16, KEP12].. 10, 137, 140

subsequent therapy This is the English term for the "Anschlussheilverfahren" in Austria. A subsequent therapy is defined as rehabilitation therapy that follows subsequently (up to 12 weeks) to the stay in a hospital [GFI+16, KEP12].. 10, 137, 139

Bibliography

[AHN+17] Shiva Alemzadeh, Tommy Hielscher, Uli Niemann, Lena Cibulski, Till Ittermann, Henry Völzke, Myra Spiliopoulou, and Bernhard Preim. Subpopulation discovery and validation in epidemiological data. In Michael Sedlmair and Christian Tominski, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2017.

[Alp10] Ethem Alpaydin. *Introduction to machine learning*. The MIT Press, 2nd edition, 2010.

[ANI+17] Shiva Alemzadeh, Uli Niemann, Till Ittermann, Henry Völzke, Daniel Schneider, Myra Spiliopoulou, and Bernhard Preim. Visual analytics of missing data in epidemiological cohort studies. In *Eurographics Workshop on Visual Computing for Biology and Medicine*, volume 4, 2017.

[BC87] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[BD11] Casey Bennett and Thomas Doub. Data mining and electronic health records: Selecting optimal clinical treatments in practice. *Proceedings of the 6th International Conference on Data Mining*, 2011.

[Bee13] Chris Beeley. *Web application development with R using Shiny: Harness the graphical and statistical power of R and rapidly develop interactive user interfaces using the superb Shiny package*. Packt Publishing, 2013.

[BL07] Renaud Blanch and Eric Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1248–1253, nov 2007.

[BM13] Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, dec 2013.

[BMA+00] Isaac Brewer, Alan M MacEachren, Hadi Abdo, Jack Gundrum, and George Otto. Collaborative geographic visualization: Enabling shared understanding of environmental processes. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*. IEEE Comput. Soc, 2000.

[BMMS91] Andreas Buja, John Alan McDonald, John Michalak, and Werner Stuetzle. Interactive data visualization using focusing and linking. In *IEEE Conference on Visualization 1991*, pages 156–163. IEEE, 1991.

[BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, dec 2011.

[Bos12] Michael Bostock. Zoomable treemaps. `https://bost.ocks.org/mike/treemap/`, 2012. Accessed: 2018-02-08.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Car99] Mackinlay Card. *Readings in information visualization: Using vision to think*. Morgan Kaufmann, 1999.

[CCB15] Florin Chelaru and Héctor Corrada Bravo. Epiviz: A view inside the design of an integrated visual analysis software for genomics. *BMC Bioinformatics*, 16(11):S4, Aug 2015.

[CCB+17] Dequan Chen, Yi Chen, Brian N Brownlow, Pradip P Kanjamala, Carlos A Garcia Arredondo, Bryan L Radspinner, and Matthew A Raveling. Real-time or near real-time persisting daily healthcare data into HDFS and ElasticSearch index inside a big data platform. *IEEE Transactions on Industrial Informatics*, 13(2):595–606, apr 2017.

[CCCD12] João Completo, Rui Santos Cruz, Luisa Coheur, and Manuel Delgado. Design and implementation of a data warehouse for benchmarking in clinical rehabilitation. *Procedia Technology*, 5:885–894, 2012.

[CGM+17] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jorg Schulz, Marc Streit, and Christian Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, jan 2017.

[CKB09] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2, 2009.

[CLG17] Jake R Conway, Alexander Lex, and Nils Gehlenborg. UpSetR: an R package for the visualization of intersecting sets and their properties. *Bioinformatics*, 33(18):2938–2940, jun 2017.

[CRM91] Stuart K Card, George G Robertson, and Jock D Mackinlay. The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 181–186. ACM, 1991.

[CV15] Francis S Collins and Harold Varmus. A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795, 2015.

[DdOL+18] Henrique Galvan Debarba, Marcelo Elias de Oliveira, Alexandre Ladermann, Sylvain Chague, and Caecilia Charbonnier. Augmented reality visualization of joint movements for physical examination and rehabilitation. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, mar 2018.

[DFE93] Jenny L Donovan, Stephen J Frankel, and John D Eyles. Assessing the need for health status measures. *Journal of Epidemiology and Community Health*, 47(2):158, 1993.

[DZK+12] Eugen Dyck, Eduard Zell, Agnes Kohsik, Philip Grewe, York Winter, Martina Piefke, and Mario Botsch. Octavis: An easy-to-use VR-system for clinical studies. *Virtual Reality Interaction and Physical Simulation*, 2012.

