Prediction of length of stay for primary THA/TKA patients using machine learning in OCON Orthopedische Kliniek

Master thesis

Industrial Engineering & Management

Vafi, M.A.

03-07-2024

Title
Prediction of Length of Stay for Primary THA/TKA patients using Machine Learning in OCON Orthopedische Kliniek

Author
Matthijs A. Vafi
s1727958

Publication date
03-07-2024

Master study programme
Industrial Engineering & Management

Specialisation
Production & Logistics Management

Research institute
University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

Company
OCON Orthopedische Kliniek
Geerdinksweg 141
7555 DL Hengelo
The Netherlands

Supervisors University of Twente
First supervisor UT: Dr. Ipek Seyran Topan
Second supervisor UT: Dr. Peter Schuur

Supervisors OCON
Dr. Judith olde Heuvel
Drs. Rob Lindeman

Management summary

This thesis is performed at OCON Orthopedische Kliniek. OCON is a hospital which specialises in treating patients with orthopaedic problems. OCON also performs surgery on patients who need a primary total knee arthroplasty (TKA) or primary total hip arthroplasty (THA).

Problem description and motivation

One of the problems that OCON faces is that it is difficult to predict how long a patient will recover in a bed after they had their surgery. The amount of time that a patient recovers in the bed department is called the length of stay (LOS). The LOS is defined as the time between when the patient arrives at a bed at the bed department up until the patient is discharged to leave the hospital. Having better predictions solves multiple problems. We divide these problems roughly in two types of problems: patient-related problems and planning-related problems. Patient-related problems refer to the inconveniences perceived by the patient due to inaccurate LOS predictions, and planning-related problems refer to how OCON's planning is negatively impacted due to wrong predictions.

Central research problem

The research of this thesis focusses on predicting the LOS for patients who underwent a primary TKA/THA. OCON wants better predictions for the postoperative LOS for these types of patients, and they want to know which patients will experience a long LOS. A long LOS is defined as a LOS of at least 3 days. This brings us to the central research problem:

"A method should be devised which is able to make better predictions for the postoperative length of stay for patients who underwent a primary TKA/THA."

To solve this research problem, we created a tool which helps predict whether a THA or TKA patient will experience a long or short LOS.

Approach

We performed an extensive literature review on predicting LOS for THA and TKA patients. We selected relevant features for predicting the LOS for such patients. We asked OCON for data while using the relevant features according to literature. Not all data for these features was received, primarily because of issues with the HiX database that OCON uses. The size of the total dataset was 260 columns/variables and 5391 rows/surgeries. Most of the 260 variables turned out to be post- or perioperative variables, making them irrelevant because the purpose of this research is to make a *preoperative* prediction of the LOS by using preoperative variables only. Other problems occurred: many features were stored as written text, or otherwise unusable because they contained no LOS-related data (26% of the data) such as discharge date and time. There were missing or faulty values for certain features. Even though we performed data imputation to fill in missing values, the imputed values are never as good as the real values. After extensive and time-consuming data analysis, we ended up with a dataset containing 2208 THA surgeries and 1766 TKA surgeries, with 22 remaining features.

We analysed relevant models used for predicting the LOS for THA/TKA patients. From the literature, we selected eight machine learning (ML) models. Those are:

1. Logistic regression (LOGR)
2. Naive Bayes classifier (NBC)
3. K-nearest neighbour (KNN)
4. Linear support vector machine (LSVM)
5. Support vector machine with radial basis function kernel (SVMR)
6. Random forest (RF)
7. Extreme gradient boosting (XGB)
8. Artificial neural network (ANN)

To further decrease the number of features needed, we performed three feature selection methods (FSM). Reducing features shows which features are important and which are redundant, which increases the interpretability of the resulting ML models. The FSMs are:

1. Filter FSM based on chi-squared tests, one-way ANOVA F-tests, and Pearson correlation coefficients. This FSM does not depend on the ML model used and the selected feature subset is the same for all ML models of a certain surgery type (TKA or THA).
2. Wrapper FSM, namely backward sequential feature selection.
3. Embedded FSM based on feature importance. Additionally, we performed Lasso on the LOGR model, where we selected the best performing feature subset and a feature subset with good performance but requiring fewer features.

We also used a dataset containing all remaining features, which we did not perform any FSM on. We did this to measure how well the feature subsets resulting from the FSM perform compared to when we use all remaining features. In the end, we ended up with 34 combinations of ML models and feature subsets per surgery type.

Results

To measure the performance of each resulting ML model, we used the area under the curve (AUC) of the receiver operating characteristic (ROC) curve of each model, where a higher AUC is better. The ROC curve is a plot which shows the classification performance of the used ML models for various cutoff values. We also took the number of features for each model into account when selecting the most promising models, where using fewer features is more favourable. Table 1 and Table 2 show the performance of each created model.

