Data Mining For Outcome Analysis In Hip Arthroplasty

Knut T. Hufthammer

Department of Information Science and Media Studies
at the University of Bergen, Norway

2021

Supervisor: Prof. Ankica Babic

June 15, 2021

Acknowledgements

I want to thank my supervisor, Dr. Ankica Babic, whose guidance, support, and advice have been invaluable throughout this journey. You have kept me motivated and focused during these difficult pandemic times. The dedication and enthusiasm you show towards your work is truly admirable.

A special thanks to Dr. Peter Ellison for providing the sample data for this research. His advice and feedback has contributed greatly to the completion of this work.

To my fellow master students that I have collaborated with for the past year. I am thankful for your support and all the helpful advice you have provided.

Thanks to the staff of the Norwegian Arthroplasty Register for taking the time to meet with us and provide us with valuable insights into their work.

Finally, I wish to express my deepest gratitude to my beloved family for their unconditional love and support. I am forever grateful for the values they have instilled in me. I dedicate this to them.

Abstract

Today, the Norwegian Arthroplasty Register (NAR) works in a traditional way with statisticians who help prepare, conduct, and report on data analyses. Doctors and biomedical engineers are often turning to the registry for the purpose of monitoring and answering their research questions. Technology-based solutions may help facilitate and streamline the above process, enabling users to interact and utilize this national database in a more accessible manner.

Using Design Science Research (DSR), we identified data mining tasks and set out to deliver a Web-based system to streamline data mining on hip arthroplasty data. In a collaborative effort between back-end and front-end developers, we implemented the prototype as a Web-based application and modeled the data mining methods after the Knowledge Discovery in Databases (KDD) process.

The contribution of this thesis is a fully functional prototype for exploring arthroplasty data and assessing hip implant performance. Among the implemented methods are Cox Regression, Kaplan-Meier analysis, and Logistic Regression.

Based on the expert evaluation, we consider the novelty of the artifact to be twofold. First, we bridge the gap between humans and statistical models by allowing end-users to assess the quality of hip implants in a direct and more tailored manner. Second, we may extend the system to include additional methods to meet diverse user needs.

Future work should further involve domain experts to suggest additional methods and carry out a comprehensive evaluation in a real clinical setting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For more than half a decade, hip arthroplasty has helped relieve pain and restore normal hip function to the hip joint making it one of the most successful and widely performed surgeries today. The demand for Total Hip Arthroplasty (THA) is increasing worldwide, with more than a million surgeries performed annually (Pivec et al., 2012, p. 1768). Improvements to surgical techniques, new bearing surfaces, and implants have lowered revision rates and reduced premature failure of hip implants. Hip replacement surgery owes part of its success to national joint registries established in the 70s and early 80s to monitor and report on implant survivorship (Lübbeke et al., 2018; Pivec et al., 2012).

The role of national registries is to collect and survey large amounts of data for differences in outcome. If such a difference is determined, the registries will conduct further analysis to identify which factors influence the outcome (Graves, 2010). Sweden established the first such registry in 1979, and other Scandinavian countries followed shortly after in the early 80s. Today, national registries are widespread and play a crucial in identifying the best surgical practices and guidelines that lead to improved clinical outcomes of joint replacement surgery (Lübbeke et al., 2018; Pivec et al., 2012; Graves, 2010). For example, research by the NAR led to the identification of an underperforming implant widely used for THA in Norway. The study observed that survival outcomes of the implant worsened from one period to another and found that the deterioration coincided with changes in surgical techniques and implant material (Hallan et al., 2012).

Increased demand for THA and the recently enforced EU regulation calls for surveillance of new implants that are introduced to the market (The European Commission, 2017; Lübbeke et al., 2018). In turn, this require us to consult and review the data more often. The registries are providing a great source of data and knowledge, but are still working in a traditional way where annual and other reports are produced with the help of statisticians. This research wants to design solutions that would enable interactive and streamlined data analysis for users, which physicians, researchers, and other health-care management staff can utilize. Currently, there are no systems offering online data analysis on arthroplasty data in Norway. This has motivated the research presented in this thesis that looked into possibilities of data mining and implementation of a system which could help users perform automatic analysis online. The artifact produced by this thesis enables users to carry out procedures for assessing risk and predict the longevity of implants. Design Science was used as the research framework that provided guidelines to design solutions for relevant stakeholders in the arthroplasty domain.

Following are the research questions that were formulated to keep the research relevant

and purposeful.

1.1 Research Questions

RQ 1: What are the qualities and characteristics of an outcome analysis tool for THA?

RQ 2: What data mining methods are useful for outcome analysis in THA?

RQ 3: Can KDD lower the barrier of entry and allow medical staff to analyze hip arthroplasty data without the need for a statistical background?

1.2 Collaborative Aspect

This thesis benefits from contemporary work from three other collaborators that each provided a distinct contribution to the project. The outcome of the collaboration is a prototypical data exploration and outcome analysis tool for a national joint registry. Two students have focused on data mining, and the latter two on Human-Computer Interaction (HCI) and data visualization. The project further makes a distinction between data mining designated for hip and knee arthroplasty data. The students working on data mining have maintained a close collaboration and supplied the two other students with data for visualization and data mining methods for their prototype. Although the focus of this thesis is primarily on applying data mining methods to hip arthroplsty data, we produced a minimal front-end application to validate and showcase our methods.

1.3 Thesis Outline

Chapter 2: Theory presents related work, THA, the national arthroplasty register, and the theoretical framework for the practical work of this thesis.

Chapter 3: Methodology and Methods presents the methodologies and methods used in this work.

Chapter 4: Establishing Requirements describes the functional and non-functional requirements of the system, and the technologies used for the prototype.

Chapter 5: Data Material describes the data sample provided for this research.

Chapter 6: Prototype Development presents the system architecture and workflow, and a detailed outline of the four development iterations of this project.

Chapter 7: Artifact presents the resulting artifact produced by this thesis.

Chapter 8: Evaluation presents feedback from the evaluation with IT and domain experts. We also present results from the System Usability Scale questionnaire.

Chapter 9: Discussion provides a discussion of the prototype development, data mining tasks, artifact evaluation, and limitations of this research work.

Chapter 10: Conclusion and Future Work concludes and summarizes the work. Directions for future work are outlined at the end.

Chapter 2

Theory

2.1 Related work

In this section, we present related work and provide background material of the NAR. We also provide a theoretical framework for the practical work of this thesis. Specifically, we provide a short introduction to machine learning and survival analysis.

2.1.1 Mining for individual patient outcome prediction in hip arthroplasty registry data

Kristoffersen (2019) explored the applicability of applying machine learning techniques on a hip arthroplasty dataset from the NAR. Using a data mining based approach, Kristoffersen investigated the efficacy of using unsupervised and supervised learning to predict individual patient outcomes. The author conducted an initial data analysis phase to identify dependent and potential independent variables in the dataset. Cluster analysis was used to identify similarities and distinctions between different patient groups. The analysis found that *age* was more or less similarly distributed across clusters, and neither males nor females were associated with worse survival outcomes. A similar proportion of men and females was found for revision rates, indicating no disproportionate distributions for either sex in the dataset. The survival length of the prosthetic device and a dichotomous indicator for revision surgery was used as target labels. The survival length of the prosthetic device was modeled as a binary outcome feature, partitioning examples into two target classes. (1) Those with a survival length under eight years and (2) those at eight years or more. Both target classes had approximately the same level of support - 54% of the sample required revision within eight years, whereas the remaining 46% lasted over eight years. Excluded from the analysis, was deceased patients and patients not actively monitored by the register. Kristoffersen trained the model on features known at the time of primary surgery. Examples of such features are patient information, device materials, and reason for indication (Kristoffersen, 2019, p. 65-67).

Three different classifiers were tested - Logistic Regression, Random Forest and a Multi-layer Perceptron Classifier (MLP). Among these, the MLP performed best, resulting in an outcome that mirrors the real empirical outcome. Approximately 54% of the examples were classified with a survival duration below eight years and 47% above eight years. However, performance measures for the classifier were less impressive. The confusion matrix shows a

False Positive Rate (FPR) of 18%. In other words, nearly 1/5 of all classified examples was wrongly assigned to false positive outcomes. The False Negative Rate (FNR) was measured at 15%, indicating that the classifier is less likely to falsely assign a negative label to an example that belongs to the positive class (Kristoffersen, 2019, pp. 66-68). In the best case, Kristoffersen (2019) obtained an area under the curve score of 0.75 for the Multi-layer perceptron classifier. The other models performed insufficient for practical use.

Kristoffersen suggests that more variance in the dataset and more details about specific prosthesis can boost performance further (Kristoffersen, 2019, p. 78). As future work, he encourages the idea of combining the models and methods into a "full software solution" for use in a real-world environment to aid decision making in hip arthroplasty surgery (Kristoffersen, 2019, p. 79).

2.1.2 HALE, the Hip Arthroplasty Longevity Estimation system

Most hip prostheses are successful short-term — approximately 90% of all hip implants last over ten years. However, complications leading to revision surgery can arise. Typically, revisions occur due to loosening of prosthesis'. Other indications for revision are bacterial infection, wear, and fracture (Hallan, 2007). To better understand why and when prostheses fail, Longberg (2018) developed the Hip Arthroplasty Longevity Estimation system (HALE). HALE is a fully-working prototype aimed at physicians for the purpose of predicting hip prosthesis longevity in patients. The project seeks to investigate the efficacy of using machine learning to predict longevity pre-surgically in order to find the most suitable and effective installment. A distinctive feature of HALE is the inclusion of a UI to lessen the entry barrier and involve medical practitioners without the need for a background in statistics or informatics (Longberg, 2018, p. 1).

Longberg pursued two different approaches of predictive modeling - multiple regression analysis and optimized classification and decision tree regression (CART). For the user-centered part of HALE, multiple linear regression (MLR) was chosen, since it offered better performance than decision tree regression (Longberg, 2018, p. 68).

The models were validated using SPSS - a well-known, validated statistical analysis tool from IBM (SPSS Inc., 2021). The MLR model showed comparable performance to a similarly constructed linear regression procedure from SPSS both in terms of accuracy and performance (Longberg, 2018, p. 72).

The usability of the system was assessed using semi-structured interviews, heuristic evaluation, and the SUS method. Feedback from the evaluation suggests users found the system easy to explore and appealing in terms of functionality. Others perceived it as being a bit too technical (Longberg, 2018, p. 73).

Statistical evaluation found that predicted longevity outcomes were 'reasonably good' and that that the machine learning component was manageable to use by novice users. Similarly to Kristoffersen (2019), Longberg recognizes the dataset as a limiting factor of his research. In conclusion, Longberg argues that the performance of the models can be further improved using a larger dataset with additional clinical variables.

2.1.3 Multiple Imputation in Predictive Modeling of Arthroplasty Database

In this thesis, Berge (2019) explores the possibility of using data mining techniques to forecast individual patient outcomes in THA. DSR was used as the research methodology and KDD was used for the data mining process. Haukeland University Hospital provided Berge with two small datasets of failed cases of THA. Berge's approach to data mining is two-fold. First, he investigates the completeness of the data by analyzing it for missing data. Then, an attempt is made to fill in the missing values by means of multiple imputation – a technique for replacing missing values in data. The second part of the paper deals with the development of a web-based prediction tool for THA patients. Berge used the programming language R for both aspects of his work (R Development Core Team, 2004).

The first dataset was an unstructured and distorted spreadsheet with tables that appeared to be out of place and without context. Figures without explanations were scattered around, and parts of the spreadsheet were formatted with colors that had no clear interpretation. Due to the difficulties of relating these tables and figures to the main table, only the main table was extracted and exported to a more friendly format for data analysis. The result was a comma-separated values (CSV) file with 27 observations and 47 variables. Berge used R to analyze the data completeness and found that roughly 1/5 of the values were missing. Variables relating to the wear of a prosthesis, osteolysis, and trace metals found in the blood were the most frequently missing variables in the dataset. The second dataset was in much better condition with less missing data, containing more observations, and the number of variables reduced to half of the original dataset (Berge, 2019, pp. 36-39). An interesting aspect of Berge's work is the visualizations that he made of missing data. For example, a "missingness pattern" plot and a bar chart showing the proportion of missing values for each variable are featured. These visualizations offer an easy and straightforward interpretation of the dataset's completeness.

Multivariate Imputation by Chained Equations (MICE), a software package in R, was used for the imputation phase of his work (van Buuren, 2021). Although imputation was performed on both datasets, the first dataset was primarily used for experimentation with methods and parameter tuning (Berge, 2019, p. 47). For the second dataset, Berge tuned parameters of the methods based on guidelines from the literature. The results of the imputation were assessed using density plots, scatter plots, and convergence plots. The plots show that the imputed data holds a similar shape to the original data, indicating that the imputation was effective, although some deviation was present. The prototypical prediction tool was made with R Shiny - a software package in R that allow for the creation of interactive web applications (RStudio, Inc, 2021). Berge's tool features a linear regression component to perform simple linear regression. The tool allows the user to input an independent and dependent variable and be presented with detailed results from running the analysis. After performing the regression analysis, users are presented with a regression plot and a detailed summary, including p-values, r-statistic, r^2 and other statistical metrics. With regards to the imputed datasets, the predictive accuracy of linear regression was somewhat ambiguous. In some cases, improvements were observed and in other cases not (Berge, 2019, pp. 74-77).

2.2 Total Hip Arthroplasty (THA)

The hip joint is a ball-and-socket joint located between the femur and acetabulum of the pelvis. Its primary function is to support the weight of the body during static posture or movement. The upper end of the femur is the femoral head (ball) which inserts into the acetabulum (socket) of the pelvis. The ball and socket are coated with a layer of thin tissue, called articular cartilage, enabling them to move smoothly. The hip joint itself is bond together with ligaments (tissue) and coated with a tissue called synovial membrane that produces a lubricating fluid within the cartilage to avoid friction during hip movement (Foran, 2015).

The most common cause of chronic hip pain is arthritis, but fracture, diseases, and dislocation due to injury can also cause pain. For example, in children, medical conditions may disrupt normal hip growth and lead to arthritis (Foran, 2015). Damage to the hip can be painful and restrict the mobility of the hip - limiting one's ability to perform daily activities. In some cases, getting in and out of bed can be a strenuous and painful task. Depending on the severity of the damage, medications and lifestyle changes may be sufficient to relieve pain and hasten the recovery process. In other cases, replacing the injured parts with artificially constructed components, known as prostheses, may be required. Such artificial replacement of a hip joint is known as hip replacement surgery (hip arthroplasty) and is a standard procedure, commonly performed on elderly affected by osteoarthritis.

Today, hip arthroplasty is performed successfully across all age groups. We can group the practice into two types of procedures - THA and hemiarthroplasty. THA replaces both the femoral head and acetabulum, while hemiarthroplasty replaces only the femoral head (Foran, 2015). In 2018, in Norway, the average age of patients receiving surgery was 67 for men and 68.9 for women. The majority of patients are women, and the primary cause of indication is osteoarthritis at 79% (on Arthroplasty and Fractures, 2019, p. 9). For younger patients, the primary indications appear to be paediatric hip diseases (33%), systematic inflammatory disease (23%), and avascular necrosis (21%). In younger patients, osteoarthritis accounts for only 4% out of all other indications (Halvorsen et al., 2019).

2.3 The Norwegian Arthroplasty Register (NAR)

Since its inception in 1987, the NAR has recorded 233 142 hip arthroplasties with a steady increase of surgeries each year. In 2018, a total of 9 553 primary surgeries were performed, along with 1 422 revisions. The latter amounts to a revision rate of 12.8% which is the lowest revision rate in the history of the register. All interventions are regularly reported to the register. Therefore, prior to surgery, surgeons are required to fill out a standardized form concerning details about the planned surgery (on Arthroplasty and Fractures, 2019, p. 9). The information collected by the register includes, but is not limited to, patient demographics, indication for THA (diagnosis), surgical procedure, implant and revision information (Dale et al., 2011, pp. 647-648).

2.3.1 Data validity

The validity of a register are typically measured across four major axis': (1) coverage, (2) registration completeness of patients/surgeries, (3) registration completeness of recorded variables, and (4) accuracy of the registered variables (Varnum et al., 2019, p. 338). The coverage is the proportion of departments reporting to the national registers out of the total number of departments performing arthroplasty. Coverage is generally high in Nordic countries because the authorities reimburse the orthopedic departments for reporting to the registers (Varnum et al., 2019, p. 338). Additionally, annual reports are provided to participating departments with results from each department, which further helps incentivize reporting of operations to the registers (Furnes and Havelin, 2002, p. 40). The completeness of registration is measure of how well the register reflects the data reported to the national patient registers (Furnes and Havelin, 2002, p. 40)(Varnum et al., 2019, p. 338). In Norway, the completeness of registration is quite high. From 2008-2012, the completeness of registration was 96.6% for THA and 95.3% for primary knee surgeries (Pedersen and Fenstad, 2016, p. 19). Another aspect concerning the validity of registers is the registration completeness of variables. This refers to the proportion of variables registered by the surgeons out of the total number of variables recorded by the register. The final axis concerns the accuracy of the information (variables) provided by the surgeons. The accuracy is the probability that the variables reported to the register are correct. Since the data is used to assess the quality of prostheses, the information must be correct and give an accurate description of the surgery performed by the surgeon. In Denmark, the accuracy of variables is evaluated in annual reports. The accuracy of variables in the NAR have also been studied (Pedersen and Fenstad, 2016, pp. 18-20).

Arthursson et al. (2005) assessed the quality and validity of the data recorded by the NAR by comparing it to data recorded by the Norwegian Patient Register (NPR) and a local hospital. They found the register a valid, reliable, and an excellent source of information for clinical data on THA. The study reviewed 5 134 THAs and revisions performed at a single hospital between 1987 and 2003. Kaplan-Meier survival curves were compared across the two registers to evaluate the possibility of missing data. Out of the 5 134 operations, only 19 (0.4%) were missing from the NAR (Arthursson et al., 2005, p. 823). In comparison, 47 operations or 3.4% were missing from the NPR. In 56 cases (1.1%), the date of surgery was misreported in the NAR. 85% of these errors were tracked back to the surgeon. The remaining 15% occurred due to typing errors at the NAR (Arthursson et al., 2005, p. 825).

The NAR is considered a high-quality and successful arthroplasty register. In fact, all Nordic countries maintain registers with high standards and are often considered the "ideal" for other countries to model their register upon. This success is due to a collaborative effort between the Nordic countries to standardize data collection, variables, and statistical methods. The collaboration was established in 2007 as the Nordic Arthroplasty Register Association (NARA), and their aim is to improve the quality of treatment and research of joint replacement surgery (Pedersen and Fenstad, 2016). The idea is, that by agreeing upon a common dataset and statistical methods, research done on one register is more likely to be applicable and comparable to another. In turn, that should lead to better research quality and eventually, improved quality of treatment (Pedersen and Fenstad, 2016).

2.3.2 Scientific landscape

Citation and co-author analysis was conducted to map out the scientific landscape of publications related to the NAR and the NARA. The purpose was to identify key figures and review the influence of NAR and NARA. Analyses were performed using VOSviewer, a software package for creating and visualizing bibliometric networks (van Eck and Waltman, 2020). Bibliographic data was obtained from Web of Science by doing an advanced search for publications referencing the Norwegian Arthroplasty Register and Nordic Arthroplasty Register Association, as well as their respective acronyms. The search was further restricted to items including the word "hip" at least once and papers published before 1987 were excluded. Due to exportation limitations with Web of Science, citation analysis was restricted to items published in the Web of Science Core Collections. Data for the co-author analysis was searched across all available databases at the Web of Science.

Two datasets were exported from Web of Science and imported into VOSViewer. The unit of analysis was 'Author' for both the citation and co-author analysis. This means that a node in the network represents an author and that the size of that node is determined by the number of documents published by that author. In the citation analysis, the relatedness of nodes is determined by the number of times they cite each other. Thus, authors who tend to cite each other will have a stronger link, and the edge between them will appear thicker. In the co-author analysis, the relatedness of nodes is determined based on their number of co-authored documents. Thus, authors who tend to appear in the same documents will have a stronger link and the edge between them will appear thicker. The color of the nodes indicates which cluster they belong to. Authors with less than five documents were excluded from the analysis to avoid cluttered visualizations and many outliers. For the same reason, the minimum link strength was set to 5. Therefore, there may be authors who have been omitted from the analysis, and some authors may appear to be disconnected from each other even if there is a connection between them. It is also possible, but unlikely that the sample from the Web of Science is not representative of the actual scientific landscape.

Figure 2.1 shows the result of the citation analysis. The largest nodes in the network appear to be Ove Furnes and Anne M. Fenstad, with 27 and 21 documents. Alma B. Pedersen is the third-largest node with 19 documents. Leif I. Havelin comes fourth with 17 documents. Johan Karrholm and Søren Overgaard follow closely with 16 documents each. Karrholm and Overgaard serve as the directors of the Hip Arthroplasty Register in Sweden and Denmark, respectively (Höftprotesregistret, nd; Register, nd). Interestingly, all of these authors are connected with a link strength of 33 or greater. The authors have Scandinavian names and apart from a few exceptions, most of them have a link to each other which shows the extent of collaboration between the Nordic countries.

Furthermore, there appears to be a strong link between the current and former director of the NAR, Ove Furnes, and Leif I. Havelin. Furnes is Havelin's strongest link with a link strength of 42. Havelin founded the register together with Lars B. Engesæter and served as the director from 1987 until 2002. Co-founder Engesæter is also one of Havelin's strongest links. Havelin has since worked as a chief physician at the Department of Orthopaedics, Haukeland University Hospital (Tidsskriftet, nd). In 2019, Havelin and Engesæter received the Knight 1st Class award from the Order of St. Olav for their contributions to orthopedics (Kongehuset, 2018). As of June 2021, Furnes is the acting director of NAR (on Arthroplasty and Fractures, nd).

Furnes, together with my supervisor Dr. Ankica Babic, are the driving forces behind the collaboration between the Department of Information and Media Studies at the University of Bergen and the register, which has arranged for the execution of several master's theses in recent years. There has also been produced research with Dr. Peter Ellison and researcher and engineer Paul Johan Høl from the Biomedical Engineering Laboratory at Haukeland University Hospital. Table 2.1 provides an overview of master theses´ executed under the supervision of Dr. Ankica Babic and the collaboration with the register.

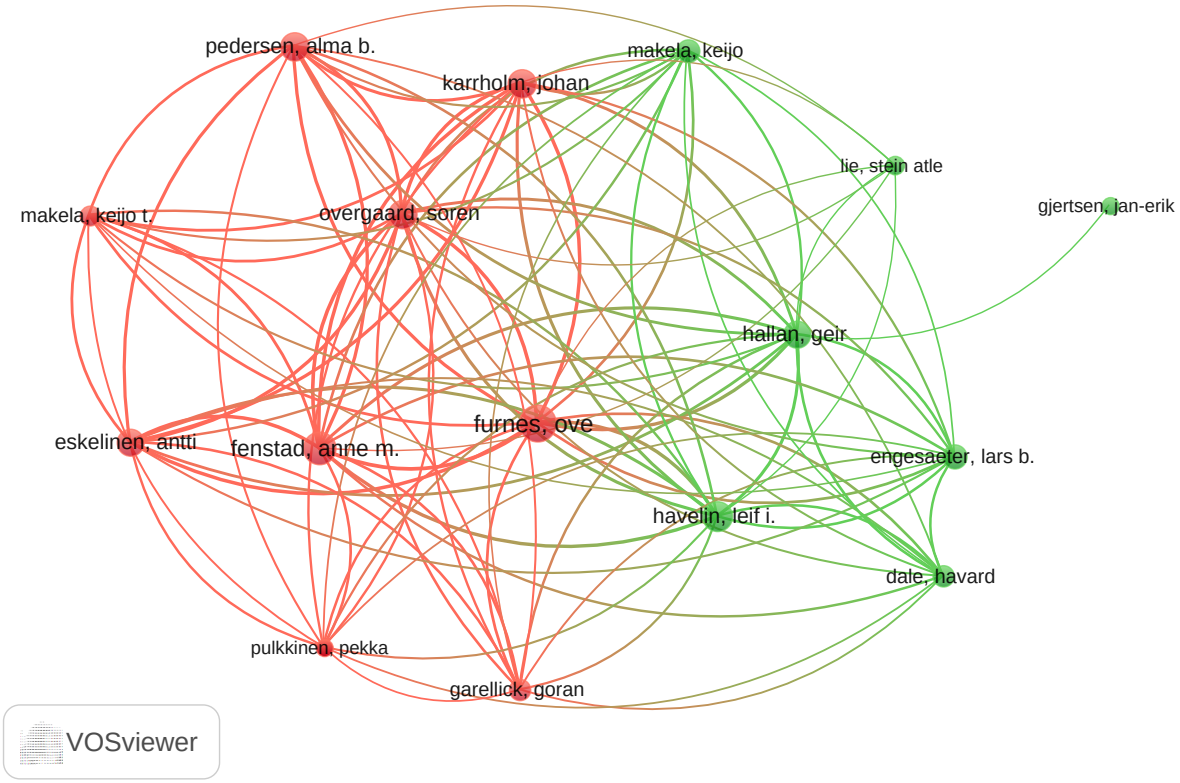


Figure 2.1: Network visualization of the citation analysis of publications related to the NAR.

Figure 2.2 shows the result of the co-author analysis. The largest node in the network is Ove Furnes with a total of 74 documents. The second largest is Leif I. Havelin with a total of 57 documents. Both of these nodes appear in the same cluster and are considerably larger than other nodes in the network. There are three clusters in total, but one of them seems to be more or less disconnected from the rest of the network. The other two clusters seem to be ordered according to geographic origin. The green cluster consists exclusively of Norwegian authors, while the red cluster contains a mixture of Scandinavian nationalities. This may indicate that Norwegian authors work closely together and are often involved in work together. There is at least one link between Furnes and every single author in the red cluster. The same is true for Havelin. Furnes and Havelin's extensive network probably reflect their leadership roles. Statistician Anne-Marie Fenstad employed by NAR is clustered together in the red cluster despite her Norwegian nationality. She appears as a highly connected node with many ties to authors from other countries in Scandinavia.