[EJP13] Pamela Enderby, Alexandra John, and Brian Petheram. *Therapy outcome measures for rehabilitation professionals: Speech and language therapy, physiotherapy, occupational therapy*. John Wiley & Sons, 2013.

[Eng77] George L Engel. The need for a new medical model: A challenge for biomedicine. *Science*, 196(4286):129–136, 1977.

[Est14] Soledad Estrella. El formato tabular: Una revision de literatura. *Actualidades Investigativas en Educacion*, 14:449 – 478, 08 2014.

[FBiB18] Ana Lúcia Faria and Sergi Bermúdez i Badia. Personalizing paper-and-pencil training for cognitive rehabilitation: A feasibility study with a web-based task generator. In *International Conference on Applied Psychology and Human Behavior*, 2018.

[FCDO80] Jeremy C Fairbank, Judith Couper, JB Davies, and JP O'brien. The oswestry low back pain disability questionnaire. *Physiotherapy*, 66(8):271–273, 1980.

[FiB15] Ana Lúcia Faria and Sergi Bermúdez i Badia. Development and evaluation of a web-based cognitive task generator for personalized cognitive training. In *Proceedings of the 3rd 2015 Workshop on ICTs for Improving Patients Rehabilitation Research Techniques - REHAB'15*. ACM Press, 2015.

[Fis07] Danyel Fisher. Hotmap: Looking at geographic attention. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1184–1191, 2007.

[Flo86] Benito E Flores. A pragmatic view of accuracy measurement in forecasting. *Omega*, 14(2):93–98, jan 1986.

[FP85] Carol E Ferrans and Marjorie J Powers. Quality of life index: Development and psychometric properties. *Advances in nursing science*, 1985.

[FTHW00] Eibe Frank, Leonard Trigg, Geoffrey Holmes, and Ian H Witten. Naive Bayes for regression. *Machine Learning*, 41(1):5–25, 2000.

[GAM+14] Theresia Gschwandtner, Wolfgang Aigner, Silvia Miksch, Johannes Gärtner, Simone Kriglstein, Margit Pohl, and Nik Suchy. Timecleanser: A visual analytics approach for data cleansing of time-oriented data. In *Proceedings of the 14th International Conference on Knowledge Technologies and Data-Driven Business*, page 18. ACM, 2014.

[GAW+11] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[GE18] Theresia Gschwandtner and Oliver Erhart. Know your enemy: Identifying quality problems of time series data. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, apr 2018.

[GFI+16] Michael Gyimesi, Gerhard Fülöp, Sarah Ivansits, Elisabeth Pochobradsky, Andreas Stoppacher, Sabine Kawalirek, and Alexander Maksimovic. *Rehabilitationsplan 2016*. Gesundheit Österreich Forschungs- und Planungs GmbH, 2016.

[GFSS00] Helena Galhardas, Daniela Florescu, Dennis Shasha, and Eric Simon. AJAX: An extensible data cleaning tool. *ACM SIGMOD Record*, 29(2):590, jun 2000.

[GGAM12] Theresia Gschwandtner, Johannes Gärtner, Wolfgang Aigner, and Silvia Miksch. A taxonomy of dirty time-oriented data. In *Lecture Notes in Computer Science*, pages 58–72. Springer Berlin Heidelberg, 2012.

[GT05] Tracy D Gunter and Nicolas P Terry. The emergence of national electronic health record architectures in the United States and Australia: Models, costs, and questions. *Journal of Medical Internet Research*, 7(1), mar 2005.

[GT15] Clinton Gormley and Zachary Tong. *ElasticSearch: The definitive guide*. O'Reilly Media, Inc., 1st edition, 2015.

[Gun98] Steve R Gunn. Support vector machines for classification and regression. *ISIS Technical Report*, 14(1):5–16, 1998.

[Gup15] Y Gupta. *Kibana essentials*. Packt Publishing, 1 edition, 2015.

[GWP14] David Gotz, Fei Wang, and Adam Perer. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. *Journal of Biomedical Informatics*, 48:148–159, apr 2014.

[HC10] Margaret A Hamburg and Francis S Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010.

[HC12] John R Harger and Patricia J Crossno. Comparison of open-source visual analytics toolkits. In Pak Chung Wong, David L. Kao, Ming C. Hao, Chaomei Chen, Robert Kosara, Mark A. Livingston, Jinah Park, and Ian Roberts, editors, *Visualization and Data Analysis 2012*. SPIE, jan 2012.