Table 1: AUC values and number of features overview of the models trained and tested on the THA data.

AUC	**All**	**Filter**	**Wrapper**	**Embedded**	**# of features**	**All**	**Filter**	**Wrapper**	**Embedded**
LOGR	***0.7517***	***0.7454***	0.7407	***0.7431***	**LOGR**	***22***	***18***	13	***13***
LSVM	0.7283	0.7250	0.7284	0.6659	**LSVM**	22	18	22	20
SVMR	0.6941	0.7001	0.6705	0.6791	**SVMR**	22	18	22	20
KNN	0.6905	0.6229	0.6468	0.6626	**KNN**	22	18	5	20
NBC	0.7220	0.7171	0.7251	0.7141	**NBC**	22	18	20	20
RF	***0.7551***	***0.7422***	***0.7531***	0.7112	**RF**	***22***	***18***	***22***	11
XGB	***0.7538***	0.7405	***0.7454***	***0.7456***	**XGB**	***22***	18	***19***	***12***
ANN	0.7398	0.7390	0.6653	0.7190	**ANN**	22	18	15	8
LassoOpt				***0.7471***	**LassoOpt**				***17***
LassoStd				0.7126	**LassoStd**				10

Table 2: AUC values and number of features overview of the models trained and tested on the TKA data.

AUC	All	Filter	Wrapper	Embedded	# of features	All	Filter	Wrapper	Embedded
LOGR	***0.7211***	***0.7251***	***0.7206***	***0.7440***	LOGR	***22***	***8***	***18***	***13***
LSVM	0.6388	0.5929	0.5271	0.6742	LSVM	22	8	8	20
SVMR	0.6936	0.6435	0.6946	0.6708	SVMR	22	8	22	20
KNN	0.6331	0.6194	0.6806	0.6520	KNN	22	8	8	20
NBC	0.6943	***0.7194***	0.7189	0.6859	NBC	22	***8***	5	20
RF	***0.7190***	***0.7327***	0.6901	0.7101	RF	***22***	***8***	21	12
XGB	0.7057	***0.7254***	0.7158	0.7061	XGB	22	***8***	14	12
ANN	0.7033	0.7092	0.6211	***0.7234***	ANN	22	8	8	***15***
LassoOpt				***0.7192***	LassoOpt				***23***
LassoStd				0.6938	LassoStd				1

In Table 1 and Table 2, the six promising models are coloured green and selected to be used in the prediction tool. We selected models with a high AUC, and we also selected models that have a good AUC and relatively few features.

ML models were for classification of output fractions. These fractions range from 0 to 1. The higher the fraction, the more likely the model expects a long LOS. The actual prediction on whether a patient is expected to experience a long LOS depends on a cutoff value, and thus the models require a cutoff value. If the output fraction is higher than this cutoff, then the model predicts the corresponding patient to experience a long LOS. Table 3 shows a summary of the selected ML models. The sensitivity is a measure of how well a model can identify patients who will experience a long LOS, while the specificity is about how well the model identifies short LOS patients. A higher sensitivity or specificity means a better performance. We chose a default cutoff with the highest sum of the sensitivity with the specificity. Picking a lower cutoff value for an ML model will increase the sensitivity at the cost of decreasing the specificity.

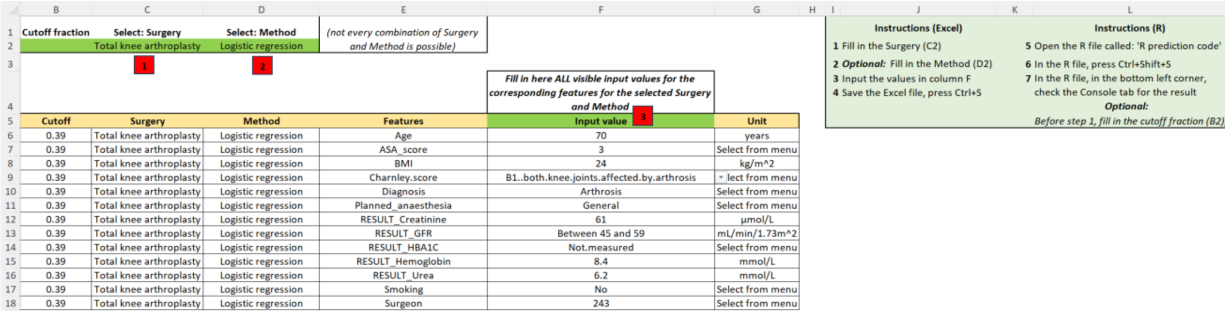
Table 3: Overview of performance metrics of each of the selected ML models for the prediction tool.