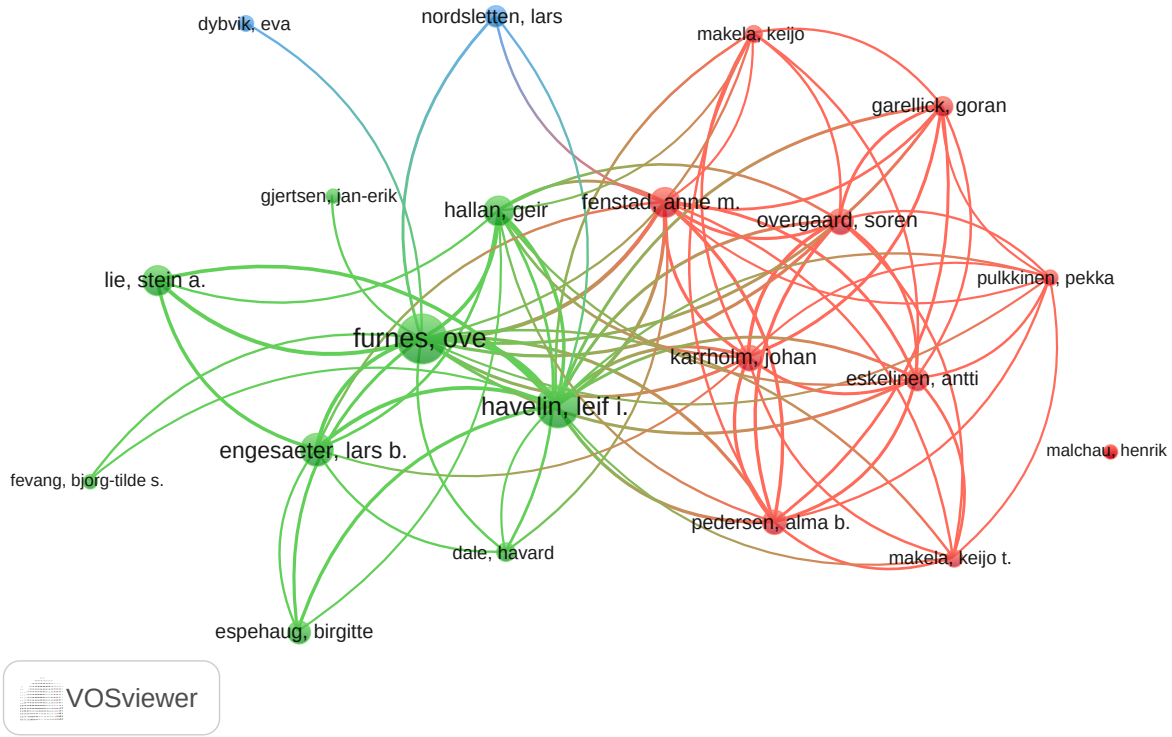


Figure 2.2: Network visualization of the co-authorship analysis of publications related to the NAR.

Overall, the network of authors referencing the NAR and NARA are mostly authors of Scandinavian origin. Although not depicted in Figure 2.2 and 2.1 , there are also articles by authors outside of Scandinavia such as England, Japan, Australia, Netherlands, and the USA. These authors did not meet the threshold for inclusion. Furthermore, apart from a few outliers, most nodes seem to be highly connected, both within and across other clusters. The most influential nodes in the network seem to be the registries' leaders, both current and past. They have ties to most other nodes in the network. In addition, these authors have considerably more publications attributed to them than other nodes in the network. In the co-author analysis, there is a cluster that consists exclusively of Norwegian authors while the other large cluster contains a mixture of Scandinavian authors.

	Paper	Year
Berntsen, Eirik	Information system for postmarket surveillance of total joint prostheses	2014
Ertkjern, Ørjan	Postmarket Surveillance of Orthopaedic Implants using Web-technologies	2015
Åserød, Hanne	Mobile Design For Adverse Event Reporting And Pharmacovigilance	2017
Carlsen, Tor Aimar	Designing an e-learning platform for patients undergoing hip replacement surgerys	2018
Krumsvik, Ole Andreas	A Self-Reporting Tool to Reduce the Occurrence of Postoperative Adverse Events After Total Hip Arthroplasty	2017
Longberg, Per-Niklas	HALE, the Hip Arthroplasty Longevity Estimation system	2018
Berge, Øyvind Svenning	Multiple Imputation in Predictive Modeling of Arthroplasty Database	2019
Kristoffersen, Yngve	Mining for individual patient outcome prediction in hip arthroplasty registry data	2019
Iden, Andreas	Data Mining Approach to Modelling of Outcomes in Total Knee Arthroplasty	2020
Blom Stolt-Nielsen, Sunniva	Design Driven Development of a Web-Enabled System for Data Mining in Arthroplasty Registry	2021
Farsund Solheim, Arle	Arthroplasty Data Visualization	2021
Ånneland, Sølve	Web-based Data Mining Tool for Total Knee Arthroplasty	2021
T. Hufthammer, Knut.	Data Mining For Outcome Analysis In Hip Arthroplasty	2021

Table 2.1: Master theses executed under the supervision of Assoc. Prof. Ankica Babic and the collaboration with the register.

2.4 Machine learning

Machine learning is the practice of applying algorithms to build statistical models that can aid in decision making in a specific application area. These statistical models attempts to learn a mathematical function from a *dataset* or collection of past observations to make inferences about future observations. Typically, this dataset is divided into two separate ones for which the larger portion is used to *train* the model whereas the remaining part is used to assess the quality of the model (validation) (Burkov, 2019, p. 3). The performance or validity of a model is often evaluated in terms of its accuracy, precision, recall rate, and ROC curve (Burkov, 2019, p. 65).

Accuracy refers to the proportion of correct predictions in the set of all predicted outcomes. Precision is the proportion of true values to the total number of predicted positive values. Recall is the ratio of correctly predicted positive values to the overall number of positive instances in the training set (Burkov, 2019, p. 66-67). Ideally, you would want to have high precision and high recall, but this is often difficult to achieve. Optimizing one metric is likely to affect others negatively. Therefore, which metric to optimize for is usually chosen on a case-by-case basis (Burkov, 2019, p. 66). For instance, in spam detection, misreporting a legitimate email can be costly. However, misreporting a small amount of spam is unlikely to have negative consequence. In such case, it may be acceptable to sacrifice recall in favor of precision gains. However, in other cases, such as medical diagnosis, one must maintain a fine balance between precision and recall. On the one hand, it is required that the classifier is precise. Moreover, it is important to exhaust every possibility to identify the right diagnosis, since missing one could be costly. In such cases, visualization techniques like the ROC curve can come in handy. ROC shows how the relationship between recall and precision fluctuates in accordance with changes in the threshold for identifying a positive outcome in the model (Burkov, 2019, pp. 67-68).

We tend to differ between four types of machine learning - supervised, semi-supervised, unsupervised and reinforcement learning (Burkov, 2019, pp. 3-4). In supervised learning, we use a dataset of *labeled examples* to learn a function that can predict the outcomes of future observations. The objective is to model the relationship between a set of *independent variables* and a *dependent variable*. The independent variables are sometimes referred to as *features* or *predictors* and are used to predict the value of the independent variable. Dependent variables are either from a continuous or discrete distribution. When the dependent variable is continuous, we use regression algorithms to model the relationship (Burkov, 2019, pp. 3-25). Examples of such algorithms are Simple Linear Regression and Multiple Linear Regression (Géron, 2019, pp. 8-9). The goal is to fit a *regression line* that best fits the observed data points in our dataset. In linear regression, a common way of estimating the regression line is using the least-squares method. The least-squares method fits the regression line by minimizing the squared sum of distances between the observed data points and the line we are trying to fit. The distances between the observed points and the line is known as *residuals*. Thus, more succinctly, we say we want to minimize the squared sum of the residuals (Géron, 2019, p. 113).

For discrete variables, we use a class of algorithms known as classification algorithms. Examples are Logistic Regression and Naïve Bayes Classifier. The former is not inherently a classification algorithm, but is commonly used in conjunction with a decision boundary to form a binary classifier. Rather than fitting a straight line through the observed data points

like in linear regression, Logistic Regression fits a *S-shaped curve* using the *logistic function* - a type of sigmoid function. A decision boundary is then drawn on the line, effectively functioning as a cut-off point to partition examples into one of two classes (Géron, 2019, 85-107). In classification problems, the dependent variable is analogous to a *target variable*. The target variable can take on a set of outcomes known as classes. A classification task with only two target classes is known as *binary classification*. Likewise, classification with more than two targets, is known as multi-class classification. We refer to models trained on a set of examples as *classifiers*. These classifiers have learned the function that allows them to assign (or predict) *labels* of future observations. The predicted label must correspond to one of the predefined target classes. For example, in a binary outcome problem, these classes may be unmarried/married or dead/alive. Binary and multi-class classifiers designate exactly one label per example. Classification problems requiring more than one label per example should use a multi-label classification algorithm. Examples of such algorithms are Random Forest and MLP (Géron, 2019, pp. 85-107). Scikit-learn offers a comprehensive documentation with a detailed overview of algorithms for binary, multi-class, and multi-label classification (Pedregosa et al., 2011).

In unsupervised learning, there are no predefined target labels to predict. Instead, we aim to partition examples into clusters or groups based on their similarity to each other. This type of learning is suitable for exploring unknown data and problems where the outcome is not yet known. Typical use cases of unsupervised learning are clustering, dimensionality reduction, and outlier detection (Burkov, 2019, p. 8).

In clustering, we aim to categorize examples based on shared attributes to identify qualities that separate one group from another. One of the most popular clustering algorithms is k-means - a very efficient technique to group data into k number of clusters (Géron, 2019, pp. 236-241). The main challenge with clustering is choosing how to sort the data and how many clusters to group the data into. Depending on the configuration, different angles or perspectives can emerge from the data. If we have too few clusters, we fail to capture the underlying structure of the data, and no interesting patterns emerge. In that case, the model is underfitted to the data. On the other hand, if we allow too many clusters, we may risk corresponding the model too closely to the underlying dataset. In that case, the model is overfitted to the data, and the clusters become difficult to interpret (Géron, 2019, pp. 27-29). It seems to be an art to interpret results of cluster analysis, it usually demands expertise and some experience with the field of research.

With dimensionality reduction, the objective is to transform each feature vector to a lower dimensionality or a simplified representation. It is the process of removing redundant or highly correlated features and the overall noise in the data. Dimensionality reduction is often used to project high dimensional spaces onto a lower dimensionality that is more suitable for visualization. With a simplified representation, we can take advantage of visualizations to uncover insightful patterns or apply other machine learning techniques such as regression analysis for further analysis (Burkov, 2019, p. 130). One of the traditional approaches to dimensionality reduction is Principal Component Analysis (PCA). This technique was recently applied by Iden in his work with Total Knee Arthroplasty (TKA) data (Iden, 2020). The application of Principal Component Analysis (PCA) showed potential for descriptive modeling and was advocated for its usefulness in scenarios with a large number of variables (Iden, 2020, pp. 56-57).

In outlier detection, we attempt to detect examples within the dataset that differ from the

typical example in the dataset. Such analysis can be very important in assessing the significance of our findings since outliers can skew our results in a particular direction. Therefore, outlier detection is commonly used as a preprocessing step to remove anomalies in datasets. One-class classification learning algorithms is typically used for detecting outliers (Burkov, 2019, p. 90).

Algorithms that work with both unlabeled and labeled data are known as semi-supervised learning. The few labeled examples are used to train a supervised model. Then, the remaining examples are used with an unsupervised algorithm to improve the performance of the supervised model. These algorithms combine unsupervised and supervised techniques (Géron, 2019, p. 13).

2.5 Survival Analysis

The following section gives a short introduction to survival analysis and its terminology. A complete introduction to survival analysis falls out of the scope of this text - only concepts and methods relevant to this thesis are presented. More detailed descriptions can be found in the literature [1]. The formulas and mathematical notation that follow have been adopted from Kleinbaum and Klein, 2012.

Survival analysis is a statistical discipline originating from the medical community in the seventeenth century. Initially used to study lifetimes in demographic groups, survival analysis is today an integrated component of theoretical statistics. It has expanded to other fields such as engineering, behavioral, and actuarial sciences (Andersen and Keiding, 2005).

In survival analysis, the object of study is *time-to-event data*, and the response or outcome variable is *time until an event occurs* [2]. Time begins with the *follow-up* or 'birth' event of an individual and elapses until a *death event* (death, relapse, failure) occurs. Individuals who are lost to follow-up before the observational period ends, are said to be *censored*. Individuals becomes *lost to follow-up* due to external circumstances preventing us from recording their lifetime history (Kleinbaum and Klein, 2012, pp. 4-6). For example, the individual may go missing, withdraw from the study or move abroad. Furthermore, individuals whose survival time is greater than the observed survival time are said to be *right-censored*. For right-censored individuals, we only know their *observed lifetime*, not their actual lifetime. In other words, we know that the individual survived up to some point in time, but the exact duration remains unknown. One of the core assumptions in survival analysis is that censorship is regarded as *non-informative*. That is, we regard censored individuals as having the same survival prospect as their uncensored counterparts (Kleinbaum and Klein, 2012, p. 5-8).

The two primary tools for modeling lifetime data in survival analysis are the survival and hazard function (Kleinbaum and Klein, 2012, p. 8). The *survival function* $S(t)$ defines the probability of survival past time T and can be calculated by subtracting the Cumulative Distribution Function (CDF) $F(t)$ from one. Formally, we have:

$$S(t) = 1 - F(t) = Pr(T > t) \tag{2.1}$$

Here, T is a non-negative random number denoting the time of death. The lowercase t

[1] See Kleinbaum and Klein (2012) for an introductory text to survival analysis

[2] The outcome variable is time until an event occurs

denotes a specific value for T. Thus, $Pr(T > t)$ is the probability of survival past time t, i.e. the probability that the time of death T is greater than t.

The survival function has the following properties:

- $S(t)$ is monotonically decreasing as time t increases, i.e., $S(1) \geq S(2), S(3), ..., S(n-1) \geq S(n)$
- $S(0) = 1$. The probability of survival past time 0 is 1.
- $S(t)$ tends to 0 as $t \to \infty$. The probability of survival will tend towards 0 as t approaches ∞ (Kleinbaum and Klein, 2012, pp. 8-9).

Furthermore, by computing the derivative of the survival function, we obtain a *Probability Density Function (PDF)*:

$$f(t) = \frac{dS(t)}{dt} \tag{2.2}$$

Likewise, integrating over the PDF from t to ∞ gives us the *survival function*:

$$S(t) = \int_{t}^{\infty} f(t)dt \tag{2.3}$$

Note that to obtain the survival function, we integrate from $t \to \infty$ over the PDF. Conversely, to get the CDF we integrate from $0 \to t$ over the PDF. Since the total probability must equal one, we have that one minus the CDF of t is the probability of surviving past T.

The second tool for modeling time-to-event data in survival analysis is the *hazard function*. The hazard function gives the instantaneous potential for an event to occur within a unit of time, t, conditioned that the individual has survived up to time t [3]. The hazard function, h(t), is defined as follows:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T \leq t + \Delta t | T \geq t)}{\Delta t} \tag{2.4}$$

Here Δt is a unit of time. The numerator to the right of the limit is a conditional probability statement, i.e., a statement in the form of $P(A|B)$ (probability of A given B). The expression to the left of the 'given' part of the conditional probability statement describes the interval that the event occurs between t and t plus some unit of time Δt. The value obtained from the function is a rate since we calculate the ratio of two quantities - a probability (numerator) and a unit of time (denominator) [4]. The result is the rate of occurrences at time t or the probability of having an event per unit of time. The output of the hazard function varies depending on the unit of time used (hours, days, years, etc.) and can take on values ranging from 0 up to infinity (Kleinbaum and Klein, 2012, pp. 9-12). A hazard ratio of one means that the *exposure variable* does not affect the outcome variable. A hazard ratio greater than one indicates that the exposure variable is a risk factor having a negative effect on the outcome variable. Conversely, a hazard ratio less than one means that the exposure variable is a protective factor, positively affecting the outcome variable (Kleinbaum and Klein, 2012, p. 33).

[3] The hazard ratio gives the instantaneous potential for an event to occur at time t given that the individual has survived up to time t.

Hereafter, I will refer to the survival and hazard function simply as $S(t)$ and $h(t)$, respectively.

2.5.1 Kaplan-Meier

There are several ways in which we can model or approximate $S(t)$ and $h(t)$. These approaches can be grouped into non-parametric, semi-parametric, and parametric models. Non-parametric models do not impose any assumptions on the distribution of the data. In particular, non-parametric models do not assume that the distribution holds a specific shape or form. An example of such a model is the KM [5]. In this model, the survival curve is plotted using a piecewise constant function (step function) where the curve remains constant across selected time intervals (Kleinbaum and Klein, 2012, pp. 52-53). By plotting the KM estimate for two subject groups, we can show how the survival experience of the groups differ from each other. Such a comparison can be useful in clinical trials where you want to measure the effects of a treatment in which one group is given the treatment and the other is not (placebo) (Goel et al., 2010). In Figure 2.3 we see an example of such a comparison. The plot shows survival curves for two widely used femoral stems in Norway. The event of interest is revision surgery and the timeline shows years until revision surgery. The plot is presented with 95% confidence intervals. Although the two stems has comparable survival curves, Lubinus SP II seems to perform better from 2 years onward.

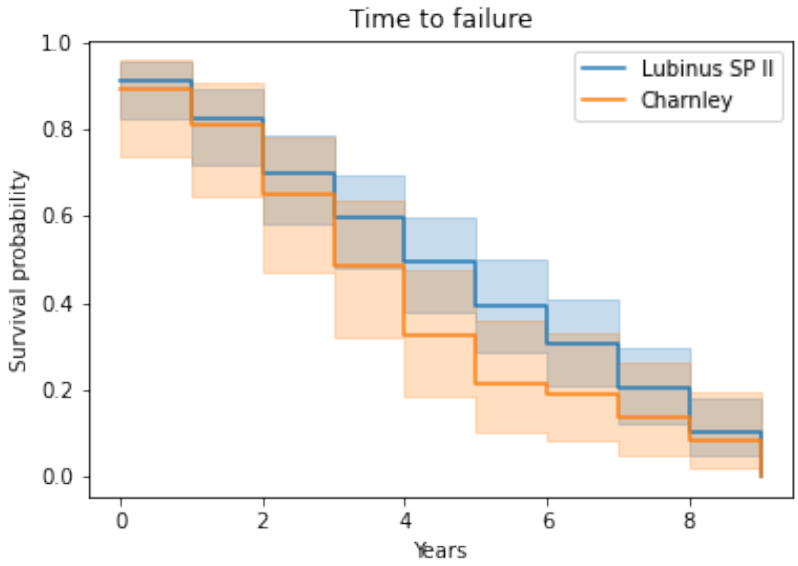


Figure 2.3: KM survival curves comparing two widely used prostheses in Norway. The plot was produced using lifelines - a software library for survival analysis (Davidson-Pilon et al., 2021)

The KM estimator is defined as follows:

$$\hat{S}(t(j-1)) = \prod_{j-1}^{i=1} \hat{Pr}(T \geq t(i) | T \geq t(i)) \tag{2.5}$$

[4]The hazard function is a rate, not a probability

[5]The KM estimator is also known as the 'product limit estimator'

Here, $\hat{S}(t(j-1))$ denotes the probability of survival past $t(j-1)$. $\hat{Pr}(T \geq t(i)|T \geq t(i))$ gives the probability of survival for all individuals i up to time $j-1$. These are individuals that are still considered to be at risk. Those at risk are individuals who have (a) not experienced an event or (b) are not censored. The total probability of surviving to time j is computed by multiplying all preceding probabilities of survival up to $j-1$ as specified by the product operator ($\prod$) (Kleinbaum and Klein, 2012, pp. 54-57).

One drawback of the KM method arises when there are no events within a given interval. In that case, the KM estimate over that interval will remain constant, i.e., a survival probability of 1. While such a curve may indeed exist, it is more likely to result from a small sample size. When the sample size is large, the KM estimate is an excellent approximation of the actual survival curve because it approaches a smooth estimator as the sample size grows without assuming any shape or form of the distribution. Note that, in the absence of censoring, the KM reduces to the CDF (Borgan, 1997, p. 9). Another drawback of the method is the difficulty of incorporating covariates. One way is to plot the survival curves for both groups and use the log-rank test to assess whether the two groups are statistically significantly different, i.e., that the null hypothesis is satisfied. However, the KM method can not adjust for confounding variables within the groups (Jager et al., 2008, p. 565). Therefore, it can be difficult to determine the actual effect of the treatment since a confounding variable could coincide with the effect observed from the treatment.

2.5.2 Cox Proportional Hazard Model

To investigate the influence of individual covariates, the Cox Proportional Hazard Model (hereafter Cox model) might be more appropriate since it allows us to adjust for confounding and interaction effects.

The formula for the Cox model is the product of two components, a non-parametric baseline hazard, and a parametric exponential component. The baseline hazard is an unspecified function of time that does not involve covariates. Conversely, the exponential part of the formula includes covariates, but does not consider time, i.e., the covariates are time-independent. Since the covariates are assumed to be time-independent, the estimated hazard ratios of a covariate should remain constant or proportional over time. We refer to this key assumption of the Cox model as the proportional hazard assumption. For this reason, the Cox model is known as a semi-parametric model (Kleinbaum and Klein, 2012, p. 90-94). The Cox model is defined as follows:

$$h(t, X) = h_{0(t)} \times e^{\sum_{i=1}^{p} \beta_i X_i}$$

,

In the survival library used for this research, the baseline hazard $h_0(t)$ represents the *average* subject at time t. We find this subject by computing the mean for each covariate included in the model at each time point. The rightmost term in the expression is computed as e to the linear sum of $\beta_i X_i$ over p predictor variables (Kleinbaum and Klein, 2012, p. 93-94). This latter term corresponds a set of hazard ratios computed for each covariate that functions to inflate or deflate the hazard from the baseline.

The minimum number of observations to include in a Cox regression to avoid sampling bias has been discussed in the literature. Peduzzi et al. (1995) suggest that a minimum of 10

cases per predictor variable divided by the smallest proportion of positive cases (where an event occurred) adequate for regression analysis with proportional hazards. When all events are observed, the proportion of positive cases is equal to 1, and the minimum number of cases to include is simply 10 times the number of predictors. Long (1997) recommends that the number of cases is further increased to at least 100 when Peduzzi et al.'s method results in less than 100 cases.

We typically assess the fit of a Cox model using the concordance-index metric, which is a generalization of the AUC score commonly used in Logistic Regression. The c-index considers the *rankings* of the predictions, not the predictions themselves. In other words, we evaluate the order of the predictions and report the number of concordant pairs out of the total number of pairs in the model. A model that can accurately predict the order of all observations has a c-index of 1. A random model has a c-index of 0.5 (Raykar et al., 2007).

A Cox regression analysis outputs estimated regression coefficients, hazard ratios, p-values, standard errors, and CIs that describe the influence that covariates have on the survival outcome. To explain the output in more detail, we fitted a Cox model to 241 synthetic observations of the Charnley and Lubinus SP II stem. We based the synthetic data on variables representative of actual arthroplasty data and distributed the covariates to the best of our judgment. We leveraged a simulation model from the PySurvival library to create survival times based on the Weibull distribution with a scale parameter of 0.05 and a shape parameter of 4.5 (Fotso et al., 2019). The latter parameter represents the time when 63.3% of the population has experienced an event. We set the risk function to *linear* and the corresponding *risk parameter* to 1. Lastly, we set the coefficient used to calculate the censored distribution to 7. We included the following explanatory variables in the model: age at primary surgery, gender, ASA classification, and implant type. A summary of the fitted model is available in Figure 2.4.

The c-index shows that the goodness of fit is better than a random model but not particularly good (0.68). The *formula* property shows the predictors included in the model, and *computed residuals* indicates whether residuals plots were produced to assess the proportional hazard assumption. In this case, we chose to include scaled Schoenfeld residual plots as a graphical diagnostic of the proportional hazard assumption. Figure 2.5 shows scaled Schoenfeld residual plots for a subset of the covariates in the model. In these plots, we want to verify that the residuals represented as green dots do not form a pattern of change over time, i.e., that the residuals are more or less randomly distributed over time. The p-values shown below each plot aids in this assessment, signifying whether the residuals follow a random distribution (Davidson-Pilon et al., 2021, pp. 120-121). In our case, the test indicated that all p-values were non-significant and thus that the proportional hazard assumption is satisfied.

The table shown in Figure 2.6 shows the effect of each covariate on the outcome along with p-values and 95% CIs. The *coef* column shows the estimated regression coefficients computed for each covariate in the model. The coefficients quantify the effect associated with a unit increase in the covariate. A negative coefficient indicates a decrease in hazard, and a positive coefficient indicates an increase. To interpret the regression coefficients, we exponentiate the coefficients to obtain hazard ratios, also known as relative risks. The *exp(coef)* column shows the exponentiated coefficients or hazard ratios for each covariate. To reiterate the explanation of hazard ratios from Section 2.5, a ratio above 1 indicates a risk factor, and a ratio below 1 indicates a protective factor.

model	CoxPHFitter
duration column	SURVYRS
event column	ANT_REVISJONER
strata	null
baseline estimation	breslow
computed residuals	scaled_schoenfeld
number of observations	241
number of events observed	195
partial log-likelihood	-846.004
time fit was run	2021-06-07 15:27:26 UTC
Concordance	0.682
Partial AIC	1710.007
formula	P_FEMUR_PRODUKT + ALDER + PAS_KJONN + P_ASA

Figure 2.4: An overview table showing details about the fitted Cox model from the Web interface of our prototype

We can see from the table that the factors associated with the greatest relative risk are the ASA classifications *Mangler* and *Moribund*. For example, *ASA = Mangler* yields a hazard ratio of 3.143, i.e. a 314% greater relative risk. This means that, if we hold everything else equal, the ”presence” of *Moribund* will result in a 314% increase in hazard. Likewise, *ASA = Moribund* yields a hazard ratio of 1.442 - a 44% greater relative risk of failure. The CIs for both of these covariates is fairly wide and crosses unity (1) which signifies that the estimates are insignificant. The p-values shown in table for *Mangler* (p = 0.280) and *Moribund* (p = 0.580) confirms the above and we should therefore accept the null hypothesis of no significant effects.

We can also use the Cox model to visually assess the effects of adjusting a single covariate on survival. We can produce a plot to assess the influence that explanatory variables have on the survival outcome with respect to a primary exposure variable. For example, in Figure 2.7, we consider the type of implant to be the primary exposure variable and *ASA classification* as an explanatory variable. The plot shows the effects of varying *P_ASA* factors *Moribund* and *Mangler* for both the Charnley and Lubinus SP II stem. Based on the plot, healthy (Frisk) patients with the Lubinus SP II stem seem to be associated with a slightly better outcome than other configurations. *Moribund* patients with the Charnley stem are associated with the worst outcome. It is important that we take into consideration the goodness of fit (c-index) when interpreting these curves because the baseline hazard in which these curves are inflated or deflated from depends on the fit of the model (Davidson-Pilon et al., 2021). Recall that the data is synthetic and does not reflect the actual performance of the prostheses.