[HH13] Maria M. Hofmarcher-Holzhacker. *Das österreichische Gesundheitssystem: Akteure, Daten, Analysen*. MWV Medizinisch Wissenschaftliche Verlagsgesellschaft mbH & Co. KG, 2013.

[HMM+81] Sonja M Hunt, Stephen P McKenna, James McEwen, Jan Williams, and Evelyn Papp. The nottingham health profile: Subjective health status and medical consultations. *Social Science & Medicine. Part A: Medical Psychology & Medical Sociology*, 15(3):221–229, 1981.

[IIC+13] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, dec 2013.

[JHI+09] Diane U Jette, James Halbert, Courtney Iverson, Erin Miceli, and Palak Shah. Use of standardized outcome measures in physical therapist practice: Perceptions and applications. *Physical Therapy*, 89(2):125–135, feb 2009.

[JJB12] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: Towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405, may 2012.

[JS91] Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *IEEE Conference on Visualization 1991*, pages 284–291. IEEE, 1991.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In *Information Visualization*, pages 154–175. Springer, 2008.

[KCH+03] Won Kim, Byoung-Ju Choi, Eui-Kyeong Hong, Soo-Kyung Kim, and Doheon Lee. A taxonomy of dirty data. *Data Mining and Knowledge Discovery*, 7(1):81–99, 2003.

[KCK17] Kyungyoon Kim, John V Carlis, and Daniel F Keefe. Comparison techniques utilized in spatial 3d and 4d data visualizations: A survey and future directions. *Computers and Graphics (Pergamon)*, 2017.

[KCS+13] Friedbert Kohler, Carol Connolly, Aroha Sakaria, Kimberly Stendara, Mark Buhagiar, and Mohammad Mojaddidi. Can the ICF be used as a rehabilitation outcome measure? A study looking at the inter- and intra-rater reliability of ICF categories derived from an ADL assessment tool. *Journal of Rehabilitation Medicine*, 45(9):881–887, 2013.

[Kei02] Daniel A Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[KEP12] Christine Knaller, Alexander Eisenmann, and Daniela Pertl. *Wirksamkeit der stationären Rehabilitation für Erwachsene nach zwölf Monaten*. Gesundheit Österreich Forschungs- und Planungs GmbH, 2012.

[KGD+19] Ethan Kerzner, Sarah Goodwin, Jason Dykes, Sara Jones, and Miriah Meyer. A framework for creative visualization-opportunities workshops. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):748–758, jan 2019.

[KGHS87] Robert Allen Keith, Carl V Granger, Byron B Hamilton, and Frances S Sherwin. The functional independence measure: A new tool for rehabilitation. *Advances in Clinical Rehabilitation*, 1:6–18, 1987.

[KHA10] Nicholas Kong, Jeffrey Heer, and Maneesh Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):990–998, nov 2010.

[KLG+16] Paul Klemm, Kai Lawonn, Sylvia Glaßer, Uli Niemann, Katrin Hegenscheid, Henry Völzke, and Bernhard Preim. 3D regression heat map analysis of population study data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):81–90, 2016.

[KMH01] Theresa M Kay, Anita M Myers, and Maria PJ Huijbregts. How far have we come since 1992? A comparative survey of physiotherapists' use of outcome measures. *Physiotherapy Canada*, 53(4):268–275, 2001.

[KMS+08] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual analytics: Scope and challenges. In *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin Heidelberg, 2008.

[KOJL+14] Paul Klemm, Steffen Oeltze-Jafra, Kai Lawonn, Katrin Hegenscheid, Henry Völzke, and Bernhard Preim. Interactive visual analysis of image-centric cohort study data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1673–1682, 2014.

[KPS16] Josua Krause, Adam Perer, and Harry Stavropoulos. Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):91–100, 2016.

[Lan00] Karl Landsteiner. Zur Kenntnis der antifermentativen, lytischen und agglutinierenden Wirkungen des Blutserums und der Lymphe. *Zentralblatt für Bakteriologie*, 27:357–362, 1900.

[LBI+12] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Empirical studies in information visualization: Seven scenarios.

IEEE Transactions on Visualization and Computer Graphics, 18(9):1520–1536, 2012.

[LK14] Glyn Lewis and Helen Killaspy. Getting the measure of outcomes in clinical practice. *Advances in Psychiatric Treatment*, 20(3):165–171, may 2014.