Surgery	Method	Number of features	AUC	Default cutoff	Sensitivity	Specificity	Accuracy
Total hip arthroplasty	Random forest	22	0.76	0.191	0.86	0.55	0.62
Total hip arthroplasty	Logistic regression	13	0.74	0.215	0.7	0.67	0.68
Total hip arthroplasty	Artificial neural network	8	0.72	0.176	0.79	0.57	0.62
Total hip arthroplasty	XGBoost	12	0.75	0.197	0.72	0.66	0.67
Total knee arthroplasty	Random forest	8	0.73	0.253	0.82	0.55	0.61
Total knee arthroplasty	Logistic regression	13	0.74	0.39	0.51	0.84	0.75

We utilised the knowledge gathered by designing a *prediction tool*. The tool is validated by OCON staff, and we made adjustments based on the gathered feedback. This prediction tool has an Excel interface, as shown in Figure 1, where the user inputs - for a specific patient - the surgery type, machine learning method, the values for each corresponding feature, and optionally the cutoff fraction. Output will be whether the patient is expected to experience a long LOS. Figure 2 shows the R script which uses the data in Excel to run the ML models. The steps for using the tool correspond to the red boxes and are as follows:

1. Fill in the Surgery (C2), which is either THA or TKA.
2. Fill in a machine learning method at the Method (D2).
3. Input the values for the corresponding features in column F.
4. The user must save the Excel file, such as by pressing Ctrl+S.
5. Open the R file called: 'R prediction code'.
6. In the R file, press Ctrl+Shift+S to run the script, or press the Source button.

7. In the R file, in the bottom left corner, check the Console tab for the resulting prediction. This is either long LOS of at least 3 days, or a short LOS of less than 3 days.



Cutoff fraction	Select: Surgery	Select: Method	*(not every combination of Surgery and Method is possible)*
	Total knee arthroplasty	Logistic regression	

Fill in here ALL visible input values for the corresponding features for the selected Surgery and Method

Cutoff	Surgery	Method	Features	Input value	Unit
0.39	Total knee arthroplasty	Logistic regression	Age	70	years
0.39	Total knee arthroplasty	Logistic regression	ASA_score	3	Select from menu
0.39	Total knee arthroplasty	Logistic regression	BMI	24	kg/m^2
0.39	Total knee arthroplasty	Logistic regression	Charnley.score	B1..both.knee.joints.affected.by.arthrosis	lect from menu
0.39	Total knee arthroplasty	Logistic regression	Diagnosis	Arthrosis	Select from menu
0.39	Total knee arthroplasty	Logistic regression	Planned_anaesthesia	General	Select from menu
0.39	Total knee arthroplasty	Logistic regression	RESULT_Creatinine	61	µmol/L
0.39	Total knee arthroplasty	Logistic regression	RESULT_GFR	Between 45 and 59	mL/min/1.73m^2
0.39	Total knee arthroplasty	Logistic regression	RESULT_HBA1C	Not.measured	Select from menu
0.39	Total knee arthroplasty	Logistic regression	RESULT_Hemoglobin	8.4	mmol/L
0.39	Total knee arthroplasty	Logistic regression	RESULT_Urea	6.2	mmol/L
0.39	Total knee arthroplasty	Logistic regression	Smoking	No	Select from menu
0.39	Total knee arthroplasty	Logistic regression	Surgeon	243	Select from menu

Instructions (Excel)

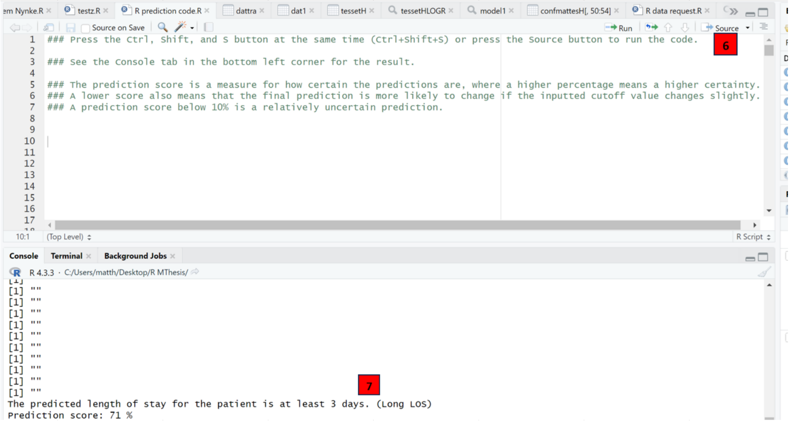
1 Fill in the Surgery (C2)
2 ***Optional:*** Fill in the Method (D2)
3 Input the values in column F
4 Save the Excel file, press Ctrl+S

Instructions (R)

5 Open the R file called: 'R prediction code'
6 In the R file, press Ctrl+Shift+S
7 In the R file, in the bottom left corner, check the Console tab for the result

Optional:
Before step 1, fill in the cutoff fraction (B2)

Figure 1: Updated interface of prediction tool in Excel based on feedback.



```
### Press the Ctrl, Shift, and S button at the same time (Ctrl+Shift+S) or press the Source button to run the code.

### See the Console tab in the bottom left corner for the result.

### The prediction score is a measure for how certain the predictions are, where a higher percentage means a higher certainty.
### A lower score also means that the final prediction is more likely to change if the inputted cutoff value changes slightly.
### A prediction score below 10% is a relatively uncertain prediction.
```

```
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
The predicted length of stay for the patient is at least 3 days. (Long LOS)
Prediction score: 71 %
```

Figure 2: Updated R code of prediction tool based on feedback.