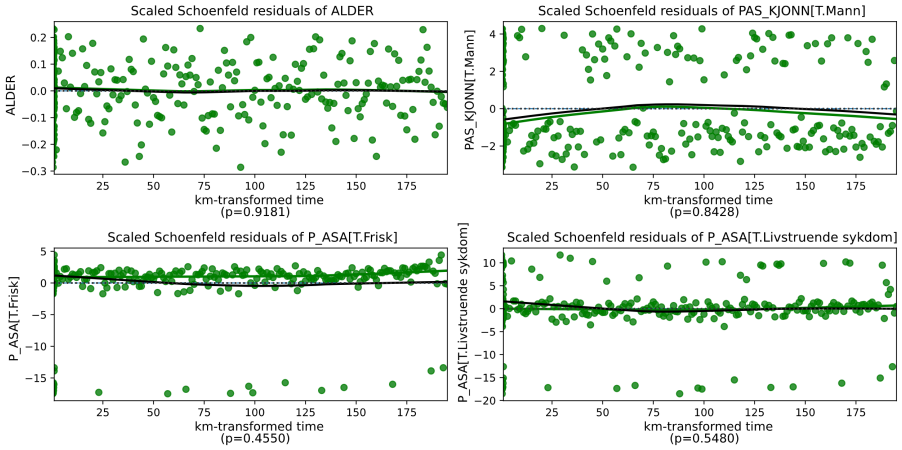


Figure 2.5: The figure shows scaled Schoenfeld residual plots for a subset of covariates in the model. The p-values indicates that the residuals does not establish a changing pattern over time and thus that the proportional hazard assumption is satisfied.

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	z	p	-log2(p)
covariate										
ALDER	-0.012	0.988	0.008	-0.028	0.003	0.972	1.003	-1.562	0.118	3.079
PAS_KJONN[T.Mann]	-0.146	0.864	0.159	-0.459	0.166	0.632	1.181	-0.917	0.359	1.478
P_ASA[T.Frisk]	0.083	1.087	0.307	-0.518	0.685	0.595	1.983	0.271	0.786	0.347
P_ASA[T.Livstruende sykdom]	-0.003	0.997	0.369	-0.726	0.720	0.484	2.055	-0.008	0.994	0.009
P_ASA[T.Mangler]	1.145	3.143	1.060	-0.933	3.223	0.394	25.102	1.080	0.280	1.836
P_ASA[T.Moribund]	0.366	1.442	0.662	-0.932	1.665	0.394	5.283	0.553	0.580	0.785
P_ASA[T.Symptomatisk sykdom]	0.309	1.362	0.364	-0.405	1.022	0.667	2.779	0.848	0.396	1.336
P_FEMUR_PRODUKT[T.LUBINUS SP II]	-0.339	0.712	0.162	-0.656	-0.023	0.519	0.978	-2.101	0.036	4.809

Figure 2.6: The summary table shows the estimated regression coefficients and hazard ratios along with 95% CI and p-values. The screenshot is taken from the Web interface of our prototype.

2.6 Web API

The following section introduces the concept of a Web Application Programming Interface (Web API) - the data interchange medium used for the development of the prototype in this thesis.

A Web API is a mechanism for exchanging resources over a network. Web APIs work by publicly exposing a set of endpoints associated with resources located on a server. Examples of such resources are text content, images, videos, Portable Document Format (PDFs), and structured data. Client applications can query Web API endpoints using the HTTP protocol to retrieve, send, or update resources on a server. The most commonly used HTTP methods are GET and POST. The GET method requests a resource, while POST sends a resource to the server (Park, 2019). Both GET and POST requests accept parameters used to retrieve a selected resource or specify an action to perform on a given resource. See Request for Comments (RFC) 7231 (J. Reschke and R. Fielding, 2014) for a complete list of methods defined by the HTTP protocol. Resources are transported over the network in standardized formats such as plain text, HTML, binary data, or a structured data interchange format such as JavaScript Object Notation or XML (Park, 2019).

There are many good reasons for choosing a web-based Web API as an interchange

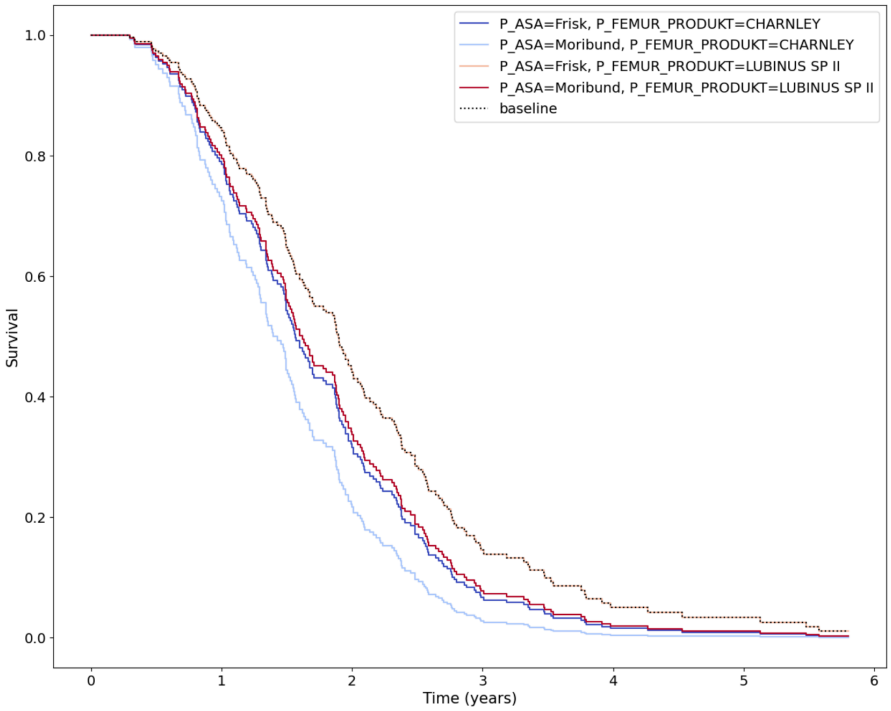


Figure 2.7: Adjusted survival curves showing the partial effects on outcome for ASA classification *Moribund* and *Frisk* (Healthy)

medium for data. Firstly, the data transmission protocol (HTTP) is based upon an open standard. Open standards facilitate adoption and ensure that the capabilities and limitations of the protocol are well understood. Secondly, effective decoupling of the client application from the server application. The client application does not need to be aware of any implementation details of the Web API – only the interface it exposes. Thirdly, due to excellent availability of HTTP libraries among programming languages, front-end developers are given the freedom of choice to use whichever technologies they like.

Chapter 3

Methodology and Methods

3.1 Design Science Research (DSR)

The project will be executed through the application of the DSR methodology. DSR seeks to develop and create purposeful artifacts in the form of constructs, models, methods or instantiations. Here, constructs are defined as vocabulary or symbols. Models are defined as abstractions or representations that use constructs to represent a real-world situation. Methods are algorithms and practises used to search the solution space. Lastly, instantiations are implementations of prototypical applications. The purpose of the produced artifact is to solve concrete organisational problems or business needs (Hevner et al., 2004, p. 77).

DSR is not only concerned about the development of an artifact and emphasises evaluation as one the key activities in the research cycle. After all, the stated goal of DSR is utility and this can only be measured through evaluation (Hevner et al., 2004, p. 80). As such, the justification for the artifact is a measure of its utility that can be assessed through both qualitative and quantitative evaluation methods. (Hevner et al., 2004, p. 78) stresses the distinction made above by considering design to be both a process (set of activities) and a product (artifact). Together, they constitute a so-called build-and-evaluate loop that needs to be iterated a number of times before the final artifact is obtained.

They also advocate the complementary use of both behavioural-science and DSR in developing information technology solutions. They argue that truth (the goal of behavioural science) and utility (the goal of DSR) are "two sides of the same coin" and that both paradigms are paramount for relevance and effectiveness of IS research (Hevner et al., 2004, p. 77) . While utility is usually derived from truth, the authors imply that the application of DSR can aid in the discovery of truth (Hevner et al., 2004, p. 98). To illustrate the interplay between these paradigms, the authors present a conceptual framework for understanding, executing and evaluating IS research. Please see Figure 3.1 on page 24 for a complete overview of the DSR framework.

The environment encompasses the people, organisations and technological infrastructure for which the artifact is to be deployed. The development of the artifact is driven by the business needs of the people within the organisation. These business needs are defined by the goals or tasks of the organisation, or the opportunities that they have identified. Behavioural science is used develop and justify theories that explain a particular phenomenon related to the business needs. On the other hand, DSR is used to develop and evaluate artifacts in order to meet the identified business needs. Both the theory and artifact may need refinement by the

justify/evaluate activities to conform to the business need. The knowledge base is composed of methodologies and fundamental theories, frameworks, instruments, constructs, models, methods, and instantiations for which the artifact is supported by or built upon. Methodologies used in behavioural science includes data collection and empirical analysis techniques. The quality and effectiveness of artifacts is typically assessed using computational and mathematical methods, such as performance metrics (Hevner et al., 2004, pp. 80-81).

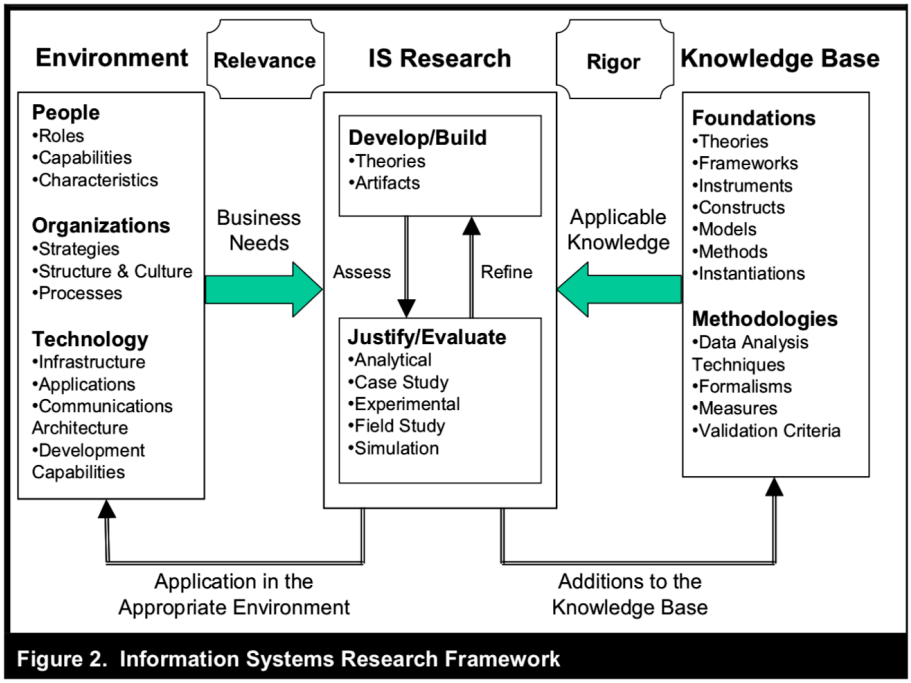


Figure 3.1: Information System Research Framework diagram depicted in Hevner et al. (2004)

Hevner et al. (2004) presents seven guidelines for DSR in Information Science (IS) research that outline the fundamental principles and values of the methodology. These guidelines are listed below as they are described in (Hevner et al., 2004, p. 83).

1. **Design as an Artifact** - Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
2. **Problem Relevance** - The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3. **Design Evaluation** - The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
4. **Research Contributions** - Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
5. **Research Rigor** - Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
6. **Design as a Search Process** - The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7. **Communication of Research** - The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.

The guidelines direct attention towards an understanding of the problem space and provides valuable guidance for producing purposeful artifacts and demonstrate their utility through well-defined and rigorous evaluation methods. These guidelines serve as a valuable asset to the application of DSR and will be considered more carefully throughout the project timeline.

3.2 Development Methods and Methodologies

This section present the requirement specification process and describes the Dynamic System Development Methodology (DSDM) that was used to develop the artifact of this thesis. DSDM was chosen for its collaborative strength and focus on delivering on time. The main points of collaboration were the implementation of the data mining tasks and the Web API. In addition, two members of the project team were working on the front-end development and exchanged aspects of Human-Computer-Interaction (HCI) design. The results from the back-end development has served as input to the front-end development. There were also shared valuable feedback and comments between the back-end and front-end team.

3.2.1 Requirement Specification

Requirements development is an important activity in software engineering and a fundamental step toward building a solution. The purpose is to define the needs and requirements

and describe the functionality of the system. We typically differ between two types of requirements - functional and non-functional ones. Functional requirements describe what the system is supposed to do, such as which actions or procedures to support. Functional requirements are absolutely necessary to implement for a system to fully function. Qualities related to how the system supports these operations are the non-functional requirements of the system. Non-functional requirements refer to qualities such as speed, responsiveness, reliability, and usability that one wants the system to possess (McConnell, 2004, pp. 38-43).

There are several good reasons for specifying the requirements upfront. First, it allows the customer to review, give feedback and redefine the specifications to fit their needs. Second, it sets a baseline for what the system is supposed to do. If this baseline is met early in development, the requirements can be tweaked and extended. However, in practise, there are often cost and time constraints involved, which the development team must consider. These constraints are more maintainable when specifications for the system are set in advance because it makes sure that the most critical components of the system are prioritized. Third, well-defined requirements help avoid arguments between involved parties that can occur due to disagreements or misunderstandings regarding what the system is supposed to do. Should such disputes nevertheless arise, a look at the requirements can aid in resolving misunderstandings and conflicts (McConnell, 2004, pp. 38-43).

3.2.2 Dynamic System Development Methodology (DSDM)

First developed as an offspring from the Rapid Application Development (RAD) approach, the DSDM is a proven and versatile framework for Agile project management. The foundations of DSDM was laid in 1994 to address the shortfalls of the RAD method like lack of quality control and project structure, and the slowness and inflexibility of the traditional approach. Recognizing their strengths and weaknesses, DSDM combines the best of both worlds - the rigidity and quality control from the traditional approach and transparency, flexibility, communication, and business involvement from RAD. This makes it suitable for both small and larger projects requiring more oversight, structure and governance. At the core of the DSDM lies eight principles that team members must embrace to fully take advantage of the methodology (Craddock, 2014):

- Focus on the business need
- Deliver on time
- Collaborate
- Never compromise quality
- Build incrementally from firm foundations
- Develop iteratively
- Communicate continuously and clearly
- Demonstrate control

These principle underpin the philosophy of DSDM which states that: ”best business value emerges when projects are aligned to clear business goals, deliver frequently and involve the collaboration of motivated and empowered people” (Craddock, 2014). In DSDM, focus, collaboration, and a clear understanding of the business needs are key for successful project delivery. The method attempts to deliver a minimal subset of requirements and a functional solution, without compromising cost and time. This is in stark contrast to the traditional approach where the project specification remain fixed and time and cost are adjusted in an ongoing manner. In the worst case, the project can go out of budget and fail to deliver on time. From a business perspective, this can be detrimental - both internal and external to the organization. DSDM mitigates this risk by ensuring that, at the very least, a minimum set of features are always delivered on time and within budget. This is achieved by having a fixed deadline and budget, and a baseline acceptance criteria for quality. Some leeway is accounted for by allowing the requirements specification to be adjusted should time be scarce or the cost too high. In practise, this is accomplished using timeboxing and the MoSCoW technique. MUST have, SHOULD have, COULD have, WON’T have (MoSCoW), is the technique used to group work items into a logical order according to their importance. Timeboxing breaks the project into smaller, incremental parts, each with its own fixed budget and deadline. The most important features are implemented first and less critical ones are assigned a lower priority (Craddock, 2014). Timeboxes start with a *kick-off* or a brief session in which the team discusses the objectives and deliverables for the upcoming Timebox. Similarly, we conclude Timeboxes with a close-out meeting to debate, inspect and potentially accept deliverables. The close-out is also an opportunity for the team to reflect, learn and prepare for the next Timebox (Consortium, 2021). The DSDM offers two formats for timeboxing: DSDM structured Timebox and free format Timebox. A structured Timebox requires an investigation, refinement, and consolidation phase between the kick-off and close-out and requires more supervision than a free format Timebox. In contrast, the free format Timebox is more loosely structured and does not demand formal review points between kick-off and close-out. However, the free format Timebox relies on the presence of a Technical Coordinator to provide feedback in an ongoing manner (Consortium, 2021).

If these practises are followed, DSDM, guarantees that a minimum subset of requirements are delivered in the worst case scenario (Craddock, 2014).

The success of a DSDM project is dependent on a number of factors. First, the approach has to be embraced by all parties that are involved in the project. Second, a healthy relationship with the customer must be maintained at all times. Third, team members must be committed to their roles and work closely together. Lastly, end-user involvement is encouraged and should occur frequently throughout the project (Craddock, 2014).

3.3 Knowledge Discovery in Databases (KDD)

The last few decades have seen an unprecedented growth in terms of processing power and storage capacity of computers. As processing power and storage is becoming increasingly cheaper, the amount of data accumulating is growing as well (Chen and Rossman, 2014). To be able to leverage this data in a useful way, KDD has been established as a young and interdisciplinary field attempting to bring forth real value and insight from the data.

Fayyad et al. (1996) defined KDD as: "...the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Fayyad et al., 1996, pp. 40-41). KDD can be considered a multi-step process for extracting knowledge from raw data. Moreover, KDD establishes a framework for how to store, access, and apply algorithms efficiently and interpret and visualise data, with emphasis on knowledge as the final end-product.

The first step of the KDD approach is to develop an understanding of the application domain and define a set of questions we want to answer. Then, we select a data sample, often raw data, that we can extract and discover new knowledge from. Careful selection is advised, as introducing too many or too few variables can influence the results negatively. The third step, referred to as data cleaning and preprocessing, concerns the handling of missing data and the removal of noise or outliers that may distort a selected dataset. The fourth step is data reduction and projection which is the practise of reducing dimensionality or transforming data to other representations. For example, one may apply Principal Component Analysis for dimensionality reduction, scale continous variables or dummy encode categorical data.

Next, we have to decide on the goal of the KDD process. This influences the data mining step where we must choose the appropriate model for the problem we are trying to solve. For example, are we concerned with a classification or regression analysis task?. The sixth step deals with the technicalities such as choosing which data mining algorithm(s) and which models and parameters to use. Afterwards, we perform data mining and search for patterns of interest in a particular representational form, such as classification rules or clustering. Following data mining, we must interpret the results and evaluate whether the identified patterns are actually meaningful. It is important to reason about the patterns to understand how they originate and to evaluate their relevancy. Visualising the data may aid greatly in gaining intuition about these patterns. The end-product of KDD is high-level domain applicable knowledge that must be documented and possibly incorporated into other systems for application. If the knowledge derived from the process proves inapplicable or deficient, we may need to step back to earlier activities in the process for refinements. Often times, several iterations are necessary to obtain desirable results (Fayyad et al., 1996, p. 42). Figure 3.2 shows an overview of the KDD process.

Numerous science and business applications owe its success to the KDD process. For example, SCIKAT - a system for analysing and classifying sky objects have been used in astronomy with great success. In business, the model has been used for marketing, investment and fraud detection systems. Some notable examples are HNC's Falcon and Nestor's PRISM systems which is being used for credit card fraud detection. Within aerospace engineering, the CASSIOPEE troubleshooting system is used to diagnose and predict problems for the Boeing 737 (Fayyad et al., 1996). KDD may be particularly beneficial for the healthcare sector due to the vast amount and complex nature of healthcare data being accumulated. Here, KDD has a broad range of potential use cases, including the evaluation of treatment effectiveness, predicting the length of stay, diagnosis prediction, hospital resource management and decision support systems (Tomar, 2013).

In summary, the KDD process lays out the activities involved in knowledge discovery in a structured and distinctive order. Each activity builds on one another and have its own objective. The emphasis on knowledge as the final-end product serves to direct the process in the right direction. In case of inapplicable or deficient results, the process allows us to step back and do the necessary the refinements in order to get more favorable results. KDD has

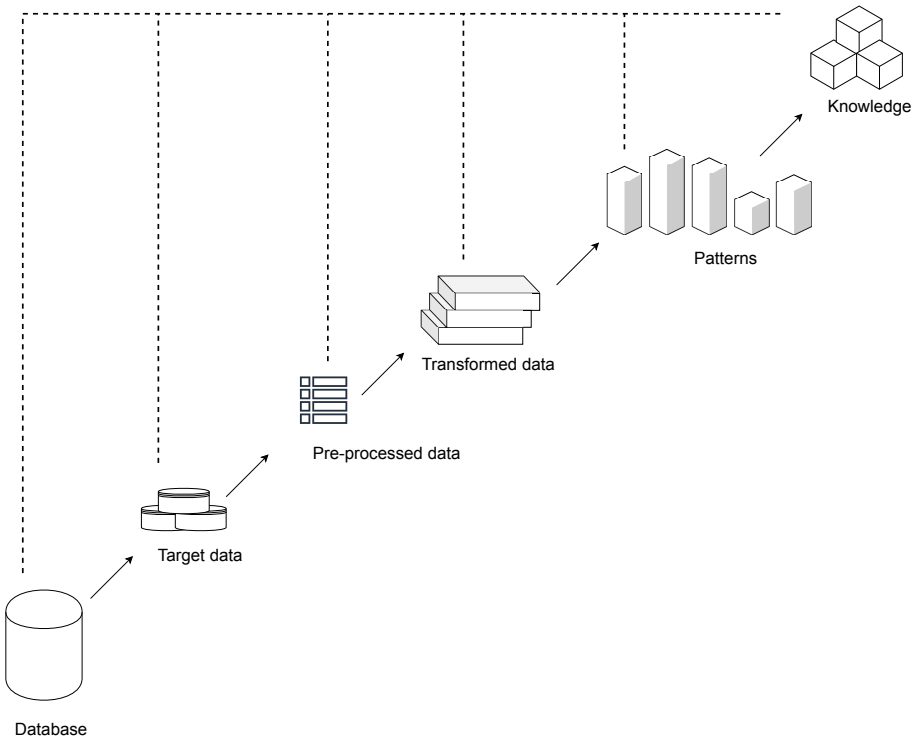


Figure 3.2: The KDD process. The stippled lines shows the iterative nature of the process.

been used for both small and large projects with success. For these reasons, I have chosen to adhere to the principles and practises of KDD in this project.

3.4 Evaluation

The overall aim of DSR is to produce a viable and relevant artifact with well-defined utility and efficacy. The evaluation activity stands as one of the guiding principles of DSR. The guidelines emphasize the importance of demonstrating utility, quality, and efficacy with rigorous and well-executed evaluation methods (see Section 3.1). In this thesis, we consult relevant stakeholders and experts for feedback and evaluate the design artifact with the SUS tool. The evaluation form used to validate the design artifact in this thesis is presentation and demonstration in conjunction with the SUS questionnaire.

3.4.1 System Usability Scale (SUS)

The SUS is a quick and dirty" technique for assessing the usability of a system that has established itself an established industry standard shown to produce reliable and valid results. The questionnaire consists of 10 predefined statements related to usability traits of the system, such as its complexity, frequency of use, and ease of use. Each statement accepts five potential responses - ranging on a scale from *Strongly agree* to *Strongly disagree*. We interpret the results on a scale ranging from 0-100, in which 68 is considered average. Scores above the average grade of C is considered to be within the acceptance range. A score between 74 and 80 corresponds to a grade of B, and an excellent score (80 or greater) corresponds to the grade A (Sauro, 2011a). Figure 3.3 shows a visual representation of the SUS scale with its

various grades and acceptance ranges.

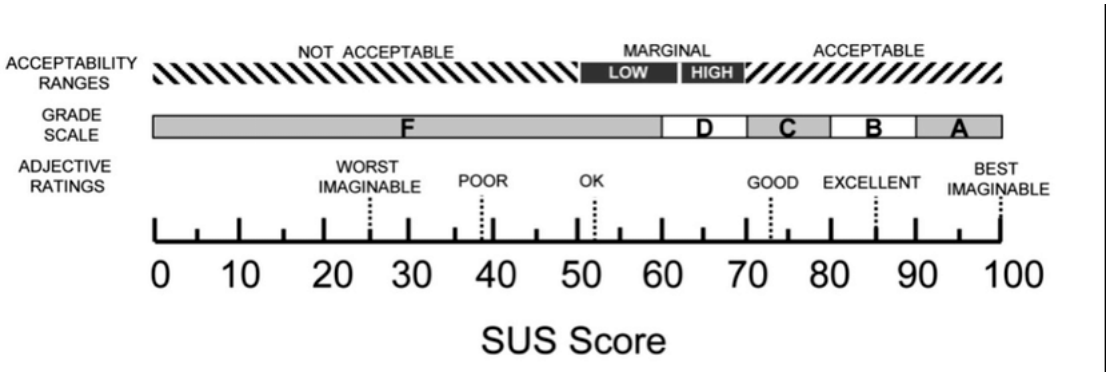


Figure 3.3: Visual representation of the SUS adopted from (Bangor et al., 2009)

Chapter 4

Establishing Requirements

4.1 Meeting with the register

A meeting with the registry took place on the second of October 2020 at the register's offices in Bergen. Attending the meeting was my supervisor and the three other master students working on the project. Present from the registry was the director and researcher, Ove Furnes, statistician Anne Marie Fenstad, researcher and engineer Paul Johan Høl, and Dr. Peter Ellison. The plan was to introduce the new master students, the theme of our master theses, inquire about personas and database access, and discuss clinical questions to model using data mining. Also on the agenda was how to handle the data safely and responsibly and to plan out activities for the coming months. The contracts for the datasets were also a topic of discussion.

Each student gave a short presentation outlining the vision and ideas for the project. The presentation was well received and led to a lengthy discussion on various topics, including the arrangement of interviews with patients and staff at the register. There was also talk about the necessity of defining a 'benchmark' to compare and evaluate prosthetic devices using statistical models. For the hip register, there was broad agreement that the Charnley prosthesis was a good starting point. For the knee register, the "Profix" prosthesis showed excellent results. The register informed us about a population-based registry study reported in Hallan et al. (2012) evaluating the performance of a particular titanium femoral stem widely used in Norway since 1984. Despite excellent results up to the year 2000, the performance of the stem began to deteriorate in the period 2001-2008. The study used KM survival curves and Cox regression to measure the effects of covariates of interest. They found that patients who underwent surgery in 2001-2008 had an adjusted relative risk that was 4.7 times higher than for 1996-2000 and that high stem offset, male sex, and small stem size were risk factors for revision surgery Hallan et al. (2012). The register urged us to review the paper and attempt to replicate some of the findings in the study. In particular, the register was interested in whether underperforming prostheses can be detected earlier using machine learning. The register found a similar research paper for the student working with Total Knee Arthroplasty data.

We were advised that the Kaplan-Meier method and Cox Regression were the de facto standard for registry-based studies on survival analysis. The widespread use of these methods led us to explore survival analysis further and include both Kaplan-Meier and Cox regression in the requirement specification of the system.