[LPW+18] Keith R Lohse, Anupriya Pathania, Rebecca Wegman, Lara A Boyd, and Catherine E Lang. On the reporting of experimental and control therapies in stroke rehabilitation trials: A systematic review. *Archives of Physical Medicine and Rehabilitation*, 99(7):1424–1432, jul 2018.

[LSR+16] Keith R Lohse, Sydney Y Schaefer, Adam C Raikes, Lara A Boyd, and Catherine E Lang. Asking new questions with old data: The centralized open-access rehabilitation database for stroke. *Frontiers in Neurology*, 7, sep 2016.

[MA14] Silvia Miksch and Wolfgang Aigner. A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics*, 38:286–290, 2014.

[McK12] Wes McKinney. *Python for data analysis: Data wrangling with pandas, NumPy, and IPython*. O'Reilly Media, 2012.

[MF05] Heiko Müller and Johann-Christoph Freytag. *Problems, methods, and challenges in comprehensive data cleansing*. Informatik-Berichte, Institut für Informatik, Humboldt Universität zu Berlin. Humboldt-Univ. zu Berlin, 2005.

[ML17] Bhavana Maradani and Haim Levkowitz. The role of visualization in tele-rehabilitation: A case study. In *7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pages 643–648. IEEE, 2017.

[Mun09] Tamara Munzner. A nested process model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, (6):921–928, 2009.

[Mun14] Tamara Munzner. *Visualization analysis and design*. AK Peters/CRC Press, 2014.

[Nat11] National Research Council (US) Committee on a Framework for Developing a New Taxonomy of Disease. Toward precision medicine: Building a knowledge network for biomedical research and a new taxonomy of disease. 2011.

[NTC+16] Yee Sien NG, Kristin Hx Tan, Cynthia Chen, Gilmore C Senolos, Effie Chew, and Gerald Ch Koh. Predictors of acute, rehabilitation and total length of stay in acute stroke: A prospective cohort study. *Annals of the Academy of Medicine, Singapore*, 45(9):394–403, 2016.

[NZL+16] Milad Zafar Nezhad, Dongxiao Zhu, Xiangrui Li, Kai Yang, and Phillip Levy. SAFS: A deep feature selection approach for precision medicine. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, dec 2016.

[O+01] World Health Organization et al. *International classification of functioning, disability and health: ICF*. World Health Organization, 2001.

[Ö18a] Österreichischer Nationalrat. Bundesgesetz vom 9. September 1955 über die Allgemeine Sozialversicherung (Allgemeines Sozialversicherungsgesetz – ASVG), bgbl. i nr. 59/2018, 2018. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008147.

[Ö18b] Österreichischer Nationalrat. Bundesgesetz über Krankenanstalten und Kuranstalten (KAKuG), bgbl. i nr. 37/2018, 2018. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10010285.

[PBG98] Marian Petre, Alan Blackwell, and Thomas R Green. Cognitive questions in software visualization. *Software visualization: Programming as a multimedia experience*, pages 453–480, 1998.

[Pen17] Pensionsversicherungsanstalt. Jahresbericht 2017, 2017. http://www.pensionsversicherung.at/cdscontent/load?contentid=10008.657372&version=1531897802.

[PVG+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[PvW08] Johannes A Pretorius and Jarke J van Wijk. Visual inspection of multivariate graphs. *Computer Graphics Forum*, 27(3):967–974, may 2008.

[R f] R foundation. What is R? https://www.r-project.org/about.html. Accessed: 2018-01-20.

[Rah17] Thomas Rahlf. *Data visualisation with R*. Springer International Publishing, 2017.

[Rai17] Renata Raidou. *Visual analytics for digital radiotherapy: Towards a comprehensible pipeline.* TU Eindhoven, mar 2017.

[RCMA+18] Renata Raidou, Oscar Casares-Magaz, Aleksandr Amirkhanov, Vitali Moiseenko, Ludvig Paul Muren, John P. Einck, Anna Vilanova, and Meister Eduard Gröller. Bladder runner : Visual analytics for the exploration of RT-induced bladder toxicity in a cohort study. *Computer Graphics Forum*, 37(3):205–216, jun 2018.

[RF14] Raj M Ratwani and Allan Fong. Connecting the dots: Leveraging visual analytics to make sense of patient safety event reports. *Journal of the American Medical Informatics Association*, oct 2014.

[Rib] Severino Ribecca. The data visualisation catalogue. https://www.datavizcatalogue.com. Accessed: 2018-01-20.