Discussion and recommendations

The main limitation in our research is the received data amount. The received dataset was relatively small compared to datasets used in literature. Most of the literature work with datasets of at least 100,000 surgeries. Despite the small amount of data, our ML models have similar performance to models used in literature. The AUC values of our models range between 0.72 and 0.76, while AUC values in literature range between 0.69 and 0.83. The literature also often uses comorbidity scores as features, which were not present in our received data. It would be beneficial to train new ML models in the future when there is more data, as the models have the potential to improve further. We recommend OCON the following:

- Use the prediction tool and register the predictions done by OCON staff, predictions done by the tool, and whether the concerning patient experienced a long LOS or not. This helps keeping track of the performance of the tool and whether the cutoff value should be adjusted to OCON’s preferences.
- Collect more features which are relevant according to literature, such as comorbidity scores.
- Collect more surgeries to be used as input data for training the ML models. More data should improve the AUC of the models and thus their performance.

Various academic sources use ANN models. The ANN model in our research was limited to one hidden layer. Making ANN models with more hidden layers could provide models with higher AUC. Finally, incorporating LOS predictions into capacity plannings could help optimise scheduling.

Preface

This report is the result of my research done for the Master study Industrial Engineering and Management with the Production and Logistics Management specialisation, which was executed in cooperation with OCON Orthopedische Kliniek. With this report, I finish my studying period at the University of Twente.

I would like to thank my supervisors who helped me throughout the execution of my research. My first UT supervisor, Ipek Seyran Topan, has been a stellar supervisor during both my bachelor and master graduation. She was always very helpful and available for questions. I have had countless discussions with her about all kinds of issues, and her support has always been paramount. During my graduation periods, she witnessed all my ups and downs that I experienced, and I could always count on her for support and understanding. I have great admiration for her empathy and dedication. I was lucky to know her throughout my challenging graduation times and I learned a lot from her. My second UT supervisor, Peter Schuur, has been very supportive and kind throughout my research process. He made me feel comfortable and always had a bunch of great stories and life lessons to tell during our supervisor meetings. I admire his patience and positivity. I appreciate the support, empathy, and guidance that I received from him. Ipek and Peter made my meetings very enjoyable and stressless, and still constructive and very helpful.

My first company supervisor, Judith olde Heuvel, helped me find my way at OCON, and provided me with feedback and guidance during my research. I would like to thank her patience and for giving me space to follow my pursuits. My second company supervisor, Rob Lindeman, was my first OCON contact person and accepted me as a student to perform research at OCON. I am grateful that Rob gave me this opportunity. I would like to thank Rob for his patience and knowledge. My meetings with him were always constructive and pleasant. Rob helped me find the right people at OCON to talk to and provided me expert opinion on medical subjects. Furthermore, I would like to thank some other people who also helped me throughout my graduation period. Emily Bakker helped me with gathering the input data, Erwin Veldhuis showed me around at the preoperative screening, and Nynke Koopman showed me around at the bed department. I met with Emily, Nynke and Erwin multiple times and they helped me understand patient processes and data-related issues. I have also had meetings with Dagmar Wikkerink, Madeleine Quik-Haarhuis and Hermien Rodijk-Meijerink which helped me gain more insight in problems related to a patient’s length of stay. I am grateful for the time that everyone made available for me, and I thank them for their input and knowledge.

Finally, I would like to thank my friends and family for their support throughout my research.

Matthijs Vafi

Enschede, July 2024

Table of Contents

List of Figures

List of Tables

Glossary

Term	**Definition**
ANN	Artificial neural network
AUC	Area under curve
Caret	An R library which we use to program ML methods
Features	Variables which are used for predicting the target variable
FSM	Feature selection method
KNN	K-nearest neighbour
Lasso	A method which reduces required features while maximising a model's predictive performance
LOGR	Logistic regression
Long LOS	A length of stay of at least 3 days
LOS	Length of stay
LSVM	Linear support vector machine
ML	Machine learning
NBC	Naive Bayes classifier
RF	Random forest
ROC curve	Receiver operating characteristic curve
SVMR	Support vector machine with radial basis function kernel
Target variable	The outcome variable which needs to be predicted
THA	Primary total hip arthroplasty
TKA	Primary total knee arthroplasty
XGB/XGBoost	Extreme gradient boosting

Chapter 1: Introduction

In this chapter, we elaborate on the research setup and the context of the research of this thesis. We describe the background and discuss the problems which are the motivation behind the research. Finally, we propose research questions which are the foundation of the research.