4.1.1 Requirements

The following section describes the requirements and the intended functionality of the prototype developed in this thesis.

The following requirements were set during and in the days following the meeting with the register.

4.1.2 Functional Requirements

- Present statistics relevant to clinicians and researchers.
- Provide predictions of clinical outcomes of THA.
- Provide classification of clinical outcomes.
- Produce survival curves for different subject groups (KM method).
- Allow the user to perform cluster analysis
- Allow the user to identify risk factors for THA (Cox regression).

Non-functional Requirements

- Let the user choose which input data and parameters to train the model with (flexibility).
- Documentation should be available for all tasks in the system.
- The system should be tested frequently to identify faults.
- Use complementing technologies to allow for interoperability between heterogeneous systems. The choice of technologies for the data mining component should not limit which technologies the front-end team chooses to use (and vice versa).

4.2 Technologies

The following section introduces the main technologies and tools for the prototype development.

Scikit-learn

Scikit-learn is a software package written in Python (Van Rossum and Drake Jr, 1995) with numerous machine learning algorithms for supervised and unsupervised problems (Pedregosa et al., 2011). The project values code quality and testing in favor of providing as many features as possible. The development of Scikit-learn is community-driven, and the core team is open to contributions from external contributors. The documentation is broad, with approximately 300 pages of class references and a comprehensive collection of complete and detailed examples. The API follows a consistent and minimalistic interface which eases the learning curve for new developers. For instance, most algorithms in Scikit-learn implement

one or more of the following objects: estimator, predictor, transformer, and model. An estimator is used to fit data to a model. Predictors implement the 'predict' method used to make predictions on in-sample or out-of-sample data. A transformer implements the 'transform' method to manipulate data into a suitable format. Lastly, a model implements the 'score' method to estimate the goodness of fit of the provided data (Pedregosa et al., 2011, pp. 2825-2830).

Scikit-learn was selected for the clustering and classifier component of the system for all of the above advantages. The following reasons influenced our choice as well: (1) Earlier work by Kristoffersen (2019); Longberg (2018) demonstrated the use of regression and clustering in Scikit-learn using a similar dataset, thus laying the groundwork for further research in the area. Longberg observed that visualization was key to understanding data and proposed that future work should consider automating this process in a user-friendly interface. (2) Scikit-learn is distributed under the BSD license, making it suitable for academic and commercial usage, including Pharma and healthcare industry (Brajer et al., 2020).

Lifelines

Lifelines - a survival analysis module for the Python programming language was used for the survival analysis component. Lifelines provide implementations of a wide collection of parametric, semi-parametric, and non-parametric survival models such as the KM model, Cox Proportional Hazard model, and Weibull model Davidson-Pilon et al. (2021). A total of 21 univariate and regression models are available in the library. Methods to compare the difference in survival between two or more populations are also available such as the logrank test and the restricted mean survival times metric. There are also plots for evaluating a model's performance and comparing the survival curves produced by two different models. The library has ample documentation and provides a large number of examples covering most of its API. Interested readers can find an introduction to survival analysis and tutorials with real-world applications on the website of Lifelines. Lifelines was chosen for this research because it provides implementations of the Cox Proportional Hazard Model and the KM method out of the box.

FastAPI

The Web API was built on top of FastAPI - a web framework written in Python (FastAPI, 2021). FastAPI was chosen for several reasons. (1) The features specified in the requirements 4.1.2 require satisfactory performance. In particular, some of the computations carried out by the system can be computationally heavy and slow. (2) Berge (2019) highlighted some of R and R Shiny's limitations and suggested Python as a candidate for a more complex system. He notes that customization of the user interface is tedious, limited, and restricted to predefined components. According to Berge (2019), future systems should consider Python as R seems more suitable for smaller applications. He argued that for future work, a back-end/front end model would increase efficiency and be more beneficial from a user's point of view by relieving them of tasks and being more user-friendly. The limitations of R Shiny concerning customizability were an important design consideration. Although R provides a powerful and broad range of statistical tools, the limitations with respect to user interface design and Berge (2019)'s considerations lead us in the direction of Python and thus FastAPI.

These considerations were discussed and made with the other collaborators working on this project. (3). Scikit-learn and Lifelines were explored ahead of time, and FastAPI seemed to complement these libraries well because they are all written in Python.

Web Technologies

The front-end is built on top of standard Web technologies, namely HTML, CSS, and JavaScript. We relied upon Bootstrap for the UI layout and UI components. DataTables was used to enhance HTML tables with search and sort functionality (SpryMedia Ltd., nd), and we produced plots using the Open-source graphic library Plotly.js (Plotly Technologies Inc., 2015). We communicate with the Web API programmatically using native JavaScript.

Trello

Trello is a web-based and Kanban inspired collaboration tool for organizing projects (Trello, 2021). Trello organizes work in a board of lists containing cards indicating a basic unit of work. Cards or work items are transferable across lists which allows for a Kanban styled workflow. For example, one can group items into To-Do's, Doing, and Done and transfer items from one list to another in a sequential order. We used the service to organize and break larger assignments into smaller and more approachable increments. We also used the platform as a discussion board among teams members for less formal discourses deemed inappropriate for Git issue tracker. These discussions was tied to the work item being worked at which helped prevent potential derailments. The tool integrated well with the DSDM by allowing us to color-code the list of requirements according to the MoSCoW prioritization technique (see Section 3.2.2). An overview of the Trello board used for the back-end is depicted in Figure 4.1.

Other technologies

In addition to the technologies above, a handful of other software modules were used to a varying degree. A few noteworthy mentions are pandas, NumPy, pydantic, patsy, formulaic, and SQLAlchemy (Reback et al., 2020; Wes McKinney, 2010; Harris et al., 2020; Bayer, 2012). We used Git for version control (Git, 2021) and GitHub for repository management (GitHub, 2021).

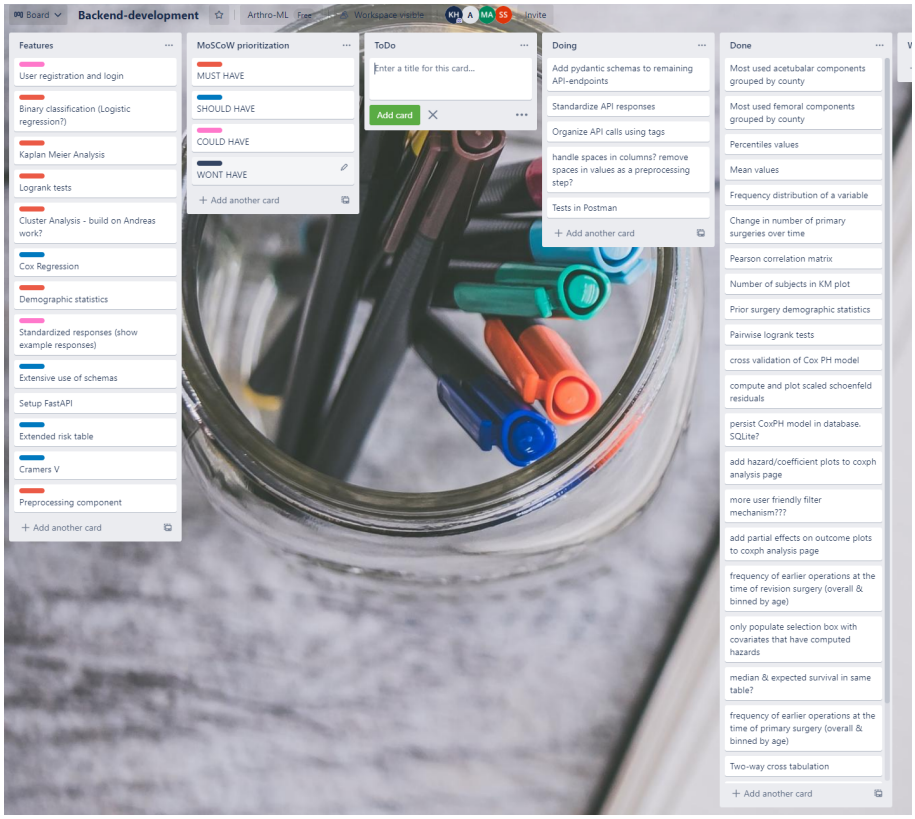


Figure 4.1: Trello board used by the back-end development team.

Chapter 5

Data Material

5.1 Data

The following section describes the dataset and the data mining tasks that was defined and used in the development of the prototype. The data is experimental and reflects the registry data closely in terms of variables and their values.

The material used in this thesis was provided by Dr. Pete Ellison from the Biomedical Engineering Laboratory at Haukeland University Hospital in the form of a data sample of THAs conducted in Norway during 1995-2018. A total of 10 000 THAs with a 9 year follow-up until revision surgery was included. The starting point is the year of primary surgery and the endpoint is the year of revision surgery. The event of interest was revision surgery and the outcome variable of interest was time until revision surgery (failure). The mean age for patients undergoing primary surgery was 55 years (SD = 10) and the mean survival duration for implants was 4.12 years (SD = 2.72). The number of male patients is 5013 and the number of female patients is 4987. In total, the dataset contains 156 variables. The majority of these are categorical (n = 154). The two continuous variables are the implants survival recorded in years and the age of the patient during primary surgery. An overview of patient demographics can be found in Tables 5.3 and 5.2.

Excluding expired products, the most frequently used femoral stem was the Lubinus SP II (n = 163, m = 83). For the acetabular cup, the most frequently used product was the Mittelmeier (n = 158, m = 80).

Patients health status is indicated by the American Society of Anesthesiologists (ASA) classification system. This system is used to assess a patient's health on a scale of 1 through 6. In the first category, ASA I, we have healthy patients, while in the fifth category we have patients who are likely to die if surgery is not performed (ASA, 2021). The excerpt used in this research has only the first five categories. These are, in order of severity: healthy patients, asymptomatic condition that increases risk, symptomatic disease, life-threatening disease and moribund. Studies have shown that higher ASA scores is associated with a higher risk of revision in joint replacement surgery (Schaeffer et al., 2015; Ferguson, Silman, Combescure, Bulow, Odin, Hannouche, Glyn-Jones, Rolfson, and Lübbeke, Ferguson et al.). Therefore, ASA classification score should be considered a potential predictor of early revision surgery. To investigate the potential of ASA score as a predictor of revision surgery, pairwise comparisons using the logrank test were performed for all $n \geq 2$ unique groups in ASA. Table 5.1 shows that all comparisons have a p-value greater than the cut-off

		test statistic	p	log2(p)
Asymptomatic condition	Healthy	0.54	0.46	1.11
	Life-threatening disease	1.59	0.21	2.27
	Moribund	1.64	0.20	2.32
	Symptomatic disease	1.10	0.29	1.77
Healthy	Life-threatening disease	0.29	0.59	0.76
	Moribund	0.30	0.59	0.77
	Symptomatic disease	0.11	0.74	0.43
Life-threatening disease	Moribund	0.00	0.96	0.05
	Symptomatic disease	0.05	0.83	0.27
Moribund	Symptomatic disease	0.05	0.83	0.27

Table 5.1: Pairwise logrank tests was calculated with Lifelines to compare the survival experience between groups with different ASA categories (Davidson-Pilon et al., 2021). The tests are chi-squared under the null hypothesis with 1 degree of freedom. The p-values show there is no significant difference among the groups ($p > 0.05$ for all comparisons).

value of 0.05, indicating that the null hypothesis should be accepted. These tests were calculated for the total population; running the tests on sub-populations may still yield significant differences. However, subsequent tests for the Charnley and Lubinus SP II resulted in no significant difference among the various ASA scores.

Variable	**n**	**Min**	**q_1**	**$\tilde{x}$**	**$\bar{x}$**	**q_3**	**Max**	**s**	**IQR**	**#NA**
Age	10000	31	48	55	54.6	62	78	10.0	14	0
Follow up time	5011	0	2	4	4.1	6	9	2.7	4	4989

Table 5.2: Patient characteristics: continuous variables.

Variable	Levels	n	%	∑%
Gender	Male	4987	49.9	49.9
	Female	5013	50.1	100.0
	all	10000	100.0	
ASA	Healthy	1632	16.3	16.3
	Asymptomatic condition	1648	16.5	32.8
	Symptomatic disease	1724	17.2	50.0
	Life-threatening disease	1641	16.4	66.5
	Moribund	1640	16.4	82.8
	Missing	1715	17.1	100.0
	all	10000	100.0	
Surgical Position	Lateral	3249	32.5	32.5
	Supine	3419	34.2	66.7
	Missing	3332	33.3	100.0
	all	10000	100.0	
Surgical Approach	Anterior (Smith-Petersen)	1602	16.0	16.0
	Anterolateral	1718	17.2	33.2
	Lateral	1685	16.9	50.0
	Posterolateral	1643	16.4	66.5
	Other	1663	16.6	83.1
	Missing	1689	16.9	100.0
	all	10000	100.0	
Stem Material	Steel	3356	33.6	33.6
	Titanium	3333	33.3	66.9
	Cobalt-chrome	3311	33.1	100.0
	all	10000	100.0	
Femoral Fixation	Cemented with antibiotics	2591	25.9	25.9
	Cementless without antibiotics	2457	24.6	50.5
	Cementless	2516	25.2	75.6
	Missing	2436	24.4	100.0
	all	10000	100.0	
Acetabular Fixation	Cemented with antibiotics	2412	24.1	24.1
	Cementless without antibiotics	2621	26.2	50.3
	Cementless	2504	25.0	75.4
	Missing	2463	24.6	100.0
	all	10000	100.0	

Table 5.3: Patient characteristics: nominal variables.

Chapter 6

Prototype Development

We developed the artifact in small increments within four major iterations timeboxed at 2-3 weeks. We implemented features incrementally according to MoSCoW priority, and added tests in an ongoing manner. Although we spent most of the time developing the API, we devoted a substantial amount of time exploring the data and learning about technologies related to the project.

Timeboxes were initiated with 'Kick-offs' (see Section 3.2.2) or short and informal briefings establishing the objectives for the following weeks, such as what features to implement or rework. We opted for the free format Timebox of DSDM rather than the standard DSDM structured Timebox format because we deemed the loosely structured Timebox more appropriate for such a small team (see Section 3.2.2).

Frequently held meetings compensated for the lack of structured timeboxes and helped to keep the project on track. We concluded each Timebox with a 'Close-out' summarizing accomplishments, discussing challenges, and rescheduling work for the next Timebox.

6.1 Initial work without data

In the early phases of our work, we envisioned a system comprised of three major workflows. Firstly, allow the user to assemble a pipeline of transformations and create new datasets. Secondly, let the user create a machine learning model and train it on one of the transformed datasets. Thirdly, enable them to run predictions on out-of-sample data and present results to the user.

6.1.1 Procedure for preprocessing

In the weeks leading up to the data delivery, we worked on a procedure for automatic preprocessing of data. The idea was to use pipelines from Scikit-learn to assemble a series of transformation steps such as feature selection, imputation of missing data, scaling of numerical features, and encoding of categorical features. The output of a pipeline is a transformed dataset that we can feed into a machine learning model. A powerful feature of pipelines is that flow can be split within the pipeline, allowing us to selectively transform specific parts of the data. For instance, we may want to impute a subset of features and give different treatment to categorical and numerical features.

We managed to successfully implement a preprocessing routine on the API that utilized the pipelines module from Scikit-learn. Although functional, our solution had a couple of flaws. Firstly, compositing such a complex data structure in a user interface is a non-trivial task. See Appendix B for a depiction of the data structure required to build a pipeline in our system. Secondly, original feature names did not remain intact after processing the data. The last part is important because we must be able to map the transformed features back to the original features in the UI meant for end-users. The processing routine was left unfinished and was not integrated into the final system. Further descriptions of the preprocessing routine is found in Appendix B.

6.2 First iteration

The prototype development began in late October of 2020 after Dr. Peter Ellison at Haukeland University Hospital provided us with test data to build the prototype around. We spent the first week exploring the dataset, structuring the API, and scheduling two sessions with the front-end team to discuss what demographics might be interesting to extract from the dataset. There was consensus within the group that gender distribution, indications for revision surgery, and the frequency distribution of implants were interesting aspects to consider. However, it quickly became clear that we lacked the necessary insight from the intended user groups, such as patients and arthroplasty researchers. Stolt-Nielsen from the front-end team later conducted a survey confirming our suppositions that patients were interested in implant use and indications for revision surgery. The survey found that patients would like to know more about their implants, possible risk factors and implant survivorship. In addition, patients were also interested in information regarding patients within their age cohort citepsunniva-nielsen.

During the following two weeks, we implemented API endpoints for extracting the above information and wrote tests in Postman (Postman, 2021) to ensure the methods adhered to specification. Figure 6.1 shows an example of tests run in Postman. The tests helped us identify errors and bugs before we pushed code to version control (Git). We ended the first iteration by tutoring the front-end team on setting up and interacting with the API. Table 6.1 shows a list of endpoints completed during the first iteration.

Endpoints	**Description**
Most popular femoral stems	Five most used femoral stems
Most popular femoral stems by county	Five most used femoral stems by county
Most popular acetabular cups	Five most used acetabular cups
Most popular acetabular cups by county	the five most used acetabular by county
Indications for revision	Distribution of indications for revision
Indications for revision by age groups	Indications for revision broken down by age groups
Average implant survival	Average survival duration of implants
Missing values	Total number of missing values for each variable

Table 6.1: API endpoints implemented during the first iteration

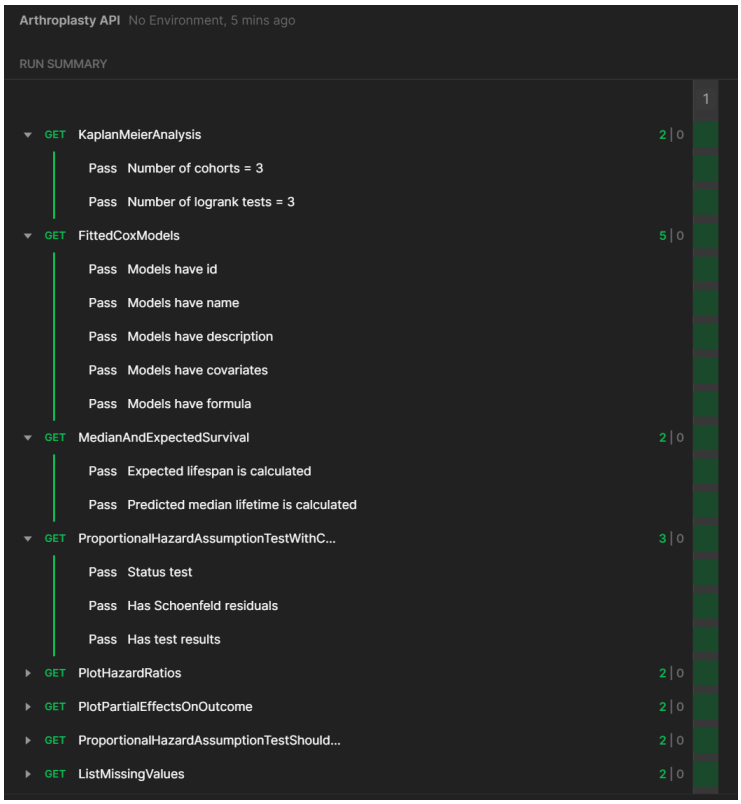


Figure 6.1: Testing of API endpoints in Postman

6.3 Second Iteration

The second iteration began on the 16th of November and lasted approximately two weeks. The plan for these weeks was to incorporate Lifelines and Scikit-learn and add endpoints for KM analysis and binary classification using Logistic Regression.

We started by revisiting and refactoring procedures from the first iteration. Some minor mistakes were overlooked and picked up by the front-end team. For example, implants tagged as 'expired' were not dropped from the list of most used implants. These implants are troublesome because we do not know the actual name of the implant used.

Following these corrections, we added a routine for computing KM survival curves and log-rank tests. This routine takes as input a variable (treatment) corresponding to the various groups we want to compare. We dispose of unwanted groups by applying a filter that selects only the groups of interest. We may apply additional filters to impose further restrictions upon the groups. For example, we can use the filter *gender == "male"* to compute the survival curves of males but not females. We perform pairwise log-rank tests for all $n > 2$ unique groups and output results from these together with the computed survival curves.

The output from this routine includes survival curves, confidence intervals, extended risk tables, median survival time, and log-rank tests. All of which are conveniently accessible through Lifelines. Pairwise log-rank tests were later added as a standalone routine because such tests are useful outside of a KM analysis as well.

Next, we added a routine for training and performing classification using a Logistic Regression model from Scikit-learn. We split the data into a training and test dataset internally and allow the user to supply a formula based on Wilkinson-style notation to construct the

model. For example, the formula $y \sim \ gender + age$ creates model with predictors *gender* and *age*, and the intercept y. The response variable, y, was hardcoded as a binary outcome variable corresponding to implant survival less than 4 years or greater than or equal to 4 years.

The output from the Logistic Regression endpoint is a precision-recall curve, ROC curve, and a classification report. The plots are computed on the API and transported over the network as Base64 strings. Base64 encoding allows us to conveniently represent binary data as an ASCII string and transfer graphics over the API without corrupting the asset (Josefsson, 2006).

Halfway through the iteration, we held a short meeting debating whether we should build a minimal front-end application to complement the data mining methods on the Web API. We realized that simply providing 'raw' data analysis over an Web API is insufficient for KDD. To fulfill the last and possibly the most important step in the KDD process, *Knowledge presentation* (see Section 3.3), we had to present results in a suitable representable form that allows for evaluation and interpretation of the analysis'. This last step is what ultimately validates our system and provides knowledge and utility for end-users.

Based on the above, we decided to shift our efforts towards the development of a front-end to accompany the methods available on the Web API. To save time, we based the front-end on the micro web framework, Flask, and built a minimal Web application to interact with the Web API. We designed the UI with the free and Open-source front-end framework, *Bootstrap*, and use 'vanilla' JavaScript to communicate with the Web API. The data used to populate the UI elements comes from the Web API.

Towards the end of the second iteration, we integrated the Logistic Regression procedure into the front-end. We created a layout that organizes functionality and results from the model into four distinct sections. Figure 7.10 shows the UI that was designed for the procedure. In the first section, we provide a simple UI to fit a model to a possibly filtrated dataset. The second section contains a classification report with various classification metrics describing the fit of the model, such as *precision* and *recall* (see Section 2.4). The third and fourth section reports a ROC and precision-recall curve (see Section 2.4).

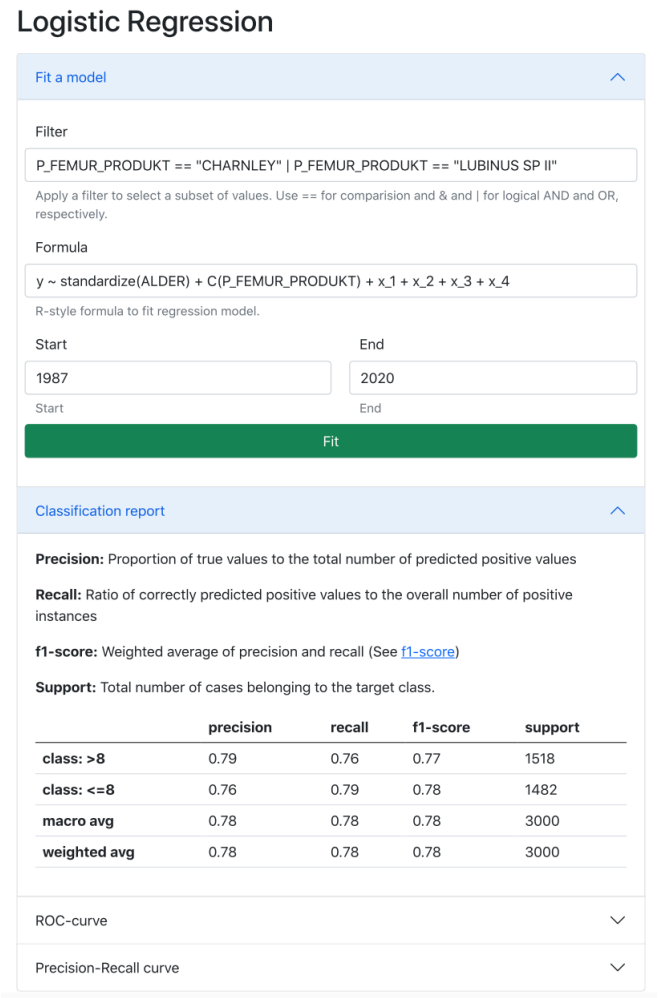


Figure 6.2: UI for the Logistic Regression component

Although the Logistic Regression procedure required some effort to incorporate into the front-end, we were able to integrate the KM procedure without encountering any major problems. During close-out, we discussed possible improvements and additions for the next iteration. For instance, we wanted to output the number of subjects next to group labels in the KM plot and allow the user to choose the observation period. Table 6.2 shows a list of methods that were implemented in iteration two.

Endpoints	**Description**
KM analysis (with UI)	KM analysis with extended risk tables and log-rank tests
Logistic Regression	Binary classification using Logistic Regression
Pairwise log-rank tests	Pairwise log-rank tests for $n > 2$ unique groups

Table 6.2: API endpoints implemented during the second iteration

6.4 Third Iteration

The third iteration took place in December and lasted approximately four weeks. The plan for this iteration was to integrate the other methods into the front-end and add methods to help explore and understand the data better.

We faced a few challenges while working on the front-end. First of all, we struggled to develop suitable UIs to gather the necessary input from the user. For example, it took us a while to come up with an appropriate UI to construct the boolean expression used to filter data in the KM method. Our first design accepted the input through a simple text box (see Figure 7.10). We eventually settled for a multi-step solution allowing the user to build the expression one step at a time. This solution is presented in Figure 6.3.

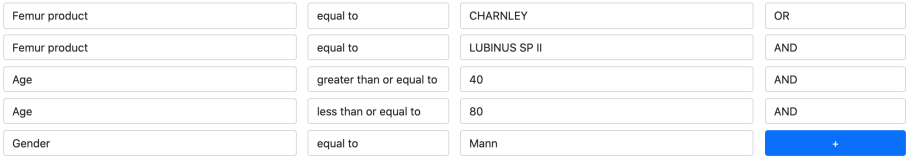


Figure 6.3: Filter mechanism used to select a specific sub-population in the dataset. We used this mechanism for the survival table.

To complement the KM, we added a *survival table* which aggregates survival data for a predefined period. This table shows the number of individuals who entered the study together with failures, censorings, and the total number of individuals at risk for each interval. Now that we had a front-end, we decided to add a contingency table to aid with data exploration. Contingency tables, also known as two-way tables, display the frequency of two or more variables and helps investigate the interrelationship between categorical variables. The contingency table was fairly straightforward to add, although we had to make some minor adjustments to the front-end to make the table interactive and searchable. Specifically, we included the Open-source library DataTables to enhance the table with sortable columns and real-time search. We deemed this enhancement necessary since the cross-tabulation can potentially return a lot of data depending on the user input.