[RMM01] Thomas N Ricciardi, Fred E Masarie, and Blackford Middleton. Clinical benchmarking enabled by the digital health record. *Studies in Health Technology and Informatics*, 84(Pt 1):675, 2001.

[RPOC18] Fateme Rajabiyazdi, Charles Perin, Lora Oehlberg, and Sheelagh Carpendale. Personal patient-generated data visualizations for diabetes patients. In *IEEE VIS 2018 Posters*, 2018.

[RSN+19] Jen Rogers, Nicholas Spina, Ashley Neese, Rachel Hess, Darrel Brodke, and Alexander Lex. Composer—visual cohort analysis of patient outcomes. *Applied Clinical Informatics*, 10(02):278–285, mar 2019.

[RYS16] Alejandro Lopez Rincon, Hiroshi Yamasaki, and Shingo Shimoda. Design of a video game for rehabilitation using motion capture, EMG analysis and virtual reality. In *International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 198–204. IEEE, 2016.

[SAKM+16] Patrick C Souverein, Victoria Abbing-Karahagopian, Elisa Martin, Consuelo Huerta, Francisco de Abajo, Hubert G Leufkens, Gianmario Candore, Yolanda Alvarez, Jim Slattery, Montserrat Miret, et al. Understanding inconsistency in the results from observational pharmacoepidemiological studies: the case of antidepressant use and risk of hip/femur fractures. *Pharmacoepidemiology and Drug Safety*, 25(Suppl 1):88–102, 2016.

[Sam59] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, jul 1959.

[SBS+15] Joel Stein, Janet Prvu Bettger, Alyse Sicklick, Robin Hedeman, Zainab Magdon-Ismail, and Lee H Schwamm. Use of a standardized assessment to predict rehabilitation care after acute stroke. *Archives of Physical Medicine and Rehabilitation*, 96(2):210, 2015.

[SCB+19] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 25(1):682–692, January 2019.

[SCGM00] John Stasko, Richard Catrambone, Mark Guzdial, and Kevin McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 53(5):663–694, nov 2000.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG)*, 11(1):92–99, 1992.

[Shn94] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages 1996. Proceedings.*, pages 336–343. IEEE, 1996.

[SNG+15] Anima Singh, Girish Nadkarni, Omri Gottesman, Stephen B Ellis, Erwin P Bottinger, and John V Guttag. Incorporating temporal EHR data in predictive models for risk stratification of renal function deterioration. *Journal of Biomedical Informatics*, 53:220, 2015.

[SO08] Emma K Stokes and Desmond O'Neill. Use of outcome measures in physiotherapy practice in Ireland from 1998 to 2003 and comparison to Canadian trends. *Physiotherapy Canada*, 60(2):109–116, apr 2008.

[Ste46] Stanley Smith Stevens. On the theory of scales of measurement. *Science*, 103(2684), 1946.

[Sto11] Emma K Stokes. *Rehabilitation outcome measures*. Churchill Livingstone, jan 2011.

[SWH14] Arvind Satyanarayan, Kanit Wongsuphasawat, and Jeffrey Heer. Declarative interaction design for data visualization. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 669–678. ACM, 2014.

[THJ17] Duong Thuy Tran, Alys Havard, and Louisa R Jorm. Data cleaning and management protocols for linked perinatal research data: A good practice example from the smoking mums (maternal use of medications and safety) study. *BMC Medical Research Methodology*, 17, 2017.

[Wad99] Derick T Wade. Outcome measurement and rehabilitation. *Clinical Rehabilitation*, 13(2):93–95, apr 1999.

[WBWK00] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119. ACM, 2000.

[WLR+18] Anna Marie Williams, Yong Liu, Kevin R Regner, Fabrice Jotterand, Pengyuan Liu, and Mingyu Liang. Artificial intelligence, physiological genomics, and precision medicine. *Physiological Genomics*, 50(4):237–243, apr 2018.

[Wor04] World Health Organization. *International statistical classification of diseases and related health problems*, volume 1. World Health Organization, 2004.

[Wor11] World Bank World Health Organization. World report on disability, 2011.

[WT04] Pak Chung Wong and J Thomas. Visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21, sep 2004.

[Yao99] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

[YKS07] Ji Soo Yi, Youn Ah Kang, and John Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, nov 2007.

[ZGP15] Zhiyuan Zhang, David Gotz, and Adam Perer. Iterative cohort analysis and exploration. *Information Visualization*, 14(4):289–307, 2015.