1.1 Description of the background

The company which concerns this master thesis assignment is OCON. OCON is short for Orthopedisch Centrum Oost-Nederland. OCON Orthopedische Kliniek is a categorical hospital which specialises in treating patients with orthopaedic problems (*Organisatie*, 2024). The hospital offers a total package of high-quality sports and orthopaedic medical care. OCON is situated within Ziekenhuis Groep Twente (ZGT), which is a hospital group. OCON has two settlements: one in Hengelo, and one in Almelo. We execute our research at the Hengelo settlement of OCON.

OCON's team of medical specialists consists of fifteen orthopaedic surgeons, six anaesthesiologists, and approximately 180 employees. The patient receives tailor made medical care, which is specifically meant to fit the individual, and is in line with state-of-the-art scientific insights. The orthopaedic surgeons and anaesthesiologists form one coherent team of medical specialists, which collaborate intensely in a goal-oriented manner with OCON's medical staff to take care of patients and their needs. This collaboration ensures that the experience and knowledge of different fields are aligned, and this makes sure that the treatment process of patients undergo a constant monitoring and tuning process.

Every year, OCON treats approximately 22,000 different patients and performs about 5,000 surgeries. Surgeries take place in OCON Hengelo. The orthopaedic specialisations which are present in OCON are:

- Knee and hip replacement surgery
- Spine surgery
- Sports orthopaedics and sports medicine
- Foot and ankle surgery
- Wrist and hand surgery
- Shoulder and elbow surgery
- Prosthesis revision surgery
- Paediatric orthopaedics
- Traumatology

The focus of this master assignment is about patients who undergo a primary total knee arthroplasty (TKA) or primary total hip arthroplasty (THA). This surgical procedure replaces the whole knee or hip of a patient with a prosthesis. ‘Primary’ means that the TKA/THA has been performed for the first time for a patient, instead of a revision for example. In 2020, OCON performed approximately 700 THAs and 630 TKAs (Elawady, 2021).

We explain the patient flow of TKA/THA patients briefly as follows. The patient first visits primary care and is then forwarded to a consult with an orthopaedic surgeon. This surgeon analyses the patient at the outpatient clinic. The surgeon judges whether the patient requires a TKA/THA, and if necessary, OCON forwards the patient to pre-operative screening (POS). At the POS, a nurse or physician assistant assesses the patient's ability to undergo a surgery. The goal of this screening is to ensure the medical staff is well prepared for potential complications that might occur during or after a surgery. If the patient is considered too frail for surgery, then OCON could work on prehabilitation before the patient is allowed to undergo a surgery. For example, if the patient is severely overweight, smokes, often drinks alcohol, and is of an age above 70, then it is possible that the surgery could cause dangerous complications. Thus, by making the patient stop drinking, quit smoking, or lose weight, OCON minimises the risks.

After the patient has been approved at the POS, the anaesthesiologist also assesses the surgical applicability of the patient. After this approval, the patient makes an appointment for the surgery. A week before the surgery, OCON invites the patient for a rapid recovery conversation, in which they discuss relevant information regarding the surgery and recovery.

After the surgery, OCON moves the patients to a bed in the nursing department and the patients stay there for a certain number of days until they are ready to go home and able to either take care of themselves, or someone else is able to sufficiently take care of them. It is also possible that the patient transfers to a nursing home instead. The time period which a patient is spending in a bed in the nursing department while they are recovering until they are discharged is called the length of stay (LOS). We measure the LOS in days, plus a fraction of a day. For example, if a patient arrives at the bed department at 11:00 on Monday, and is discharged Wednesday 17:00, then the LOS is 2 days plus 6 hours, which is 2.25 days. The research of this thesis focusses on predicting the LOS for patients who underwent a primary TKA/THA. OCON wants better predictions for the postoperative LOS for these types of patients. Having better predictions helps solving multiple problems. We explain these problems in Section 1.2. In this thesis, whenever the prediction of the LOS is mentioned, it refers to the LOS for patients who underwent a primary TKA or THA.

1.2 Problem context

The first step in this research is to understand the problem context. Thus, we analyse the current processes and problems concerning TKA/THA patients. We use theses done by other students who have done research in different topics at OCON in the past as a source for understanding patient processes (Abbink, 2021; Elawady, 2021; Rolink, 2023). The first few chapters of these theses introduce the background of OCON. Furthermore, one of the student theses explains the patient flow that TKA/THA patients go through when OCON treats them, which provides useful information for the background of this research.