After adding the contingency table, we shifted our attention towards the second major feature listed in the requirements specification from Section 4.1.2, namely Cox Regression. We spent quite a lot of time learning and experimenting with the Cox model before integrating it with the system. To make the integration of the Cox model more feasible, we broke the task into smaller and more manageable steps. We started by defining the possible input parameters of the model and their constraints. For example, one of the parameters of the Cox model is the *baseline estimation method* which must take on one of three possible values. We declared these constraints with JSON schemas in Pydantic - a data validation and parsing library well integrated with FastAPI. We will not go into further details about Pydantic other than to note that it helped validate user-input and 'standardize' the Web API responses. These schemas are used extensively on our Web API.

After specifying all the input parameters to the model, we extended the endpoint with functionality in an incremental manner. First, we integrated the filtering mechanism described in Section 6.3 to allow the user to fit a model to a user-specified dataset. Then, we concentrated on adding the essential elements from the Cox model, such as model summary and hazard ratios. Since Lifelines conveniently retains the parameters used to fit the model,

we could extract these properties afterward and pass them along in the response object. We encountered a couple of minor challenges during these efforts. One of them dealt with data conversion. In particular, Lifelines stores calculations with much higher precision than the response format (JSON) supports. Since, in most cases, three significant digits are sufficient, we safely converted the estimates to a 'less precise' data type. We also found that the availability of certain properties depended on the parameters supplied to the model. For example, when specifying *breslow* as the baseline estimation method, the log-likelihood property is the partial log-likelihood. However, when specifying *piecewise* or *spline*, the reported measure is the log-likelihood. Such nuances may seem minor, but they are potentially essential from a statisticians' point of view.

Although Lifelines is capable of producing plots of the survival function out-of-the-box, we wanted to allow the front-end team to enhance these plots with interactivity. Therefore, we extracted data from the figures and provided the necessary data to rebuild the figure on the front-end. We also provided the figures as Base64 strings as described in Section 6.3.

We implemented a simple UI on the front-end to fit and present results from a Cox regression. The UI and corresponding Web API request used to fit the model can be found in Appendix C.1. We concluded the iteration by reflecting on the progress and discussing potential improvements to the Cox procedure. In particular, we figured it would be better to split the routine across two sections: one for fitting the model and another for analyzing results. The idea was that by separating the two tasks, we would achieve separation of concern and reduce the complexity of the routines, both on the Web API and the front-end. Finally, we added an interaction plot that could be used to investigate possible interaction effects. We scheduled these alterations for the fourth iteration. Table 6.3 shows an overview of the methods implemented during the third iteration.

Endpoints	**Description**
Survival Table (with UI)	Table showing aggregated survival data
Contingency Table (with UI)	Table summarizing the frequency counts of two variables.
Interaction Plot (with UI)	Plot for investigating interaction effects
Cox Regression (with UI)	Procedure for performing a Cox regression analysis.

Table 6.3: API endpoints implemented during the third iteration

6.5 Fourth Iteration

We officially initiated the fourth iteration on the 25th of January, although we spent some time refactoring the codebase between the third and fourth iterations. In the kickoff meeting, we picked up the discussion from the close-out of the last iteration regarding separation of concern. We decided to separate the Cox routine into two stages - 'fitting' and analysis. We extended the existing endpoint to persist models to a database for later use. In practice, this is a two-step process that occurs *behind the scene*, not requiring any action on the user's part. First, we serialize the model to a stream of bytes in Python (Van Rossum and Drake Jr, 1995), and then we save it to a database with the help of an Object Relational Mapper known as SQLAlchemy (Bayer, 2012). This process further involves an SQLite database and a

database model (Hipp, 2020). We can consider the database model to represent a fitted Cox model that maps into a table in a relational database (SQLite). Further details and a depiction of the database model are available in Appendix C.

Mindful that Cox Regression outputs a lot of information, we designed the analysis page with the idea of separating functionality by concern. We accomplished separation using an UI component from Bootstrap that neatly compartmentalize the different aspects of Cox regression. This component has the added benefit of 'hiding' information from the user that is not relevant for the task at hand. Figure 6.4 displays the UI and the separation of functionality achieved for the Cox Regression procedure.

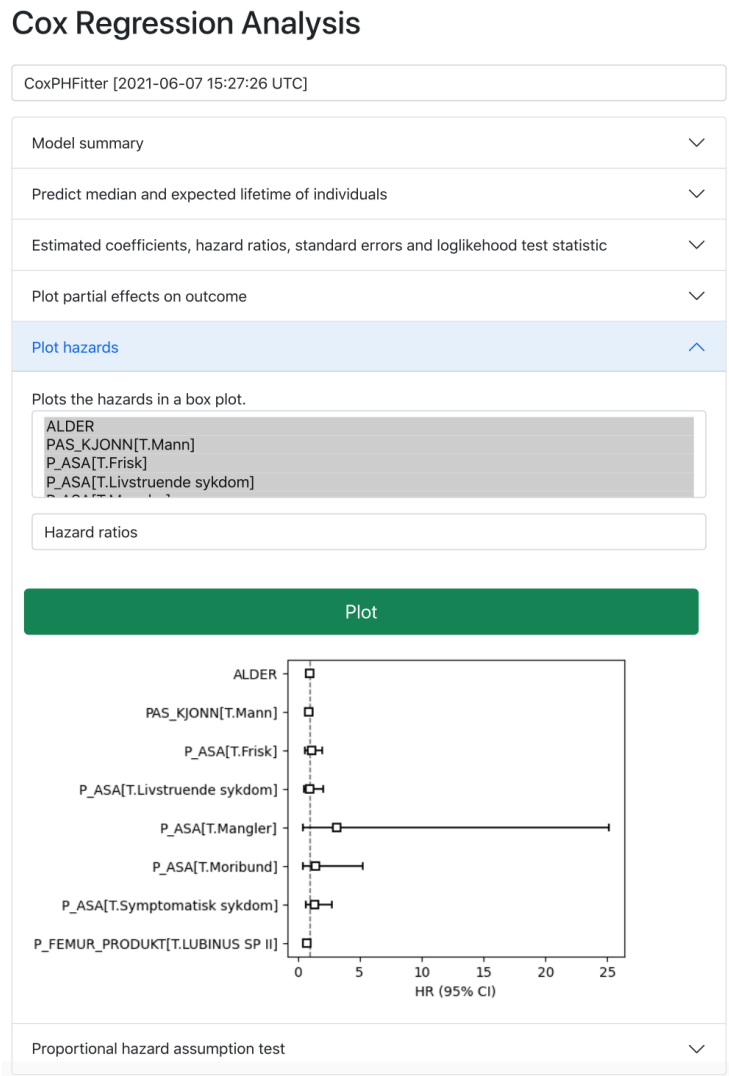


Figure 6.4: Compartmentalizion of the Cox Regression procedure.

We collaborated with Stolt-Nielsen and Farsund on complementary work related to visualization and HCI aspects of the implemented methods. We exported demographic data to Farsund and shared our methods such that Stolt-Nielsen could explore potential improvements to the user experience. During these meetings, we shared valuable comments and suggestions back and forth between the back-end and front-end team. This feedback led to several alterations and improvements to the design.

Midway through the iteration, we held two video conferences with a diverse group of

experts. Feedback from the experts helped us identify a series of inconsistencies in the UI and additional tasks that could improve the system. For instance, one expert pointed out that a plot was missing a label from an axis. Another expert suggested we include a routine for exploring the dataset more closely. Although the expert did not specify what such a routine would look like from the user's perspective, we were encouraged to include descriptions of variables and their distribution. We provide further details about the evaluation in Chapter 8.

After the evaluation, we dedicated some time to correct the issues that emerged from the expert review. We prioritized the small matters, such as inconsistent use of colors and missing labels. More demanding tasks such as the autocomplete functionality (see Chapter 8) were scheduled for future work. We did, however, take time to implement a method that provides a descriptive overview of the dataset. This method produces a LaTeX table with descriptive statistics for categorical and numerical variables in the dataset. The endpoint accepts as input an arbitrary number of variable names and generates two separate LaTeX tables — one for categorical variables and another for continuous variables. We interface with R through Python and use *reporttools* to generate the tables. The latter is a package for R specialized for generating descriptive tables in LaTeX (Rufibach, 2015). To interface with R from Python, we use a Python package known as rpy2 (rpy2, 2021). Figure 6.4 provides an example of what the table looks like for categorical variables. The implemented endpoints during the fourth iteration are available in Table 6.5.

Variable	**Levels**	**n**	%	∑%
Gender	Male	4987	49.9	49.9
	Female	5013	50.1	100.0
	all	10000	100.0	
ASA	Healthy	1632	16.3	16.3
	Asymptomatic condition	1648	16.5	32.8
	Symptomatic disease	1724	17.2	50.0
	Life-threatening disease	1641	16.4	66.5
	Moribund	1640	16.4	82.8
	Missing	1715	17.1	100.0
	all	10000	100.0	

Table 6.4: LaTeX table for categorical variables. The table is generated using the reporttools package in R (Rufibach, 2015).

Endpoints	**Description**
Cox Regression UI for analysis section	UI for the analysis section of the Cox Regression procedure
Descriptive statistics	Tables with descriptive statistics for categorical and continuous variables

Table 6.5: API endpoints implemented during the fourth iteration

Chapter 7

Artifact

The following section presents the DSR artifact produced after four development iterations of the DSDM. The artifact is twofold, consisting of two distinct but interrelated components - a Web API and a Web-based front-end application. The data mining methods are incorporated into the Web API and made accessible in a user-friendly UI on the front-end application. Figure 7.1 shows a high-level overview of the system architecture and the relationship between the two components.

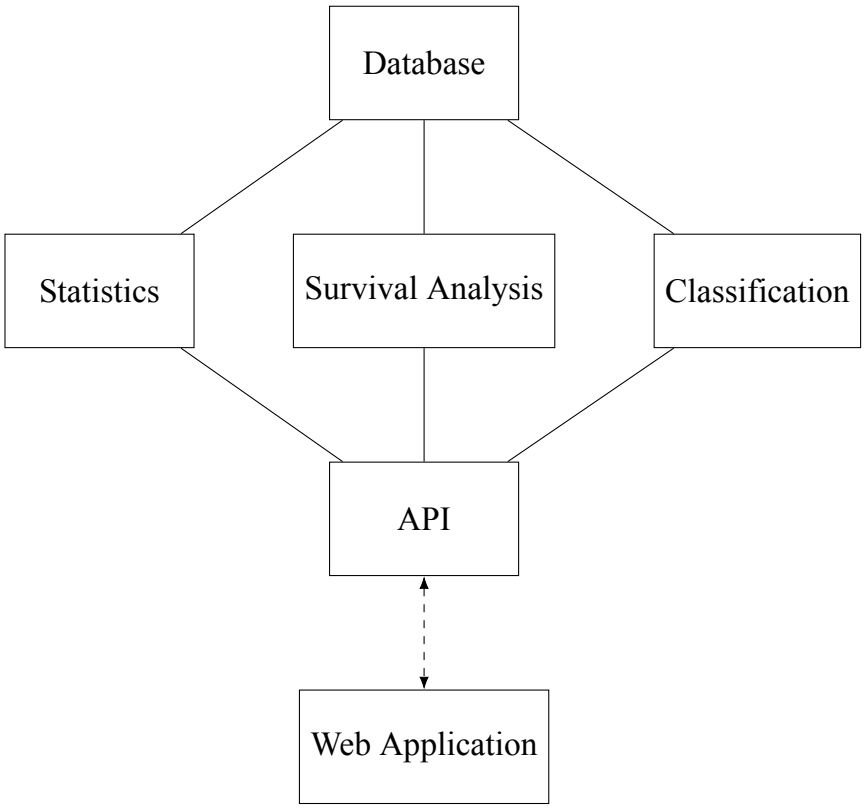


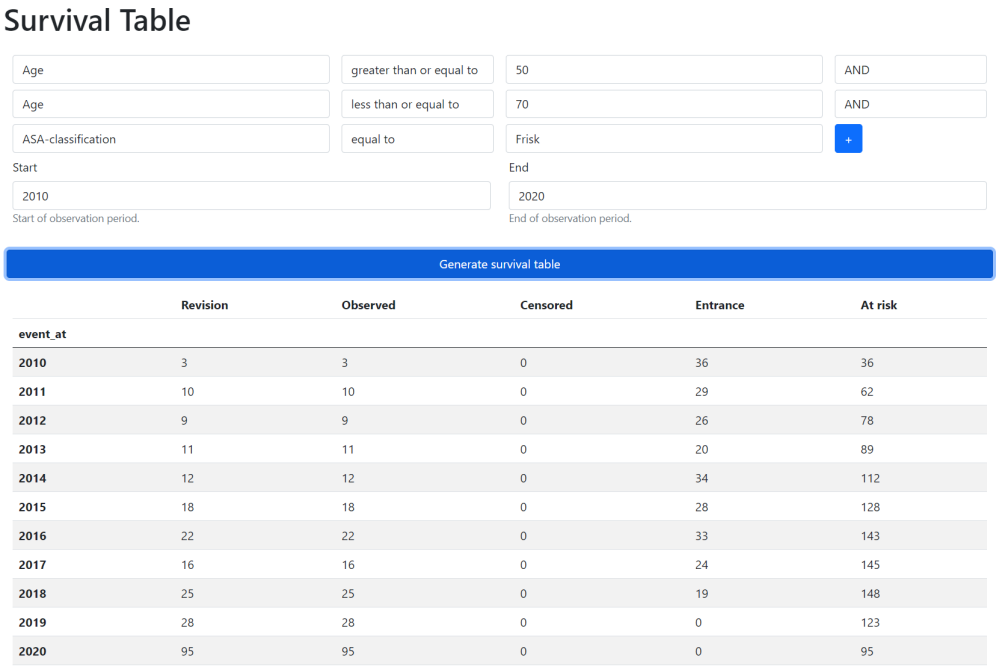
Figure 7.1: High-level overview of the system architecture. The API is the core component that acts as an interface between the data mining methods and the front-end application.

The remaining part of this chapter presents the data mining methods that are supported by the Web API. First, we show the descriptive data mining methods, namely survival tables, interaction plots, and contingency tables. The predictive methods, KM, Cox Regression and Logistic Regression are presented afterwards. In all of the following examples, data

displayed on the front-end is retrieved from the API using HTTP requests.

7.1 Survival Table

The survival table summarizes the survival experience for an entire population at each unique time point. In particular, survival tables record the number of events in total and the number of failures and censored events individually. The table also shows the number of individuals who entered the study and the total number at risk. Figure 7.2 shows a survival table for healthy patients between the age of 50 and 70 in the period 2010-2020. The column 'Removed' refers to the number of individuals removed from the study. 'Revision' refers to the number of individuals requiring revision surgery. 'Censored' refers to the number of individuals lost to follow-up. 'Entrance' refers to the number of individuals that entered the study at a given time. The total number of individuals at risk is shown in the last column.



event_at	Revision	Observed	Censored	Entrance	At risk
2010	3	3	0	36	36
2011	10	10	0	29	62
2012	9	9	0	26	78
2013	11	11	0	20	89
2014	12	12	0	34	112
2015	18	18	0	28	128
2016	22	22	0	33	143
2017	16	16	0	24	145
2018	25	25	0	19	148
2019	28	28	0	0	123
2020	95	95	0	0	95

Figure 7.2: Survival table showing the number of failures for each year. The user can specify any number of filters to focus on a sub-population.

7.2 Contingency Table

Contingency tables, also known as two-way tables, display the frequency of two or more variables. The table helps investigate the interrelationship between categorical variables. Figure 7.3 shows a contingency table from our prototype with implant type shown vertically and the implant survival shown horizontally. For example, the second furthest cell from the right shown in the top row displays the average age of patients during primary surgery for patients with the Lubinus SP II stem whose implant failed after 9 years. The selected sub-populations are patients with the Lubinus SP II and Charnley stem. Toggling the aggregation option displays frequency counts instead.

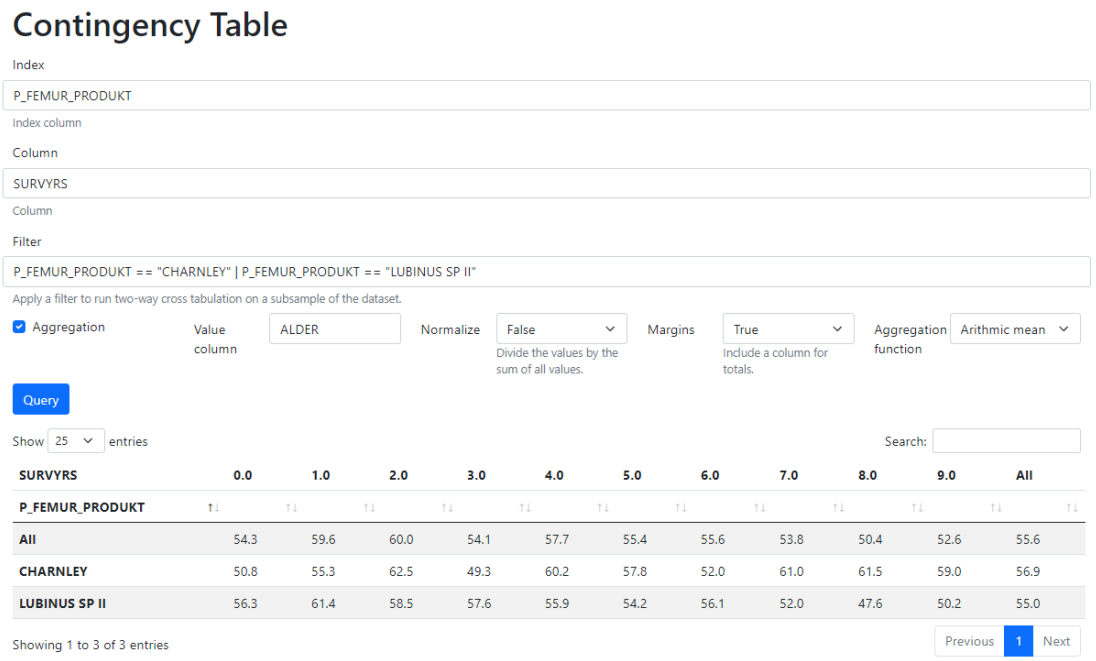


Figure 7.3: Contingency table comparing the average age at primary surgery for two implants distributed by duration of survival.

7.3 Kaplan-Meier (KM)

A simple user interface was created for the KM procedure. The user can select the period to study, desired CI and populations to compare. An arbitrary number of boolean expressions can be applied to restrict the number of curves to compute or select groups with certain characteristics, such as patients belonging to specific age groups or implants of a specific type. For example, *'(P_FEMUR_PRODUKT == "CHARNLEY" | P_FEMUR_PRODUKT == "LUBINUS SP II") & (ALDER >= 50 | ALDER <= 70)'*, selects patients with Charnley and Lubinus SP II implants who had surgery between the age of 50 and 70. Admittedly, such filters are not particularly user-friendly and can be hard to construct for non-programmers.

An improved alternate design was used for the survival table and interaction plot as shown in Figure 7.2. This design can be easily adapted for the remaining components of the system, but such improvements were not prioritized because the focus was on functionality rather than user experience which was left to the front-end team.

Figure 7.4 shows sample output from running a KM analysis in the system. The user-specified input is shown on the left, and the results of the analysis are displayed on the right. Results include an interactive KM plot with CI, extended risk tables for each group, and log-rank tests for assessing whether the groups are significantly different. Three different variations of the log-rank tests are available: Wilcoxon, Tarone-Ware, and Peto. These variations offer different weighting schemes that give more weight to certain time points.

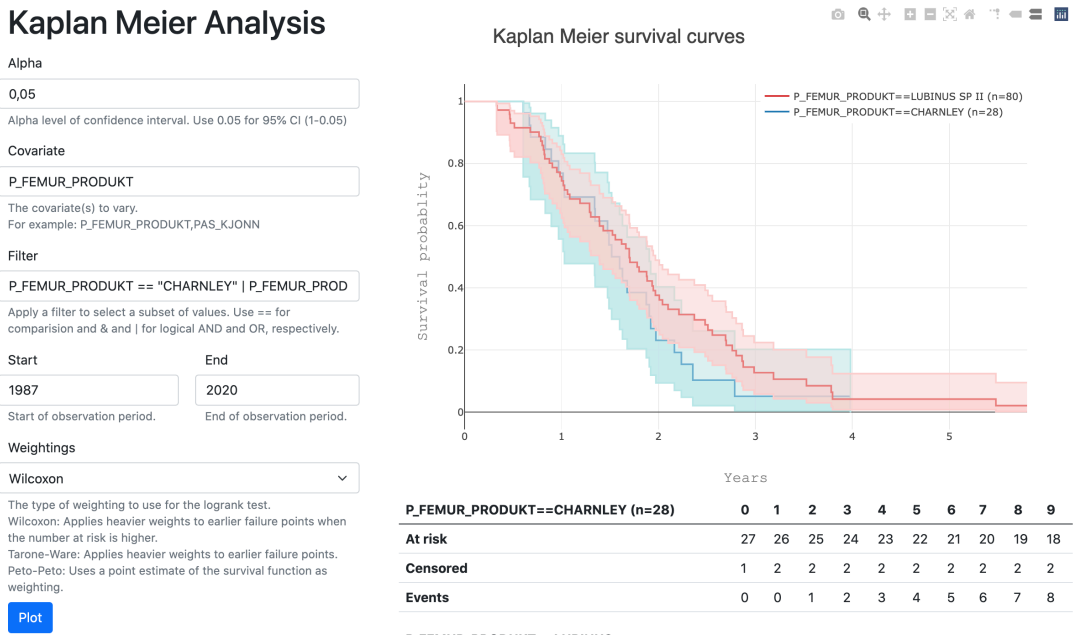


Figure 7.4: The UI for the KM procedure. The page shows a KM plot comparing the two femoral stems Lubinus SP II and Charnley. Based on the plot, it seems like the Charnley stem performs slighly better from 1.5 years onward. The data is synthetic and does not reflect the actual performance of the prosthesis and is used for illustration purposes only.

7.4 Cox Regression

Cox Regression is split across two sections: one for 'fitting' and the other for analysis and predictions. In the fitting section, we build Cox models based on input from the user and display details describing the model's fit, such as the concordance index score, likelihood-ratio test, and results from a proportional hazard assumption test. The output from the model is divided into several sections as shown in Figure 7.5. Models built in the 'fitting' section are persisted in a database and made available in the analysis section of the system.

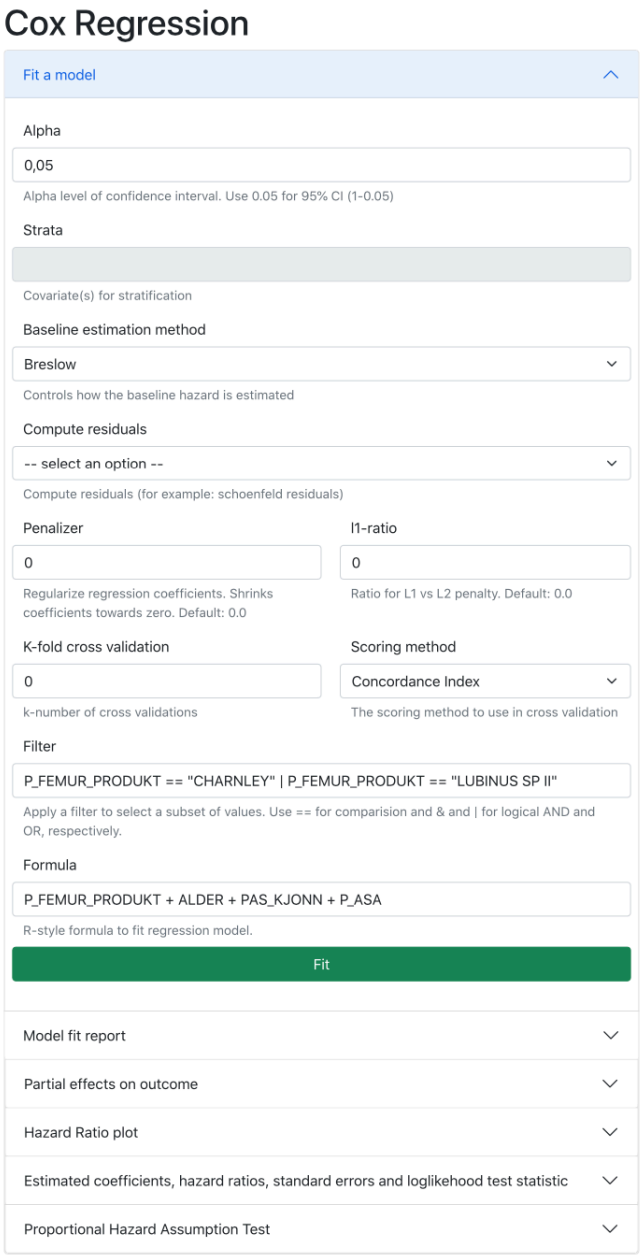
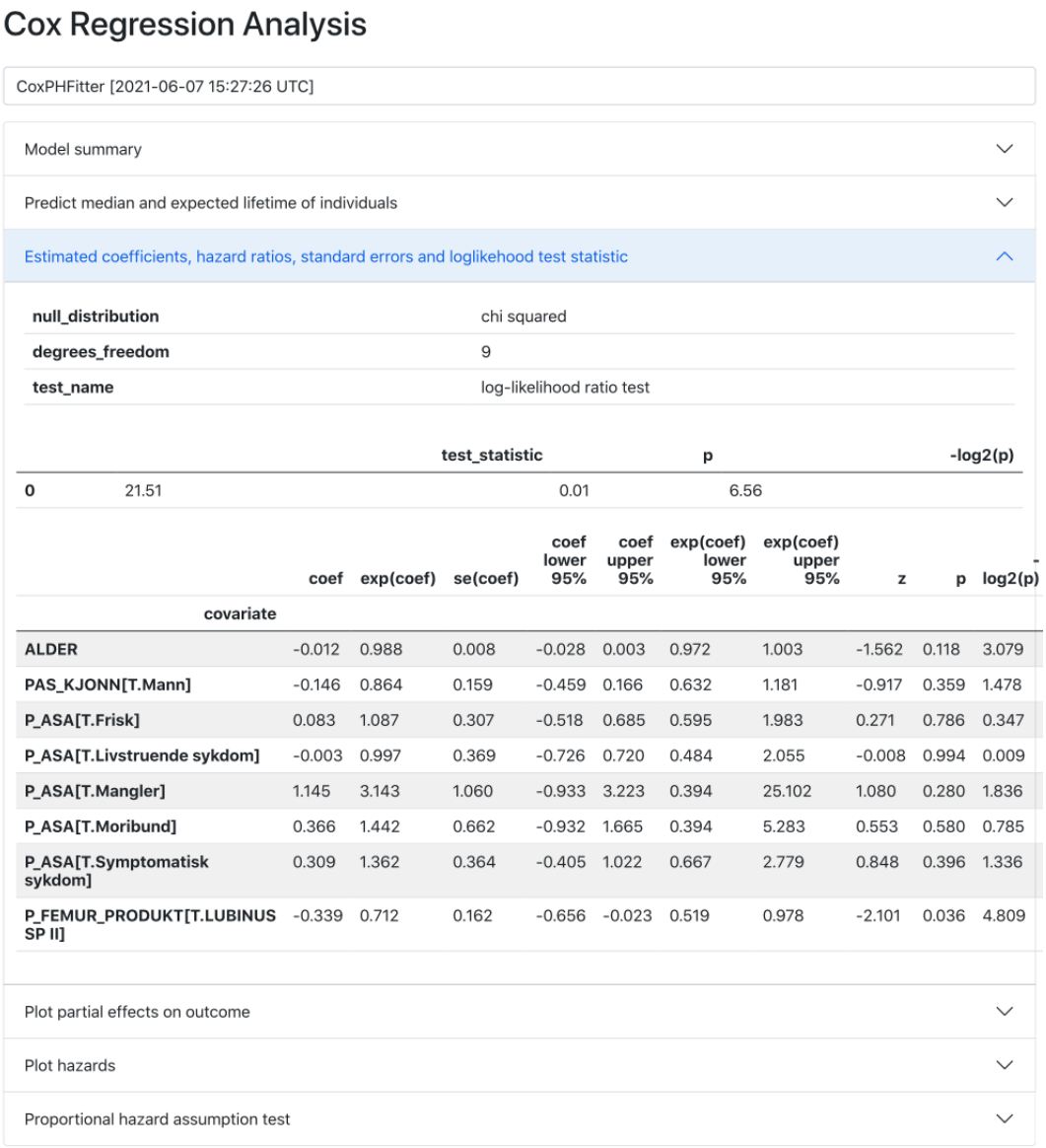


Figure 7.5: UI in the 'fitting' section of the Cox Regression procedure. Some of the available parameters have been hidden on purpose.