Additionally, we interview OCON's staff to gain more knowledge concerning the processes concerning TKA/THA patients. We walk along with a nurse at the nursing department, and with a physician assistant at the preoperative screening (POS). Furthermore, we have meetings with other OCON staff to discuss and learn more about the context of the assignment. In addition to the OCON supervisors, the physician assistant, and the nurse, we have meetings with the quality and capacity manager, the clinical department process coach, and with the data specialist. The knowledge gathered during all the meetings and interviews is summarised in the problem cluster shown in Figure 1.1.

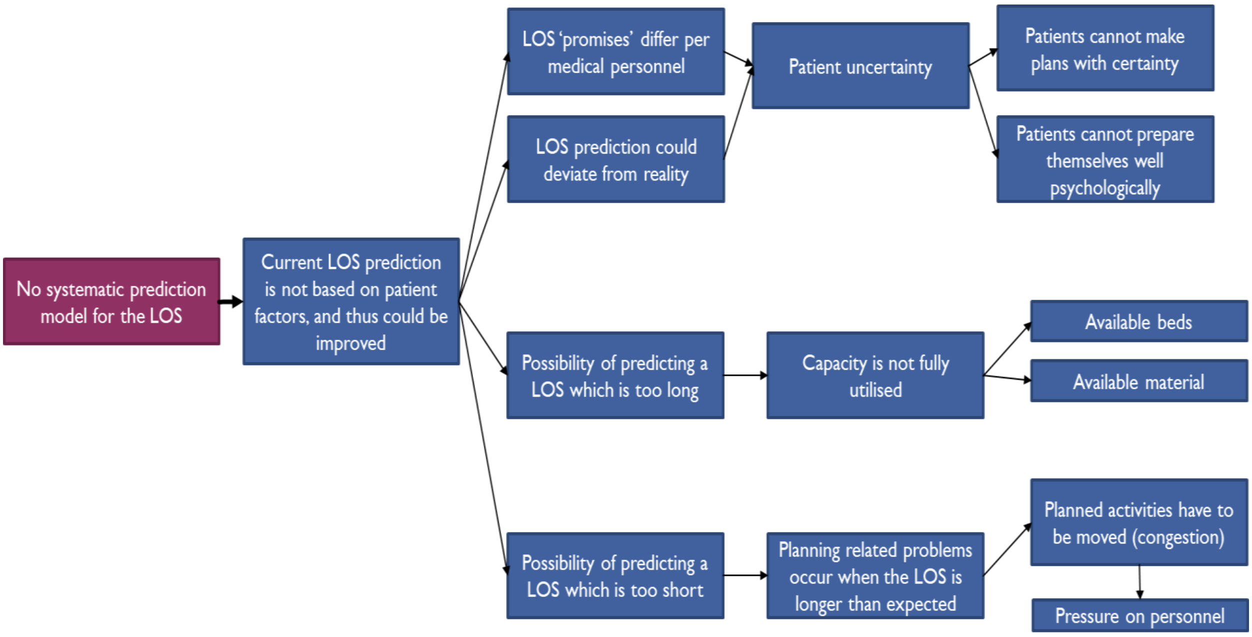

No systematic prediction model for the LOS
Current LOS prediction is not based on patient factors, and thus could be improved
LOS 'promises' differ per medical personnel
LOS prediction could deviate from reality
Patient uncertainty
Patients cannot make plans with certainty
Patients cannot prepare themselves well psychologically
Possibility of predicting a LOS which is too long
Capacity is not fully utilised
Available beds
Available material
Possibility of predicting a LOS which is too short
Planning related problems occur when the LOS is longer than expected
Planned activities have to be moved (congestion)
Pressure on personnel

Figure 1.1: Problem cluster of the current situation in OCON.

The problems in the problem cluster can be divided roughly in two types of problems: patient-related problems and planning-related problems. Patient-related problems refer to the inconveniences perceived by the patient due to inaccurate LOS predictions, and planning-related problems refer to how OCON's planning is negatively impacted due to wrong predictions.

The main problem that patients experience with an inaccurate LOS prediction, is patient uncertainty. Patients do not know for how long they will have to recover at OCON's nursing department. Consequently, these patients cannot psychologically prepare well for their recovery process. This is especially distressing for older and frail patients, who often also have a significantly longer than average LOS. Another inconvenience that patients experience is that they cannot make clear plans. When undergoing a TKA/THA, patients may have to take time off from work, they might have to arrange friends or family to take care of them at home, or they have to postpone appointments or vacation plans. The patient uncertainty comes not only from the fact that the true LOS could be different than expected, but it can also result from medical personnel giving different indications of the patient's LOS. These indications can be interpreted by the patient as promises. For example, if the orthopaedic surgeon tells the patient that it would not be a problem if the patient recovers in the nursing department for a few extra days, then a patient remembers this and base their plans on it, and this could result in patients staying longer at the nursing department than required, which can result in costs and inconvenience. It is beneficial if the medical personnel are on the same page when it comes to making such LOS 'promises' to patients.