After fitting a model, users can navigate to the analysis section to further investigate the output of the model. Figure 7.6 displays a table with estimated regression coefficients, hazard ratios, 95% CIs, p-values among other things. The topmost table shows the result of a

likelihood-ratio test that compares the fitted model against a crude model with no covariates. The test statistic for the likelihood-ratio test is chi-squared under the null hypothesis.



Cox Regression Analysis

CoxPHFitter [2021-06-07 15:27:26 UTC]

Model summary

Predict median and expected lifetime of individuals

Estimated coefficients, hazard ratios, standard errors and loglikehood test statistic

null_distribution	chi squared
degrees_freedom	9
test_name	log-likelihood ratio test

	test_statistic	p	-log2(p)
0	21.51	0.01	6.56

covariate	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	z	p	log2(p)
ALDER	-0.012	0.988	0.008	-0.028	0.003	0.972	1.003	-1.562	0.118	3.079
PAS_KJONN[T.Mann]	-0.146	0.864	0.159	-0.459	0.166	0.632	1.181	-0.917	0.359	1.478
P_ASA[T.Frisk]	0.083	1.087	0.307	-0.518	0.685	0.595	1.983	0.271	0.786	0.347
P_ASA[T.Livstruende sykdom]	-0.003	0.997	0.369	-0.726	0.720	0.484	2.055	-0.008	0.994	0.009
P_ASA[T.Mangler]	1.145	3.143	1.060	-0.933	3.223	0.394	25.102	1.080	0.280	1.836
P_ASA[T.Moribund]	0.366	1.442	0.662	-0.932	1.665	0.394	5.283	0.553	0.580	0.785
P_ASA[T.Symptomatisk sykdom]	0.309	1.362	0.364	-0.405	1.022	0.667	2.779	0.848	0.396	1.336
P_FEMUR_PRODUKT[T.LUBINUS SP II]	-0.339	0.712	0.162	-0.656	-0.023	0.519	0.978	-2.101	0.036	4.809

Plot partial effects on outcome

Plot hazards

Proportional hazard assumption test

Figure 7.6: UI from the analysis section of the Cox Regression component. The figure shows a table with hazard ratios, regression coefficients, c-index among other measures.

The *Model summary* section in Figure 7.7 shows a summary table describing the model that was fit and the model's goodness of fit. For example, information such as the total number of observations, number of observed events, baseline estimation method, cross-validation scores, c-index, partial AIC, and more can be found in the table.

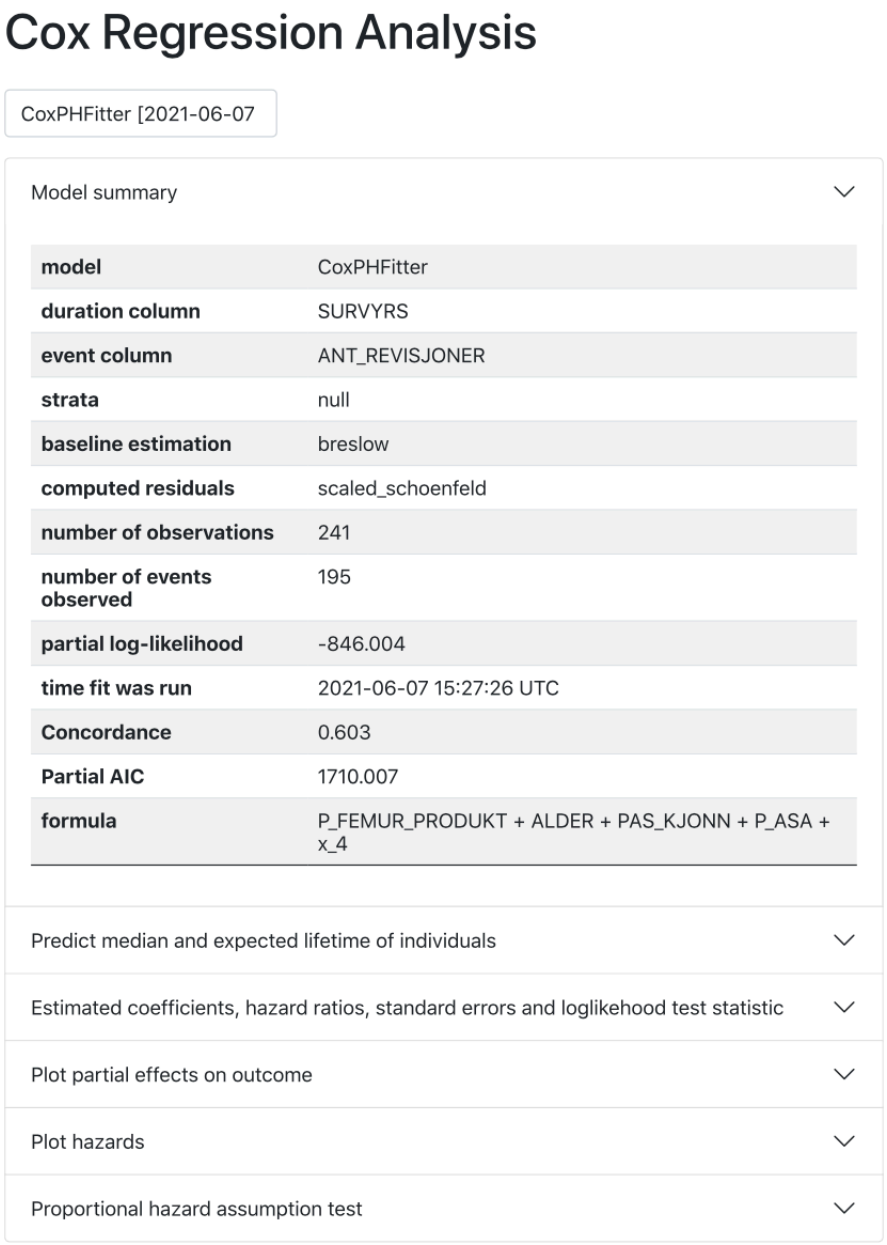


Figure 7.7: UI from the analysis section of the Cox Regression component. The figure shows a summary of the fitted model

In the *Plot partial effects on outcome* section, users can compare the influence of individual covariates on the survival outcome. Figure 7.8 displays the UI that was created for this part.

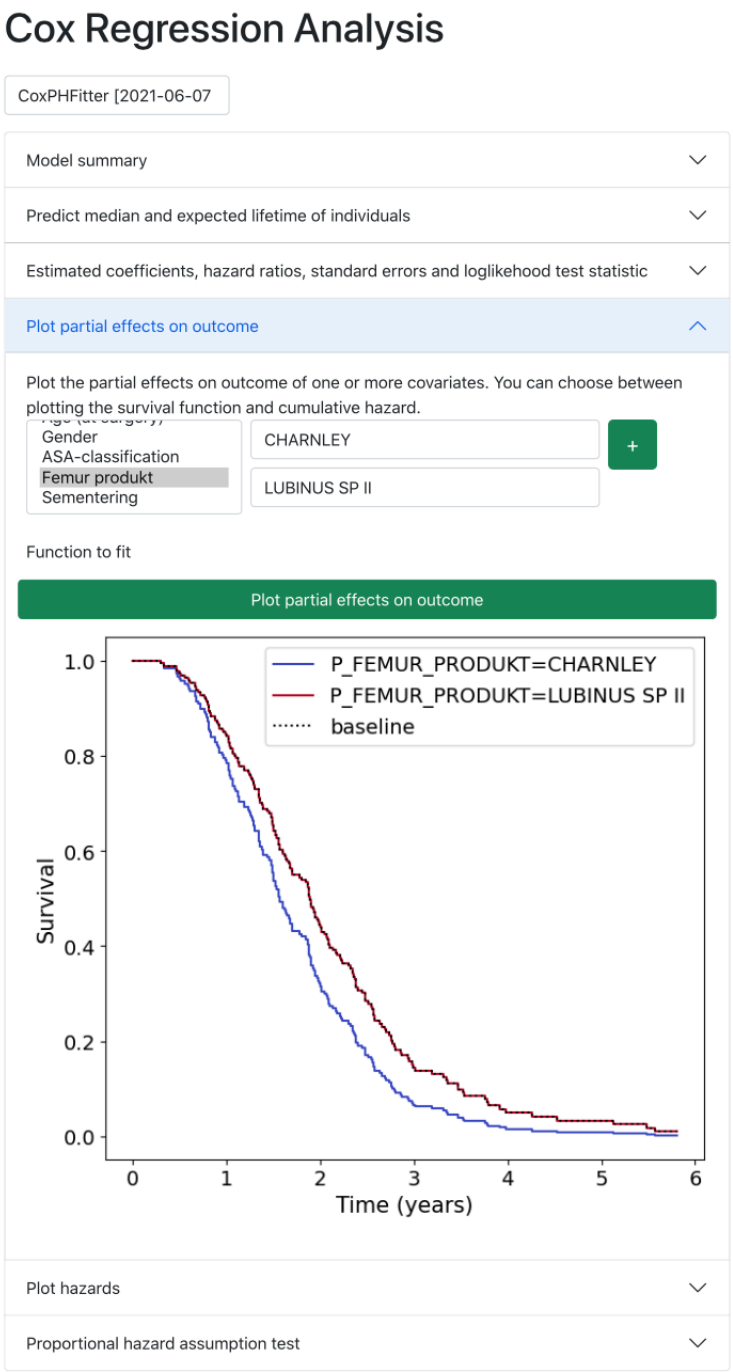


Figure 7.8: UI from the analysis section of the Cox Regression component. The figure shows the survival survival curves for the Charnley and Lubinus SP II stem.

Results from the proportional hazard assumption test is found in the bottom section (see Figure 7.9). If the assumption is violated, users are informed about the violating factor and advised on actions that can help satisfy the assumption. Users in need of further assessment of the proportional hazard assumption can compute the scaled Schoenfeld residuals of the

covariates. This produces a series of plots that become available in the analysis section, which is useful for visually inspecting and assessing the proportional hazard assumption.

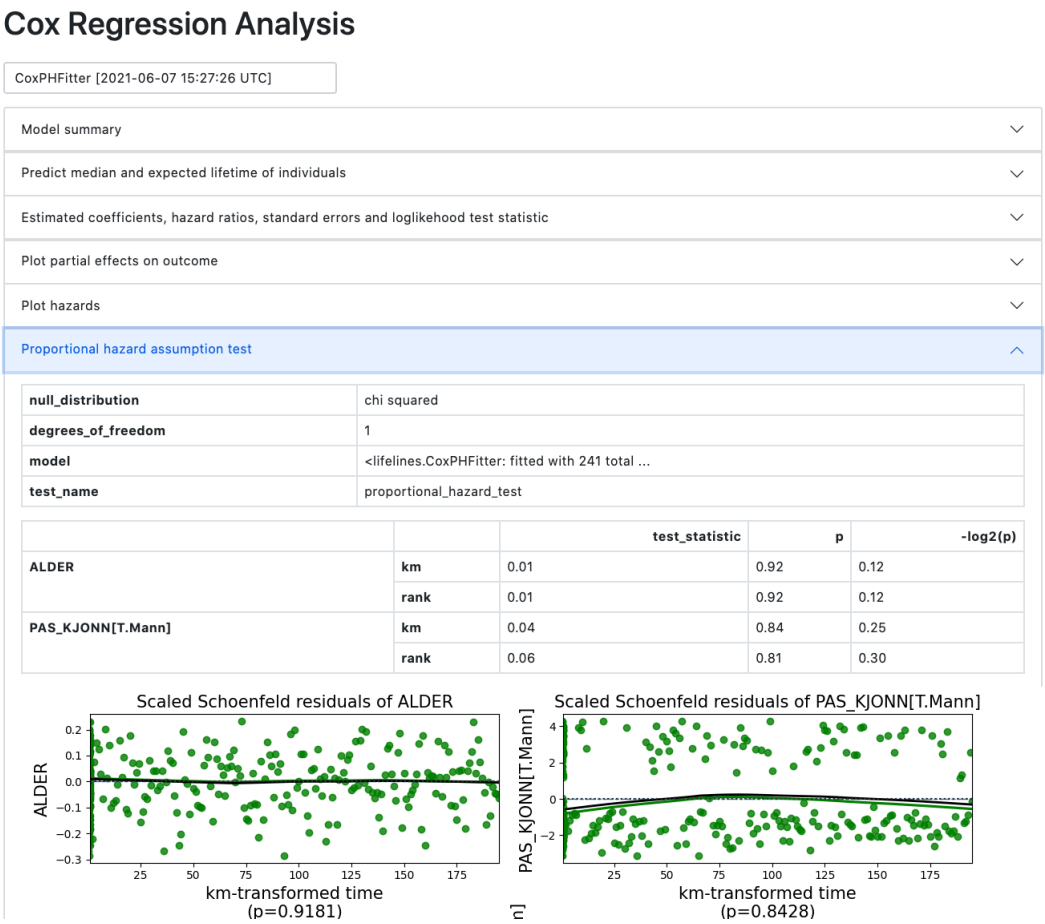


Figure 7.9: UI from the analysis section of the Cox Regression component. The figure shows results from the proportional hazard assumption test.

7.5 Logistic Regression

This page uses a Logistic Regression model from Scikit-learn to predict the longevity of an implant as a binary outcome problem. The independent variable and target classes are fixed and cannot be altered by the user. Data is split into a training and test dataset using the 70/30 rule, where the larger portion is reserved for training and the remaining 30% is used for testing. The dependent variable is the time-until revision, and the target classes are less than 8 years and larger than or equal to 8 years.

Figure 7.10 shows a classification report from the UI after fitting a Logistic Regression model to synthetic test data. The user-supplied formula specifies the model to construct, and the filter selects training data for the model. After fitting the model, the goodness of fit can be assessed using a precision-recall curve, ROC curve, and various classification metrics such as the *f1-score*, *precision*, and *recall*. Figures 7.11 and 7.12 display the ROC and precision-recall curve as they are presented in the UI. This component is merely intended as proof-of-concept and, for that reason, does not allow users to perform predictions on out-of sample observations.

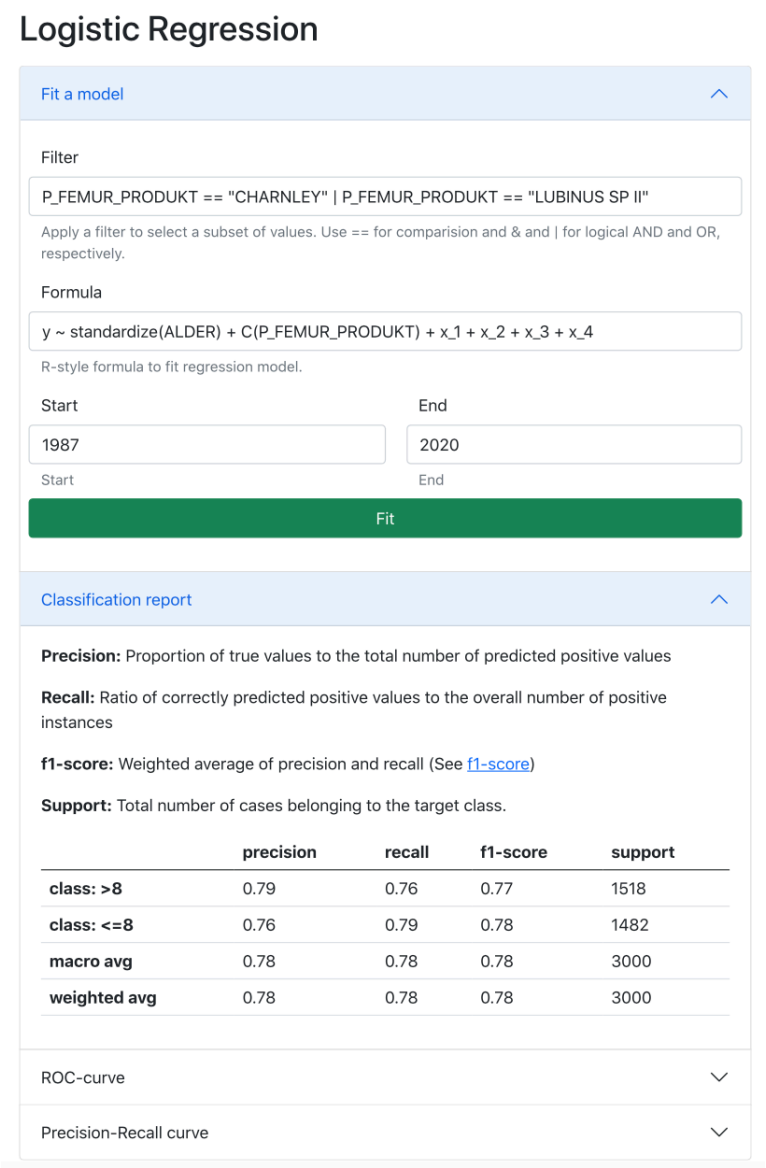


Figure 7.10: UI for the Logistic Regression component

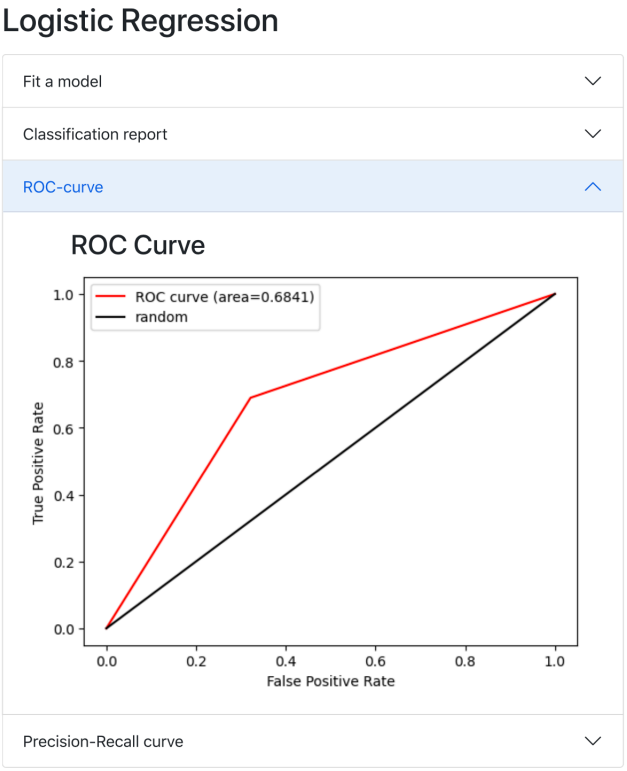


Figure 7.11: ROC curve from the Logistic Regression component

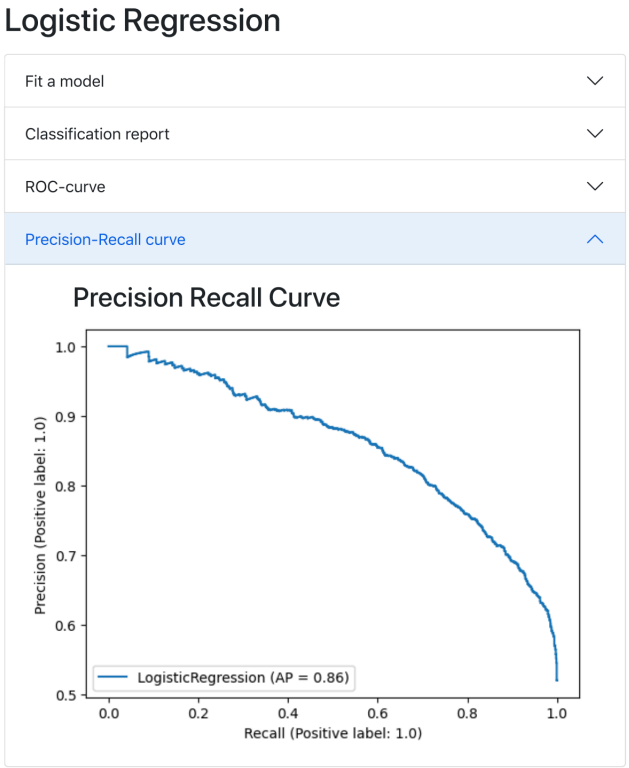


Figure 7.12: Precision-recall curve from the Logistic Regression component

7.6 Descriptive Statistics

Based on feedback from the evaluation, we created a routine that generates descriptive statistics of the dataset. The procedure is capable of generating statistics for both categorical and numerical variables. Figure 7.1 and 5.2 shows the LaTeX tables produced by this routine. At the moment, these tables are not available on the front end. However, we plan to do this in a future iteration, possibly using LaTeX.js (LaTeX.js, 2021). More details about how we implemented this procedure is available in Section 6.5.

Variable	**Levels**	**n**	%	∑%
Gender	Male	4987	49.9	49.9
	Female	5013	50.1	100.0
	all	10000	100.0	
ASA	Healthy	1632	16.3	16.3
	Asymptomatic condition	1648	16.5	32.8
	Symptomatic disease	1724	17.2	50.0
	Life-threatening disease	1641	16.4	66.5
	Moribund	1640	16.4	82.8
	Missing	1715	17.1	100.0
	all	10000	100.0	
Surgical Approach	Anterior (Smith-Petersen)	1602	16.0	16.0
	Anterolateral	1718	17.2	33.2
	Lateral	1685	16.9	50.0
	Posterolateral	1643	16.4	66.5
	Other	1663	16.6	83.1
	Missing	1689	16.9	100.0
	all	10000	100.0	
Stem Material	Steel	3356	33.6	33.6
	Titanium	3333	33.3	66.9
	Cobalt-chrome	3311	33.1	100.0
	all	10000	100.0	

Table 7.1: LaTeX table for categorical variables. The table is generated using the reporttools package in R (Rufibach, 2015).

7.7 Interaction Plot

An interaction plot helps us determine the presence of possible interaction effects. In this plot, the mean of the dependent variable is shown on the y-axis and the idependent vari-

able is displayed on the x-axis. The traces correspond to categorical variables with possible interaction effects. Parallel traces indicate no interaction effect, while crossing traces may indicate an interaction. Figure 7.13 shows how survival span fluctuates with a patient's age at surgery. The ASA category 'Moribund' is considered a possible interaction effect. In spite of minor differences, we see that the two traces follow each other closely.

Figure 7.13: Interaction plot showing the mean survival span with age at primary surgery on the x-axis. The traces show the response of healthy (red) and moribund (blue) patients.

Chapter 8

Evaluation

We assessed the utility and usability of the prototype by presenting and demonstrating the prototype to three groups of reviewers. After each demonstration, we asked the experts to fill out a SUS questionnaire. We also asked them what they thought about the starting pages and the ability to save sessions for later use. We present the response to the SUS questionnaire and the follow-up questions in Section 8.4.

The first group consisted of two biomedical researchers, one of whom has extensive domain knowledge in orthopedics. The second group included three experts experienced with building data mining and HCI solutions for the medical domain. For the third evaluation, we interviewed a medical doctor (general practitioner). Table 8.1 shows the participants involved in the evaluation.

Participant	Gender	Age	Education	Profession
P1	Male	31	Medical Degree	General Practitioner
P2	Male	25+	Master's degree	IT professional
P3	Male	25+	Master's degree	IT professional
P4	Male	45	Ph.D	Researcher (Biomedical Engineer)
P5	Male	25+	Master's degree	IT professional
P6	Male	41	Ph.D	Researcher (Biomedical Engineer)

Table 8.1: Description of the participants that took part in the expert evaluation.

For the digital meetings, we arranged two video conferences segmented into three parts: (1) prototype demonstration, (2) discussion and questions, (3) feedback. The sessions lasted approximately one and a half hours each and ended with a request to fill out a standard SUS survey. We demonstrated the most digestible features first and gradually went on to the more advanced features. We did this in order to not overwhelm the participants and keep them attentive throughout the whole session.

The features were presented in the following order: survival table, contingency table, and Cox Regression. The first two meetings took place digitally due to the ongoing COVID-19 pandemic. In addition, because of the rather technical setup of the system, we chose to show two short video presentations and demonstrate the use of the system instead of letting them

explore online. The third session with the general practitioner (GP) took place in person and lasted approximately half an hour.

8.1 Session one: domain experts

The reviewers were intrigued by the survival and contingency table but argued that having to spell out variable names would be cumbersome and difficult for most users, especially 'new' users. One expert suggested that autocomplete functionality for the variable names would likely result in a more pleasant user experience.

Both reviewers expressed interest in the mechanism for managing data used in the analyses. In particular, they wondered whether we queried data from an external SQL database upon each request to the Web API. We explained that in the current solution, data was stored locally on disk and held in memory while the Web API is running. However, we further explained that this approach was only for convenience and that future iterations would store the dataset in a database. The discussion led to the identification of at least three benefits. First, databases are always up to date. Second, databases offer an extra layer of security for the data. Lastly, query languages bundled with database management systems such as Microsoft SQL offer a more expressive and powerful language to query the data (SQL). One expert pointed out that always having up-to-date data would be an advantage of our system compared to more complete statistical software solutions such as SPSS or STATA.