OCON's planning-related problems can be caused by two scenarios: either a patient's LOS is longer or shorter than predicted. If the LOS prediction is longer than the patient's actual LOS, then this means that more resources, such as material, space, and personnel, are allocated than required. If the LOS prediction is consistently too long, then this means that patients will consistently leave the nursing department earlier than expected, which creates gaps in OCON's planning. This means that more resources are allocated than required, and these resources are not used in the end. Thus, this means that the capacity is underutilised. It is often not possible to suddenly fill in these gaps in OCON's planning. Thus, beds and other material is available instead of being used, which is an inefficient use of resources.

If the LOS prediction is consistently shorter than predicted, then this means that patients remain longer in the nursing department than expected, which causes overlap in the planning. This is problematic, because it could be the case that certain resources are suddenly required by two different patients. If this is not possible, then this results in a congested planning, which means that certain patient-related activities have to be moved or postponed. In such a situation, the medical staff of OCON is unexpectedly required to do more work, which puts pressure on them. From the conducted meetings, medical staff already have to do a lot of work, especially nurses. This observation further stresses that pressuring the medical staff even more is highly undesirable.

OCON's planning-related problems, as well as the patient-related problems, all originate from the problem that the current LOS is not based on considering patient-related factors. Patient-related factors refer to any aspect which concerns the patient's health, or the processes that the patient experiences in OCON. Examples of such factors are BMI, gender, age, surgery time, and blood loss during surgery. OCON collects these factors and writes them down in HiX. These factors might be an indication of what the LOS of a patient could be. HiX is the database that OCON uses to register information such as patient-related information, and thus it essentially functions as an electronic patient archive. Currently at OCON, there is no systematic model which considers patient-related factors for predicting a patient's LOS. According to OCON's medical staff, the current expected LOS for primary TKA patients is 3 days, and for primary THA patients it is 2 days. These estimations include the day on which the patient undergoes the surgery. The absence of a systematic prediction model for the LOS causes the aforementioned problems that can be seen in Figure 1.1, which means that the core problem, as well as the research problem is as follows:

"A method should be devised which is able to make better predictions for the postoperative length of stay for patients who underwent a primary TKA/THA."

1.3 Problem approach

After identifying the core problem, our problem approach is to devise research questions which aim to solve the core problem. The research questions are the building blocks for the solution for the core problem and answering them should solve the problem. The research questions are categorised in five categories, and each category is a chapter in this thesis. Certain research questions are divided in sub-research questions which help answering them. The (sub-)research questions are as follows:

Introduction (Chapter 1):

1. What core problem is OCON currently facing concerning primary TKA/THA patients?
    a. What is the patient flow of primary TKA/THA patients?
    b. What current problems does OCON face regarding primary TKA/THA patients?

Literature review (Chapter 2):

2. What suitable methods for LOS prediction for primary TKA/THA patients does academic literature mention?
    a. Which methods help reduce the number of required input features for such prediction methods?
    b. Which indicators can be used to measure the performance of a prediction model?

Data collection and analysis (Chapter 3):

3. What input is required for LOS prediction methods?
    a. How do we collect such input data?
    b. What steps are required to make input data suitable for the prediction methods?

Machine learning methods (Chapter 4):

4. How do we configure the LOS prediction methods to improve their performance?
    a. Which methods can reduce the number of required input features?

Results (Chapter 5):

5. Which LOS prediction methods perform the best?
6. How can we implement the LOS prediction methods in practice?

1.4 Thesis outline

In Chapter 1, we introduce OCON and the assignment, and we elaborate on the problem context in Section 1.2. After identifying the core problem and research questions, the next step in this research is to perform a literature study in Chapter 2 to search for methods which can perform predictions for a patient's LOS. We discuss and prepare the input data in Chapter 3, and in Chapter 4 we cover methods that we implement. We test these methods and analyse the corresponding results in Chapter 5. Finally, we discuss limitations, conclusions, and recommendations in Chapter 6.

1.5 Summary

In this chapter, we analyse the problem context regarding the LOS of primary TKA/THA patients. We research the patient flow of these patients to find the current problems that OCON faces regarding these patients. Consequently, we find that the core problem is that OCON requires a method to predict the LOS of such patients. We perform a literature review in Chapter 2 to find suitable methods in academic literature.