Afterward, we demonstrated the Cox Regression procedure by walking them through the process and explaining details along the way. The experts expressed interest in the feature, suggesting it might be interesting for the register. One of the experts raised concerns about the feature's usefulness and what benefits it provides to its target users. In response, we stressed that other more feature-rich software packages such as SPSS or SAS offer the same procedure. However, we argued that users would probably save time and find our system straightforward to use than SPSS and SAS. The expert fully agreed and claimed our system seemed much more tailored to its target audience, taking them through the process in a more "slicker" way than SPSS. We learned from one of the experts that the current method of reviewing data at the NAR is a manual process. The data is manually exported from a database and handed over to statisticians for further analysis. Results and findings are compiled and published in annual reports. The expert further suggested that our system may be useful to perform "quick checks" on a weekly or monthly basis. He also pointed out that the prototype may "easily" extend to function as a monitoring system to detect when a group (prosthesis) falls below a certain threshold. The ability to save the models and revisit them later was well received, with one expert labeling it a "nice feature".

8.2 Session two: IT experts

One of the IT experts proclaimed that the system seemed 'cool' and bore a resemblance "Microdata". Microdata is a system for analyzing registry data developed by the Norwegian Centre for Research Data and Statistics Norway.

The HCI experts proposed that we properly align column headers in the tables and use margins between UI elements more cautiously in our design. Specifically, the headers were misaligned, and some elements appeared too closely together and were difficult to tell apart.

Furthermore, one expert made us aware that an axis in one of our plots was missing a label. Another HCI experts drew attention to the inconsistent use of colors in the system. The expert argued that such inconsistencies might confuse the users and require them to exert more cognitive effort than necessary to use the system.

The mix of language (Norwegian and English) was highlighted as a potential usability issue by the experts. We agreed but pointed out that the interface is written in English and that the inconsistencies appear because the variable names are coded in Norwegian. However, the point still stands, and we believe that a mechanism for renaming variables could be suitable in a potential preprocessing routine of the system.

The experts argued that most usability issues seemed minor but urged us to fix them because they can potential have a substantial impact on the user experience. We concluded the meetings with a short debrief requesting the experts to fill out the SUS questionnaire.

We forwarded the criticism directed at the user interface and other HCI aspects to Stolt-Nielsen in case any of the errors had manifested themselves in her HCI work. Similarly, we shared the comments regarding consistent use of colors and the labeling of axes with citeauthorarle which was working on visualizations.

8.3 Session three: General Practitioner (GP)

We began the session by assessing the GP's experience and knowledge about survival analysis and registry-based studies. The GP was familiar with both KM and Cox Regression from the literature, but was not well-versed in how to conduct such analysis. The participant was also aware of registry-based studies, but not about the specific efforts of the NAR. To establish a better starting point, we gave the participant a short explanation of KM and Cox Regression while pointing out that the purpose of the system was to explore risk factors for early failure of implants. We further informed the participant about registry-based studies and the importance of detecting inferior implants.

In response to the survival table, the GP commented that it seemed like an ”occasionally useful feature” that felt ”somewhat highly specialized” towards its target group. He appreciated the ability to filter data and thought the output was ”well presented,” but proposed we provide more context about its usefulness. We explained that survival tables are used to learn about the survival experience of one or more groups and that they are typically part of a more in-depth analysis. The GP concurred, admitting it seemed like a valid use case.

The GP appreciated the contingency table claiming it seemed like a 'great idea,' but encouraged us to make it more user-friendly and less 'daunting'. In particular, the GP pointed out that the filtering seemed challenging to comprehend and intangible to all but the most advanced users. He further proposed that we reuse the survival table's filtering component, which felt ”much nicer”. Like the experts from the earlier evaluation, the GP noted that the variable names seemed difficult to remember. To improve the user experience, he encouraged us to provide a list of variable names somewhere on the page.

The participant suggested that the survival table shown earlier would probably complement the KM analysis well and that we should consider ”fusing” them together on one page. He further argued that the survival table seemed to make more sense together with the KM procedure than as a standalone feature.

The GP found the model creation process interesting albeit a bit ”overwhelming” with all

the available ”options” (model parameters). He maintained that the procedure would probably make sense from an expert perspective and liked the descriptions next to each option. The critique of the contingency table regarding the diffuse variable names was reiterated and suggested as a potential improvement. Moving on to the results, the GP noted that the page seemed ”very clear” and ”nicely laid out”. In particular, he enjoyed the ability to perform predictions of implant survival based on user-defined criteria, although he wondered about the reliability of the predictions. We informed that the concordance index score measures the model's ability to provide reliable rankings or, put differently, that individual risk assessments are relative to event times. The plot showing the effects of covariates on outcome was well received and considered ”fun to play with”. He suggested we show somewhere what the *baseline hazard* represents. We clarified that the baseline hazard has no meaningful real-world interpretation and represents the ”average” individual at each interval. We argued that for a dichotomous variable such as ”smoker”/”non-smoker,” it doesn't make sense to talk about ”half a smoker.” You either are a smoker, or not. However, from a mathematical standpoint, the ”average” subject has a sensible interpretation. The GP agreed with the latter assessment.

We moved our attention towards the summary table, briefly explaining the information reported in the table, such as the estimated regression coefficients, hazard ratios, standard errors, p-values, CIs, and the loglikelihood test. The GP argued that although there was ”a lot of (information) to take in, everything appeared quite tidy”. He proceeded to say that the inclusion of p-values was a ”good idea” and ”important” in research for hypothesis testing. He proposed we include a short description of what the loglikelihood test compares and what it tells us.

Afterward, we showed him the residual plots used to test the proportional hazard assumption. In response, the GP proclaimed that ”visuals (visualizations) are always good.”. We concluded the meeting with the SUS questionnaire and a short debrief asking him what he thought about the system and the choice of data mining tasks. The GP argued that the prototype seemed to have ”a lot of potential” and appeared to be ”easy to use”. He further argued that the choice of data mining tasks is ”probably good”, but that he lacked knowledge about the field to give us a definitive answer.

8.4 SUS questionnaire and follow-up questions

The calculated SUS scores was generally high and put the prototype well within the acceptance range. All scores were above 75 (average 84.5) which corresponds to a grade of C or greater. 2 out of 6 participants gave a score of 90 or greater, situating the prototype within the 'best imaginable' category. Three of the participants put the prototype within the 'excellent' category corresponding to a grade between 80 and 90. Figure 8.1 shows the calculated SUS score for each participants.

The lowest score of 77.5 was given by the GP. In contrast to the other evaluations, the review with the GP was conducted in person and with the opportunity to try the system in practise. Therefore, it is possible that the GP may have come across other usability issues that were not easy to spot during a video conference. Another possibly is that participants are more 'generous' in remote evaluations than they otherwise would have been in-person. Nonetheless, the system seems to have been well-received and the feedback from the experts

was much informative.

We presented the participants with two follow-up questions. (1) Their opinion regarding the ability to save sessions (analyses) for later use. (2) Whether they found the starting pages welcoming or not. The responses to these question are summarized in Figures 8.2 and 8.3.

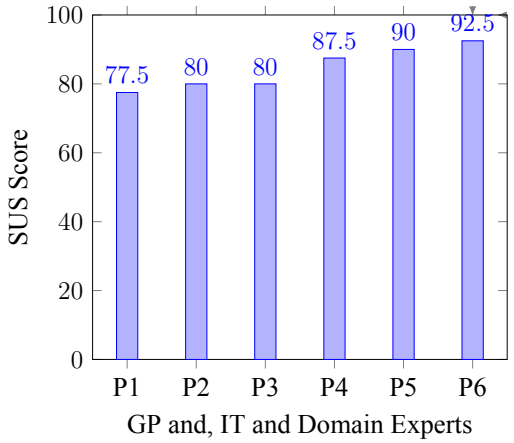


Figure 8.1: Calculated SUS scores (average score: 84.5)

.

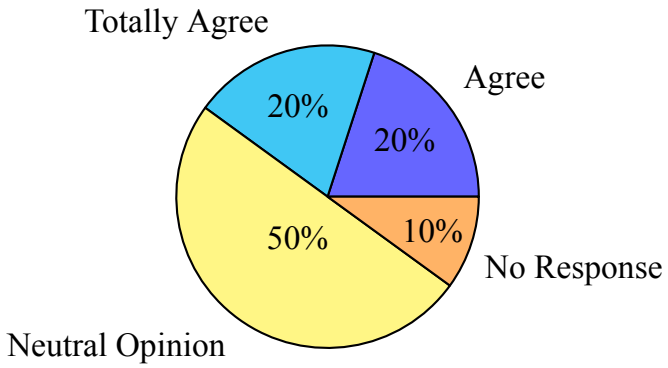


Figure 8.2: Response to the ability to save sessions for later use in prototype

.

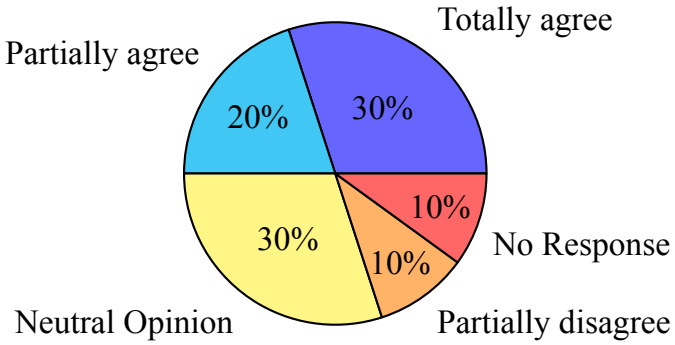


Figure 8.3: Response to whether the participants found the pages welcoming

.

Chapter 9

Discussion

The prototype benefited from feedback from a diverse group of experts, including a medical practitioner, two domain experts, and multiple HCI experts. In general, the experts found the system easy to use, albeit a bit ”overwhelming” at times, but functionality seemed appealing. Discussions with domain experts helped identify and substantiate several key benefits that the artifact exhibits or can attain in future iterations. First and foremost, we have tailored the system to the target group, and the survival analysis methods offered are trusted and actively used by the registry today. As one expert put it: ”it leads them through the process in a much slicker way”. As a result, the system could be well suited as a tool for quick analyses or checks in favor of fully-fledged statistical analysis software requiring more time and resources to manage.

In addition, the Web-enabled API architecture allows other, possibly heterogeneous systems to leverage our data mining component. One possible extension would be an implant monitoring system that alarms users when an implant goes below a certain performance threshold. Secondly, the system can easily be adapted to pull data directly from a database such that analyses are always up-to-date. Thirdly, we can extend the system with more models from Lifelines or Scikit-learn. In this regard, one might want to consider methods from Iden (2020) and Longberg (2018). Although models may vary in complexity, both Lifelines and Scikit-learn follow a consistent and straightforward interface abstracting away implementation details. Therefore, extending the API with methods from these libraries should be feasible. Other machine learning libraries may require more effort.

Lastly, based on feedback from expert evaluation, we argue that the developed artifact lowers the entry for performing survival analysis that otherwise requires some experience in programming languages or statistical software such as R, SPSS, or STATA. Contemporary work by Stolt-Nielsen addresses some of the shortcomes related to the HCI aspect of our prototype. In particular, Stolt-Nielsen propose a mechanism for filtering data in a less strenuous manner. We propose our own solution shown in Figure 7.2 and 7.13. Further work should involve target users and investigate the practicality of both of these solutions. Readers interested in enhancements pertaining to HCI aspects of our work should turn to Stolt-Nielsen (2021) and Farsund (2021) for inspiration.

9.1 Answering Research Questions

RQ 1: What are the qualities and characteristics of an outcome analysis tool for THA?

It seems like the objective of registry-based research is always related to answering concrete questions such as the longevity of an implant or risk assessment of new implants that enter the market. Although many methods could help answer these questions, the registry's role as a national resource means they must be highly diligent and scrupulous about their work. Hence their choice of well-established and widespread methods such as KM and Cox Regression. Various clinical fields highly regard these methods due to the transparency of analysis and the possibility to interpret results. Also, these methods come with measures of accuracy and significance, such as the c-index and log-rank test. It is therefore conceivable that outcome analysis tools for THA should exhibit all of these qualities and characteristics - interpretability, precision, and significance.

We identified some desirable characteristics during evaluation as well. Firstly, always up-to-date data was much valued property by the experts. Secondly, a tailored user experience seems to be a much appealing feature (Section 8.1). A tailored user experience is only attainable through well-established HCI principles which became apparent during evaluation with IT and HCI experts. Hence, an analysis tool for THA should probably strive to satisfy all of the above.

There are other methods to consider, such as Logistic Regression and other survival models besides those presented in this thesis. However, such models should be validated and scrutinized, preferably using a more realistic data sample.

RQ 2: What data mining methods are useful for outcome analysis in THA?

This research question was initially answered by consulting the literature and staff at the registry in Bergen. According to the typical tasks they conduct on the data, their experience is to use KM and Cox regression analysis to predict the longevity of implants and assess risk factors for THA. They are often presenting demographic data in the form of graphical visualization or summary tables published in annual reports and made public on their website. The literature also mentions Linear Regression, Logistic Regression, MLP, and PCA (Berge, 2019; Longberg, 2018; Iden, 2020).

Methods for inspecting the completeness of data with visualizations have also been investigated (Berge, 2019). All of these methods, including those presented in this thesis have shown potential and could prove useful for outcome analysis in THA.

In this project, we have started with methods that are already well-grounded in the domain. In the current form, the registry carries these methods out with the help of statisticians, so the idea of this project was to make them more approachable and easier to use through a proper HCI interface. We argue that such a rich national resource of data could be better utilized by opening the user-base to include all those interested in exploring the data and deriving their own hypothesis to answer research questions. Examples of such users would be physicians or end-researchers.

RQ 3: Can KDD lower the barrier of entry and allow medical staff to analyze hip arthroplasty data without the need for a statistical background?

At this point, it is not possible to answer this question with full confidence due to the lack of a more comprehensive evaluation. Further evaluation involving medical staff would help determine the value of the tool for clinical practise. However, such evaluation should be performed under circumstances that allow them to explore and try the tool in practise. The answer could also be guided by the result of the evaluation which has shown a great acceptance of the choice of methods. In addition, they were found to be reasonably easy to use and straightforward although it was clear that more knowledge of the domain and methods would be beneficial.

The choice of methods is grounded in the literature with which both clinical experts and biomedical engineers are familiar with, so it could be expected that they will be comfortable with the implemented data mining procedures (Section 8.1).

The evaluation has also shown that there is a place to add additional methods although nobody specified what could be added or what would be their particular wish. Due to the combined efforts of the back-end and fron-tend development, the evaluators have seen an interactive web application that lends itself to two application domains (Hip and Knee). Moreover, this application makes it easy to choose different data mining methods and analyse results.

Visualizations and solutions that were added to the front-end made it possible to further explore data in terms of time periods, region of country, and interactive visualizations. The work of the front-end team can be found in (Farsund, 2021) and (Stolt-Nielsen, 2021). Such experience, is certainly more user-friendly than diving into a programming environment that back-end developers are used to. There is a great potential to make the system more approachable and easy to use. However, further development and experience is required to fully tailor the system for real users and different user-groups (medical doctors/biomedical researchers). Future work will have to explore this direction.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

This thesis applied DSR to produce a novel artifact in the form of a data mining tool targeted at researchers working within the arthroplasty domain. DSR has ensured the rigor and relevance of the research and helped evaluate the utility of the artifact against potential target users and experts. After a meeting with the NAR, we established the requirements and developed the system using the DSDM. Development was carried out iteratively in small increments, and we prioritized requirements using the MoSCoW technique from DSDM. We allocated approximately 2-3 weeks for each iteration and held frequent meetings to discuss potential problems or improvements.

Our contribution is a fully functional prototype for exploring arthroplasty data and assessing hip implant performance. We implemented the prototype as a Web API and modeled the data mining methods after the KDD process. Among implemented methods are Logistic Regression and the survival analysis methods KM and Cox Regression. Those methods have also been proved applicable on knee prosthesis data in collaborative work by Ånneland (2021).

The SUS score was 84.5 which indicates that the usability of the system was well within the acceptance range. For the first first fully functioning prototype this is an encouraging evaluation that motivates further refinement.

Based on the expert evaluation, we consider the novelty of the artifact to be twofold. First, we bridge the gap between humans and statistical models by allowing end-users to assess the quality of hip implants in a direct and more tailored manner. Second, we may easily develop or adapt the system to suit the needs of an implant monitoring system to detect and warn about underperforming prostheses. Although we identified some minor usability problems during evaluation, the feedback was generally positive, and complementary work by Stolt-Nielsen already addresses some of these usability issues (Stolt-Nielsen, 2021).

10.2 Future Work

Future work should combine the efforts of the front-end and the back-end team to make a complete system for a wider audience such as patients, researchers, and physicians. Farsund (2021) and Stolt-Nielsen (2021)'s work would open the system to more general user groups such as patients or physicians with powerful visualizations of demographics and a more

user-friendly UI. Likewise, the complementary work by Ånneland (2021) adapts the system to TKA.

It is likely that future work would benefit from further involvement of domain experts to supervise the data mining aspects of the project. Additional models and methods should be examined for reliability and efficacy and carefully selected through consultation with an expert or statistician. Models that account for time-varying covariates such as age or biomaterial wear should be of prime interest. Such models may require a preprocessing component to derive these features and transform the dataset into a suitable format for time-dependent covariates. For example, we may derive the patients' age from birthyear or record the patients' weight as it varies due to lifestyle changes or medication. The Cox model can account for time-varying covariates with minor alternations. The Lifelines module offers an implementation of the Cox model that handles this.

A preprocessing mechanism provides other benefits besides allowing for time-varying models. For example, we may obtain better predictions from Logistic Regression by transforming data into a more coarse or 'simplified' representation using dimensionality reduction techniques such as PCA. As one evaluator suggested: "Aggregating data into less detail might give better prediction results.". We did not pursue this further because the test dataset was not suitable for such a treatment. However, the early phases of our work resulted in the development of an experimental routine for preprocessing data that could potentially work well for PCA analysis (see Section 6.1). Although that routine is not fully finalized, it might serve as a basis for future work.

Future work should explore additional visualizations and design interfaces to meet the needs of a broad user-group expected to be interested in national registry as a resource.

Appendix A

NSD Approval

NSD NORSK SENTER FOR FORSKNINGSDATA

NSD's assessment

Project title

Maskinlæring i norsk register for hofteproteser

Reference number

700079

Registered

18.10.2020 av Knut T. Hufthammer - Knut.Hufthammer@uib.no

Data controller (institution responsible for the project)

Universitetet i Bergen / Det samfunnsvitenskapelige fakultet / Institutt for informasjons- og medievitenskap

Project leader (academic employee/supervisor or PhD candidate)

Ankica Babic, Ankica.Babic@uib.no, tlf: 4755589139

Type of project

Student project, Master's thesis

Project period

15.11.2020 - 15.06.2021

Status

29.10.2020 - Assessed

Assessment (1)

29.10.2020 - Assessed

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet med vedlegg den 29.10.2020, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

DEL PROSJEKTET MED PROSJEKTANSVARLIG
Det er obligatorisk for studenter å dele meldeskjemaet med prosjektansvarlig (veileder). Det gjøres ved å trykke på "Del prosjekt" i meldeskjemaet.

MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:

https://nsd.no/personvernombud/meld_prosjekt/meld_endringer.html

Du må vente på svar fra NSD før endringen gjennomføres.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 15.06.2021.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

• lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
• formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke behandles til nye, uforenlige formål
• dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
• lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER
Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20).

NSD vurderer at informasjonen om behandlingen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og/eller rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!

Tlf. Personverntjenester: 55 58 21 17 (tast 1)

Appendix B

Scikit-learn pipeline

Listing 1 shows an example of a JSON structure used in the preprocessing routine described in 6.1.1. The output the routine is a Scikit-learn pipeline object that can transform a dateset into an appropriate format for machine learning algorithms. Figure B.1 shows an example of such a pipeline.

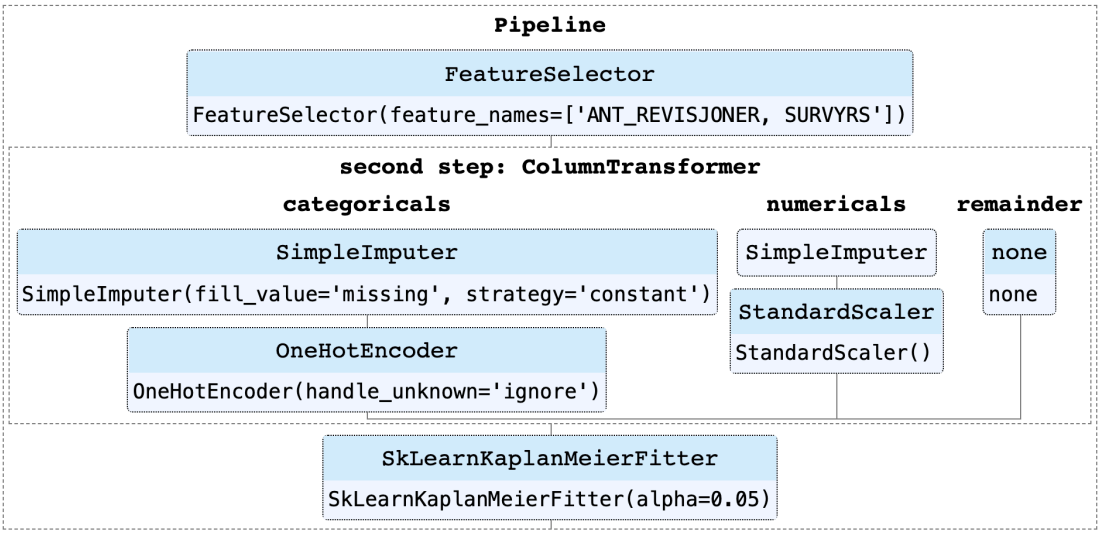


Figure B.1: Scikit-learn pipeline produced by the preprocessing routine from 6.1.1. The input to the processing routine should be a series of preprocessing steps and transformations to apply to the data. Example input is provided in Listing 1. The depicted figure is an HTML representation of the pipeline and is intended for illustration only.

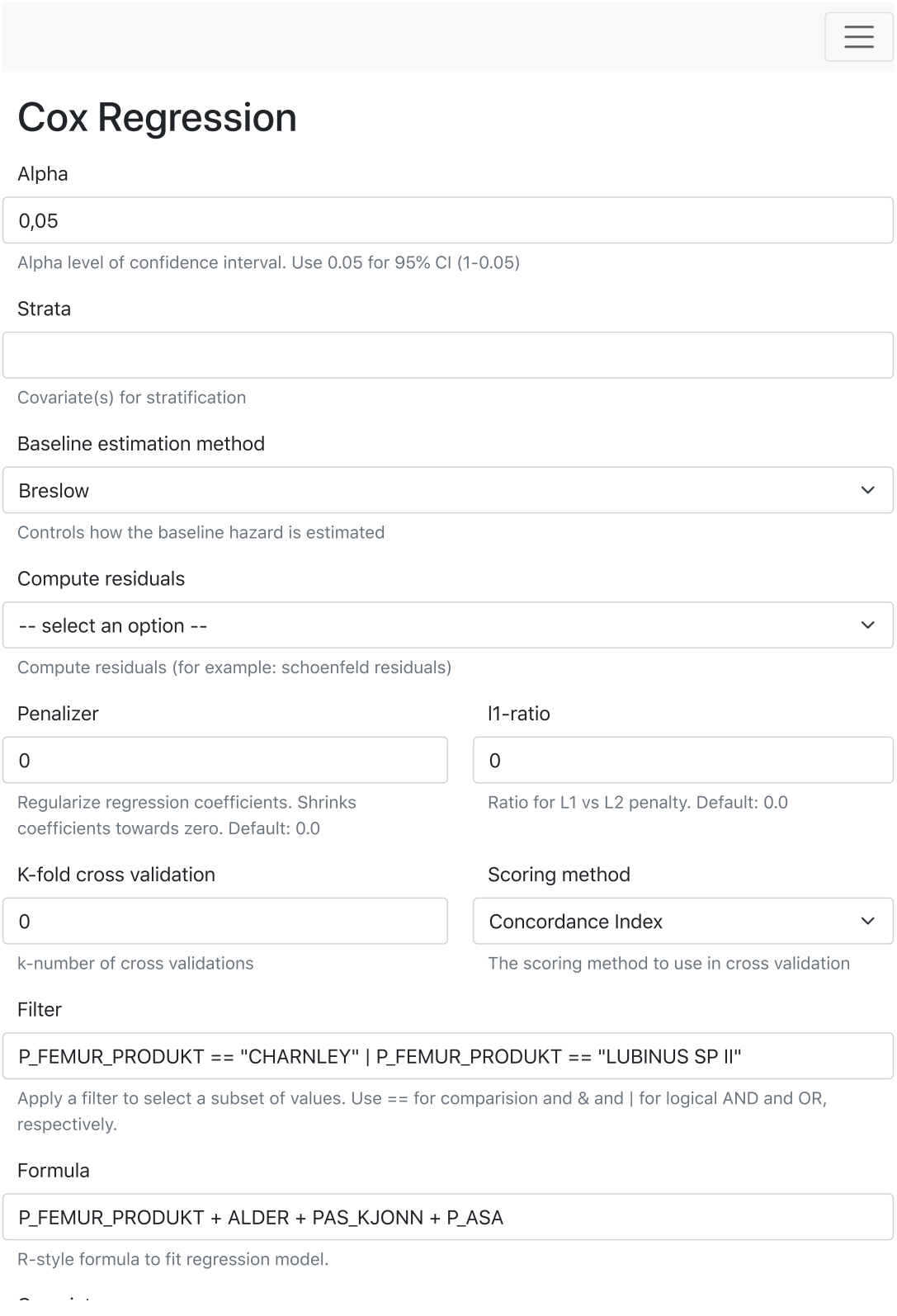
```
{
   "steps":[
      {
         "step_name":"Feature Selection",
         "step":"FeatureSelector",
         "params":{
            "feature_names":[
               "ANT_REVISJONER, SURVYRS"
            ]
         }
      },
      {
         "step_name":"Transform categorical and numerical features",
         "step":"ColumnTransformer",
         "params":{
            "remainder":"none",
            "sparse_threshold":"sparse",
            "n_jobs":"2",
            "transformer_weights":"weight",
            "verbose":""
         },
         "substeps":[
            {
               "step_name":"Pipeline for categorical features",
               "step":"Pipeline",
               "params":{

               },
               "substeps":[
                  {
                     "step_name":"Replace missing values with a constant",
                     "step":"SimpleImputer",
                     "params":{
                        "strategy":"constant",
                        "fill_value":"missing"
                     }
                  },
                  {
                     "step_name":"Encode categoricals as binary features",
                     "step":"OneHotEncoder",
                     "params":{
                        "handle_unknown":"ignore"
                     }
                  }
               ],
               "columns":[
                  "SURVYRS",
                  "ANT_REVISJONER"
               ]
            },
            // ...substep (pipeline) for numerical features.
         ]
      }
   ]
}
```

Listing 1: JSON structure used to construct a Scikit-learn pipeline object.

Appendix C

Cox Regression procedure

Figure C.1 shows the UI created in the third iteration that users can use to fit a Cox model in our prototype (See Section 6.4). Listing 2 shows the Web API request used to fit the model. Listing displays the SQLAlchemy (Bayer, 2012) representation of a fitted Cox model. This database model was used to persist the models into a database for later use.


Cox Regression
Alpha
0,05
Alpha level of confidence interval. Use 0.05 for 95% CI (1-0.05)
Strata
Covariate(s) for stratification
Baseline estimation method
Breslow
Controls how the baseline hazard is estimated
Compute residuals
-- select an option --
Compute residuals (for example: schoenfeld residuals)
Penalizer
0
Regularize regression coefficients. Shrinks coefficients towards zero. Default: 0.0
l1-ratio
0
Ratio for L1 vs L2 penalty. Default: 0.0
K-fold cross validation
0
k-number of cross validations
Scoring method
Concordance Index
The scoring method to use in cross validation
Filter
P_FEMUR_PRODUKT == "CHARNLEY" | P_FEMUR_PRODUKT == "LUBINUS SP II"
Apply a filter to select a subset of values. Use == for comparision and & and | for logical AND and OR, respectively.
Formula
P_FEMUR_PRODUKT + ALDER + PAS_KJONN + P_ASA
R-style formula to fit regression model.

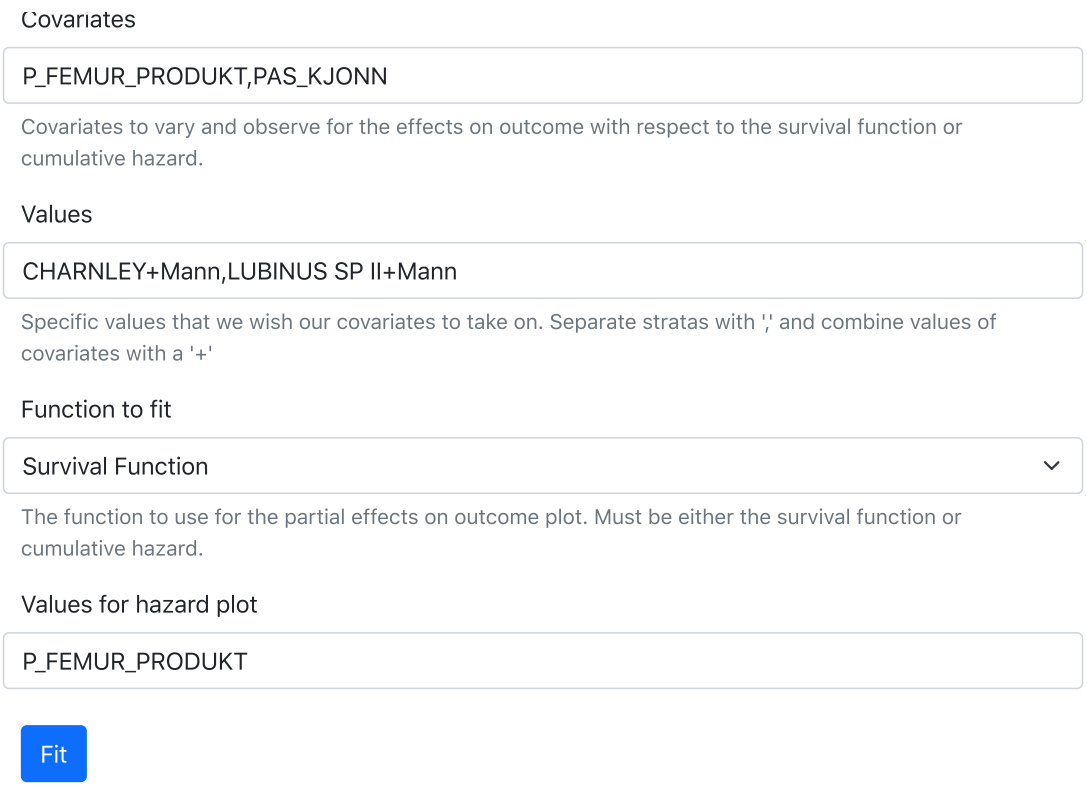


Figure C.1: The UI created for the ’fitting’ part of the Cox regression procedure.

```javascript
/**
  * Fit a Cox model
  * */
 async function getCoxAnalysis(event) {
     event.preventDefault();
     baseUrl = "http://localhost:8000/analyses/CoxPHAnalysis";
     form = event.currentTarget;
     formData = new FormData(form);
     plainFormData = Object.fromEntries(formData.entries());
     url = new URL(baseUrl);
     searchParams = new URLSearchParams();
     Object.keys(plainFormData).forEach(function(key) {
       split = plainFormData[key].split(',');
       for (i = 0; i < split.length; i++) {
         if (searchParams.has(key)) {
           searchParams.append(key, split[i]);
         }
         else {
           searchParams.set(key, split[i])
         }
       }
     })
     url.search = searchParams;

     const response = await fetch(url.toString(), {
         method: 'POST',
         mode: 'cors',
         //... other properties
     });

     if (response.ok) {
       response.json().then(data => {
         //Present results from the analysis
         printModelSummary(data);
         printLogLikelihoodRatioStatistic(data);
         plot_partial_effects_on_outcome(data);
         createTable(data);
         plot_hazards(data);
         printAssumptions(data);
       });
     }
     else {
         //Alert the user about what went wrong
       });
     }
 }
```

Listing 2: API request in JavaScript to fit a Cox model from the front-end.

```python
class CoxPHModel(Base):
    __tablename__ = "coxph_models"

    id = Column(Integer, primary_key=True, index=True)
    model_name = Column(String, index=True)
    model_type = Column(String, index=True)
    description = Column(String, index=True)
    covariates = Column(String, index=True)
    formula = Column(String, index=True)
    model = Column(LargeBinary)
    residuals = Column(LargeBinary)

    dataset = relationship("DatasetModel", back_populates="fitter", uselist=False)
```

Listing 3: Database representation of a fitted Cox model (SQLAlchemy).

Bibliography

Andersen, P. K. and N. Keiding (2005). *Survival Analysis, Overview*, Volume 6. https://onlinelibrary.wiley.com/doi/abs/10.1002/0470011815.b2a11072. 14

Arthursson, A. J., O. Furnes, B. Espehaug, L. I. Havelin, and J. A. Söreide (2005). Validation of data in the norwegian arthroplasty register and the norwegian patient register: 5,134 primary total hip arthroplasties and revisions operated at a single hospital between 1987 and 2003. *Acta Orthop 76*(6), 823–8. 7

ASA (2021). American Society of Anesthesiologists (ASA). https://www.asahq.org/standards-and-guidelines/asa-physical-status-classification-system. 37

Bangor, A., P. Kortum, and J. Miller (2009, May). Determining what individual sus scores mean: Adding an adjective rating scale. *4*(3), 114–123. 30

Bayer, M. (2012). Sqlalchemy. In A. Brown and G. Wilson (Eds.), *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*. aosabook.org. 34, 47, 83

Berge, Ø. S. (2019). *Multiple Imputation in Predictive Modeling of Arthroplasty Database*. Master thesis, The University of Bergen. https://hdl.handle.net/1956/20393. 5, 11, 33, 72

Berntsen, E. (2014). *Information system for postmarket surveillance of total joint prostheses*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/8294. 11

Bootstrap (2021). Bootstrap. https://getbootstrap.com/. 34

Borgan, Ø. (1997). Three contributions to the Encyclopedia of Biostatistics: The Nelson-Aalen, Kaplan-Meier, and Aalen-Johansen. https://www.duo.uio.no/handle/10852/10287. 17

Brajer, N., B. Cozzi, M. Gao, M. Nichols, M. Revoir, S. Balu, J. Futoma, J. Bae, N. Setji, A. Hernandez, and M. Sendak (2020). Prospective and external evaluation of a machine learning model to predict in-hospital mortality of adults at time of admission. *JAMA Network Open 3*(2), e1920733–e1920733. https://doi.org/10.1001/jamanetworkopen.2019.20733. 33

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Andriy Burkov. https://books.google.no/books?id=0jbxwQEACAAJ. 12, 13, 14, 95, 97

Carlsen, T. A. (2018). *Designing an e-learning platform for patients undergoing hip replacement surgery*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/18769. 11

Chen, E. and J. Rossman (2014). A brief history of computer science. https://www.worldsciencefestival.com/infographics/a_history_of_computer_science/. (Accessed on 01/24/2021). 27

Consortium, A. B. (2021). Timeboxing. https://www.agilebusiness.org/page/ProjectFramework_13_Timeboxing. 27

Craddock, A. (2014). The DSDM Agile Project Framework. Ashford, Kent, UK. 26, 27, 95

Dale, H., I. Skråmm, H. L. Løwer, H. M. Eriksen, B. Espehaug, O. Furnes, F. E. Skjeldestad, L. I. Havelin, and L. B. Engesaeter (2011). Infection after primary hip arthroplasty: a comparison of 3 norwegian health registers. *Acta Orthop 82*(6), 646–54. https://pubmed.ncbi.nlm.nih.gov/22066562/. 6

Davidson-Pilon, C., J. Kalderstam, N. Jacobson, S. Reed, B. Kuhn, P. Zivich, M. Williamson, AbdealiJK, D. Datta, A. Fiore-Gartland, A. Parij, D. WIlson, Gabriel, L. Moneda, A. Moncada-Torres, K. Stark, H. Gadgil, Jona, K. Singaravelan, L. Besson, M. S. Peña, S. Anton, A. Klintberg, GrowthJeff, J. Noorbakhsh, M. Begun, R. Kumar, S. Hussey, S. Seabold, and D. Golland (2021, March). Camdavidsonpilon/lifelines: 0.25.10. https://doi.org/10.5281/zenodo.4579431. 16, 18, 19, 33, 38

Ertkjern, Ø. (2015). *Postmarket Surveillance of Orthopaedic Implants using Web-technologies*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/9996. 11

Farsund, A. (2021). *Arthroplasty Data Visualization*. Master's thesis, The University of Bergen. 11, 48, 71, 73, 75

FastAPI (2021). FastAPI. https://fastapi.tiangolo.com/. 33

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth (1996). From data mining to knowledge discovery in databases. *AI Mag. 17*, 37–54. https://ojs.aaai.org//index.php/aimagazine/article/view/1230. 27, 28, 96

Ferguson, R. J., A. J. Silman, C. Combescure, E. Bulow, D. Odin, D. Hannouche, S. Glyn-Jones, O. Rolfson, and A. Lübbeke. asa class is associated with early revision and reoperation after total hip arthroplasty: an analysis of the geneva and swedish hip arthroplasty registries. 37

Foran, J. R. (2015). Total hip replacement. https://orthoinfo.aaos.org/en/treatment/total-hip-replacement/. 6, 97

Fotso, S. et al. (2019). PySurvival: Open source package for survival analysis modeling. https://www.pysurvival.io/. 18

Furnes, O. and L. Havelin (2002). Hip and knee replacement in Norway 1987-2000. 7

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, Inc. 12, 13, 14, 96

Git (2021). Git. https://git-scm.com/. 34

GitHub (2021). Github. https://github.com/. 34

Goel, M., P. Khanna, and J. Kishore (2010). Understanding survival analysis: Kaplan-meier estimate. *International journal of Ayurveda research 1*, 274–8. https://pubmed.ncbi.nlm.nih.gov/21455458/. 16, 96

Graves, S. (2010, 02). The value of arthroplasty registry data. *Acta orthopaedica 81*, 8–9. 1

Hallan, G. (2007). *Wear, fixation, and revision of hip prostheses*. Doctoral thesis, The University of Bergen. https://hdl.handle.net/1956/2166. 4

Hallan, G., B. Espehaug, O. Furnes, H. Wangen, P. J. Høl, P. Ellison, and L. I. Havelin (2012, Feb). Is there still a place for the cemented titanium femoral stem? 10,108 cases from the Norwegian Arthroplasty Register. *Acta Orthop 83*(1), 1–6. 1, 31

Halvorsen, V. B., A. M. Fenstad, L. B. Engesæter, L. Nordsletten, S. Overgaard, A. B. Pedersen, J. Kärrholm, A. Eskelinen, K. T. Mäkelä, and S. M. Röhrl (2019, 5). Outcome of 881 total hip arthroplasties in 747 patients 21 years or younger: data from the Nordic Arthroplasty Register Association (NARA) 1995-2016. *Acta Orthopaedica*. https://doi.org/10.1080/17453674.2019.1615263. 6

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant (2020, September). Array programming with NumPy. *Nature 585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2. 34

Hevner, A., A. R, S. March, S. T, Park, J. Park, Ram, and Sudha (2004). Design science in information systems research. *Management Information Systems Quarterly 28*, 75–. 23, 24, 25

Hipp, R. D. (2020). SQLite. 48

Höftprotesregistret, S. (n.d). https://shpr.registercentrum.se/in-english/contact/p/HkulEOJHz. 8

Iden, A. (2020). *Data Mining Approach to Modelling of Outcomes in Total Knee Arthroplasty*. Master's thesis, The University of Bergen. 11, 13, 71, 72

J. Reschke, E. and E. R. Fielding (2014, June). Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231, RFC Editor. https://www.rfc-editor.org/rfc/rfc7231.txt. 20

Jager, K. J., P. C. van Dijk, C. Zoccali, and F. W. Dekker (2008). The analysis of survival data: the Kaplan–Meier method. *Kidney International 74*(5), 560–565. https://www.sciencedirect.com/science/article/pii/S0085253815533681. 17

Josefsson, S. (2006, October). The Base16, Base32, and Base64 Data Encodings. RFC 4648. https://rfc-editor.org/rfc/rfc4648.txt. 44

Kleinbaum, D. G. and M. Klein (2012). Survival analysis : A self-learning text, third edition. 14, 15, 16, 17, 95, 96

Kongehuset (2018). https://www.kongehuset.no/nyhet.html?tid=165045&sek=26939. 8

Kristoffersen, Y. (2019). *Mining for individual patient outcome prediction in hip arthroplasty registry data*. Master's thesis, The University of Bergen. https://hdl.handle.net/1956/21121. 3, 4, 11, 33

Krumsvik, O. A. (2017). *A Self-Reporting Tool to Reduce the Occurrence of Postoperative Adverse Events After Total Hip Arthroplasty*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/15991. 11

LaTeX.js (2021). LaTeX.js. https://latex.js.org/. 62

Long, J. (1997). *Regression models for categorical and limited dependent variables*. Thousand Oaks: Sage Publications. 18

Longberg, P.-N. (2018). *HALE, the Hip Arthroplasty Longevity Estimation system*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/18783. 4, 11, 33, 71, 72, 96

Lübbeke, A., A. Silman, C. Barea, D. Prieto-Alhambra, and A. Carr (2018). Mapping existing hip and knee replacement registries in europe. *Health Policy 122*(5), 548–557. 1

McConnell, S. (2004). *Code complete* (2 ed.). Microsoft Press. 26

on Arthroplasty, N. N. A. U. and H. Fractures (2019, June 2019). Annual report 2019. Report, Helse Bergen HF, Haukeland University Hospital. 6

on Arthroplasty, N. N. A. U. and H. Fractures (n.d). http://nrlweb.ihelse.net/eng/. 8

Park, A. (2019). How do APIs work? An in-depth guide. https://tray.io/blog/how-do-apis-work. 20, 97

Pedersen, A. B. and A. M. Fenstad (2016, 1). NORDIC ARTHROPLASTY REGISTER ASSOCIATION (NARA) REPORT. Annual report, Nordic Arthroplasty Register Association. ISBN 978-91-639-0167-6. 7

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830. 13, 32, 33

Peduzzi, P., J. Concato, A. R. Feinstein, and T. R. Holford (1995). Importance of events per independent variable in proportional hazards regression analysis II. Accuracy and precision of regression estimates. *Journal of Clinical Epidemiology 48*(12), 1503–1510. https://www.sciencedirect.com/science/article/pii/0895435695000488. 17, 18

Pivec, R., A. Johnson, S. Mears, and M. Mont (2012, 09). Hip arthroplasty. *Lancet 380*. 1

Plotly Technologies Inc. (2015). Collaborative data science. https://plot.ly. 34

Postman (2021). Postman. https://www.postman.com/. 42

R Development Core Team (2004). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org. 5

Raykar, V. C., H. Steck, B. Krishnapuram, C. Dehing-Oberije, and P. Lambin (2007). On ranking in survival analysis: Bounds on the concordance index. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, Red Hook, NY, USA, pp. 1209–1216. Curran Associates Inc. 18

Reback, J., W. McKinney, jbrockmendel, J. V. den Bossche, T. Augspurger, P. Cloud, gfyoung, Sinhrks, A. Klein, M. Roeschke, S. Hawkins, J. Tratner, C. She, W. Ayd, T. Petersen, M. Garcia, J. Schendel, A. Hayden, MomIsBestFriend, V. Jancauskas, P. Battiston, S. Seabold, chris b1, h vetinari, S. Hoyer, W. Overmeire, alimcmaster1, K. Dong, C. Whelan, and M. Mehyar (2020, March). pandas-dev/pandas: Pandas 1.0.3. https://doi.org/10.5281/zenodo.3715232. 34

Register, T. D. H. A. (n.d). http://danskhoftealloplastikregister.dk/en/dhr/. 8

ReliaSoft (2015). *Life Data Analysis Reference*. ReliaSoft Corporation. 96

rpy2 (2021). rpy2. https://rpy2.github.io/. 49

RStudio, Inc (2021). *Web Application Framework for R*. https://cran.r-project.org/web/packages/shiny/shiny.pdf. 5

Rufibach, K. (2015). *Package 'reporttools'*. https://cran.r-project.org/web/packages/reporttools/reporttools.pdf. 49, 62

Sauro, J. (2011a, February). MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS). https://measuringu.com/sus/. 29

Sauro, J. (2011b, Feb). Measuring Usability with the System Usability Scale (SUS). https://measuringu.com/sus/. 97

Schaeffer, J. F., D. J. Scott, J. A. Godin, D. E. Attarian, S. S. Wellman, and R. C. Mather (2015). The Association of ASA Class on Total Knee and Total Hip Arthroplasty Readmission Rates in an Academic Hospital. *The Journal of Arthroplasty 30*(5), 723–727. 37

SpryMedia Ltd. (n.d). Datatables. https://datatables.net/. 34

SPSS Inc. (2021). IBM SPSS Statistics. https://www.ibm.com/products/spss-statistics. 4

Stolt-Nielsen, S. B. (2021). *Design Driven Development of a Web-Enabled System for Data Mining in Arthroplasty Registry*. Master's thesis, The University of Bergen. 11, 42, 48, 67, 71, 73, 75

Tan, S. and S. Tan (2010). The correct interpretation of confidence intervals. *Proceedings of Singapore Healthcare 19*, 276–278. https://journals.sagepub.com/doi/10.1177/201010581001900316. 95

Teetor, P. (2011). *R Cookbook* (1st ed.). O'Reilly Media Inc. 96

The European Commission (2017). *Regulation (EU) 2017/745*. The European Commission. 1

Tidsskriftet (n.d.). https://tidsskriftet.no/profil/leif-ivar-havelin. 8

Tomar, D. (2013). A survey on Data Mining approaches for Healthcare. *International Journal of Bio - Science and Bio - Technology 5*, 241–266. 28

Trello (2021). Trello. https://trello.com/en. 34

van Buuren, S. (2021). *Multivariate Imputation by Chained Equations*. https://cran.r-project.org/web/packages/mice/mice.pdf. 5

van Eck, N. J. and L. Waltman (2020). Vosviewer. https://www.vosviewer.com/. 8

Van Rossum, G. and F. L. Drake Jr (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam. 32, 47

Varnum, C., A. B. Pedersen, P. H. Gundtoft, and S. Overgaard (2019). The what, when and how of orthopaedic registers: an introduction into register-based research. *EFORT Open Reviews 4*(6), 337–343. https://doi.org/10.1302/2058-5241.4.180097. 7

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference*, pp. 56 – 61. 34

Ånneland, S. (2021). *Web-based Data Mining Tool for Total Knee Arthroplasty*. Master's thesis, The University of Bergen. 11, 75, 76

Åserød, H. (2017). *Mobile Design For Adverse Event Reporting And Pharmacovigilance*. Master's thesis, The University of Bergen. http://hdl.handle.net/1956/15990. 11

Glossary

Area under the ROC curve (AUC) The area under the ROC curve (AUC) is a metric for assessing the discriminatory power of a classifier obtained by computing the area under the ROC curve. A perfect classifier has a score of 1 and a classifier whose predictions are completely random will have an AUC score of 0.5 (Burkov, 2019, pp. 67-68) *see* ROC. 4, 18

Concordance index An evaluation metric used to assess the goodness of fit of survival models such as the Cox Proportional Hazard Model. 18, 19, 56, 72

Confidence interval (CI) A range of values where the estimate of interest (e.g. mean or median) is likely to lie within. A 95% CI means that upon multiple samplings from the same population, the true estimate of interest is expected to lie within the upper and lower 95% range in 95% of the cases. A narrower CI indicates a more precise estimate (Tan and Tan, 2010). 16, 18–20, 53, 55, 95

Confusion matrix A table summarizing the proportion of examples correctly and incorrectly classified by the classification model. (Burkov, 2019, p. 65). 3

Cumulative distribution function (CDF) The probability that an event occurs prior to or at time t. The CDF can be obtained by integrating the PDF:

$$\int_0^t f(t)dt \tag{C.1}$$

(Kleinbaum and Klein, 2012, p. 264). 14, 15, 17

DSR Design Science Research. v, 5, 23, 25, 29, 51, 75

Dynamic System Development Methodology (DSDM) An Agile development methodology suitable for both small and large projects. DSDM focuses on business needs and aims to deliver projects on time. The methodology values collaboration and iterative development (Craddock, 2014). 25–27, 41, 51, 75

False negative rate (FNR) The proportion of positive examples incorrectly predicted as negatives (Burkov, 2019, p. 16). 4

False positive rate (FPR) The proportion of negative examples incorrectly predicted as positives (Burkov, 2019, p. 16). 4

Hazard function The hazard function, $h(t)$ gives the instantaneous potential for the event to occur at a specified moment in time t, given that the individual has survived up to that point in time. The hazard function is a rate, not a probability. For that reason, the hazard function is sometimes referred to as the *conditional failure rate* (Kleinbaum and Klein, 2012, p. 9). The value obtained by the hazard function is dependent on the unit of time used, e.g. hours or years. first. 15, 16

HCI Human-Computer Interaction. 2, 48, 67, 71, 72

Hip Arthroplasty Longevity Estimation system (HALE) A prototypical system for estimating hip prosthesis longevity in hip arthroplasty patients. The system was developed by Longberg (2018) and is intended to be used by physicians. 4

Kaplan-Meier estimator (KM estimator) A non-parametric survival model for estimating survival functions. Kaplan-Meier estimates are easy to compute and simple to interpret (Goel et al., 2010). ix, 16, 17, 31–33, 43, 45, 46, 51, 53, 54, 67, 72, 75

Knowledge Discovery in Databases (KDD) A multi-step process for extracting knowledge from raw data. KDD encompasses a framework for how to store, access, apply algorithms efficiently, interpret and visualize data. The process emphasizes the importance of knowledge as the final end-product (Fayyad et al., 1996, p. 42). v, ix, 5, 27–29, 44, 75

Logistic regression A supervised learning algorithm used for binary and multiclass classification. Logistic regression classifiers works by drawing a decision boundary using the sigmoid function from which it partitions examples into different classes (Géron, 2019, 85-107). 3

Multi-layer perceptron (MLP) A supervised learning algorithm based on neural networks with one or more non-linear layers (hidden layers) (Géron, 2019, p. 289). 3, 13, 72

Multiple linear regression (MLR) A supervised learning algorithm modelling a linear relationship between a set of independent variables and a dependent variable. Multiple linear regression is a generalization of simple linear regression where there are only one independent variable (Teetor, 2011, p. 267). 4

NAR Norwegian Arthroplasty Register. v, 1, 3, 7–10, 66, 67, 75

NARA Nordic Arthroplasty Register Association. 7, 8, 10

Principal Component Analysis (PCA) A dimensionality reduction technique used to transform a high-dimensional space into a lower dimensionality. (Géron, 2019, pp. 13-14). 13, 72

Probability density function (PDF) The probability density function is a way of quantifying the relative likelihood that a random variable takes on range of values as opposed to a single value in an infinite sample space (ReliaSoft, 2015, pp.10-11).. 15

Random Forest A supervised learning algorithm combining multiple decision trees together to form a forest of trees. Each individual tree performs its own prediction and the best prediction is selected according to some algorithm. The random forest algorithm can be used for both regression and classification (Burkov, 2019, pp. 9-10). 3

Receiver Operating Characteristic (ROC) A method used to assess the diagnostic ability of a classifier, i.e. it's discriminatory power or ability to distinguish between examples from the positive and negative class. The ROC curve is a graphical plot. (Burkov, 2019, pp. 16-18). ix, 12, 44, 59, 61

RFC Request for Comments. 20

Survival function The survival function $S(t)$ is the probability of survival beyond time t first. 14–16, 47

System Usability Scale (SUS) A 'quick and dirty' method of evaluating the usability of a system. The SUS is a questionnaire that consists of ten predefined statements with a standardized response format. Participants can respond to these statements with five predefined responses ranging from 'Strongly agree' to 'Strongly disagree' Sauro (2011b). ix, 4, 29, 30, 65, 67, 75

TKA Total Knee Arthroplasty. 13, 76

Total Hip Arthroplasty (THA) A surgical intervention involving the artifical replacement of the hip joint - a ball-and-socket joint located between the femur and acetabulum of the pelvis. The purpose of the surgery is to restore normal hip function using artificial components mimicking the function of the hip joint. In THA, both the femoral head (ball) and acetabulum (socket) is replaced Foran (2015). 1, 2, 5–7, 37, 72

UI User Interface. ix, x, 4, 34, 42, 44, 45, 47–49, 51, 54–60, 76, 83, 85

Web Application Programming Interface (Web API) A service exposing a set of endpoints to exchange resources over the Web (Park, 2019). 20, 21, 25, 33, 34, 44, 46, 47, 51, 66, 75, 83