Chapter 2: Literature review

In this chapter, we answer the literature review research questions as proposed in Section 1.3. We analyse literature aims to predict with methods, such as machine learning (ML) methods, the postoperative LOS for primary TKA or THA patients. We use the database of Scopus for collecting literature. We answer the following research question: *What suitable methods for LOS prediction for primary TKA/THA patients does academic literature mention?* In Section 2.1, we discuss the collected literature and the methods that are frequently used for LOS prediction. We briefly summarise our findings in Section 2.2.

2.1 Literature overview

Table 2.1 summarises the literature, concerning the goal of the source, patient type, features, and model performance. Features are variables which describe data points. For a patient, features could be the age and BMI. Both TKA and THA can be categorised under total joint arthroplasty, and thus this term is also added to the search query. Replacement is alternative way of mentioning arthroplasty in this context. Discharge time is another possible term which can be used for describing the LOS, which is also included in the search query. Finally, learning refers to machine learning (ML) and *predict** refers to possible conjugations with the word *predict*. We add the concept of ML to the search query because it has shown to be promising in literature when it comes to predicting the LOS (Papalia et al., 2021). The search query used is:

*primary AND total AND (knee OR hip OR joint) AND (replacement OR arthroplasty) AND (length AND of AND stay) OR (discharge AND time) AND learning AND predict**

Most research done on the topic of predicting the LOS for TKA and THA patients has been performed on the American population, and these researchers often extracted their data from databases with American patients such as SPARCS, NIS, and ACS-NSQIP. The amount of research done on European patients is underrepresented, which makes research of these type of patients more innovative.

A recurring trait in the literature in Table 2.1 is that plenty of sources model the LOS as a binary or categorical variable, where time intervals are put together to form a category. If the LOS is modelled as binary, then a distinction is made between a relatively short or prolonged LOS. Thus, we expect the prediction model in this research to be more suitable if the LOS is also modelled as a binary or categorical variable. We refer to a binary variable which indicates a prolonged LOS as *LongLOS*.

Certain variables occur frequently in literature, such as age, BMI, gender, ethnicity, blood values, anaesthesia type, and comorbidity scores. In Chapter 3, we make a selection of variables used in literature for LOS prediction. We use this selection as a starting point for finding relevant input data.

Finally, academia uses a train/test split of 80:20 the most, which means that 80% of the input data is used for training a prediction model, while 20% is used to test the model's performance. However, various academic sources also used other train/test splits, such as 60:40.

Table 2.1: Overview of literature with prediction models for the LOS for primary TKA and THA patients.

Source	**Goal**	**Patient type**	**# patients**	**Methods used**	**Relevant features**	**Pre-/peri-/postoperative features**	**Best performance**	**Additional remarks**
(Zalikha et al., 2023)	Compare ML models' performance for predicting LOS, discharge disposition, and mortality.	Primary TKA patients from US between 2016 and 2017. Extracted from NIS.	~306,000	10 ML models were used, of which LSVM, CHAID, and Decision List were the 3 most promising models.	15 variables: 8 patient-specific (including Age, Gender, Race, Total number of diagnoses, All Patient Refined Diagnosis Related Groups (APRDRG) Severity of illness, APRDRG Mortality risk, Income zip quartile, and Primary payer) and 7 situational variables (including Patient Location, Month of the procedure, Hospital Division, Hospital Region, Hospital Teaching status, Hospital Bed size, and Hospital Control).	Preoperative	KPIs: AUC and accuracy. AUC: 0.689 (LSVM). Accuracy: 85.39% (Decision list). For predicting the LOS, the best performing ML models were: LSVM, Decision Lists, CHAID.	LOS is binary with a cutoff of 2 or less days, which is based on the average of the patients in the study; Patient income was measured in four categories. LOS is defined as when the patient either moves to another facility or home.
(Nham et al., 2023)	Compare patient-specific and situation perioperative variables with ML models to predict postoperative outcomes.	Primary THA patients from US between 2016 and 2017. Extracted from NIS.	~177,000	10 ML models were used, of which LSVM, CHAID, and Decision List were the 3 most promising models.	15 variables: 8 patient-specific (including Age, Gender, Race, Total number of diagnoses, All Patient Refined Diagnosis Related Groups (APRDRG) Severity of illness, APRDRG Mortality risk, Income zip quartile, Primary payer) and 7 situational variables (including Patient Location, Month of the procedure, Hospital Division, Hospital Region, Hospital Teaching status, Hospital Bed size, and Hospital Control).	Preoperative	KPIs: AUC and accuracy. AUC: 0.745 (LSVM). Accuracy: 83.88% (Decision list). For predicting the LOS, the best performing ML models were: LSVM, Decision Lists, CHAID.	LOS is binary with a cutoff of 2 or less days, which is based on the average of the patients in the study; SMOTE was applied to deal with data imbalance; Uses three data subsets (train, test, validate); Patients with deficit information were removed; 80:20 train test split; Similar authors like Zalikha et al. (2023).
(Chen et al., 2023a)	Predicting prolonged LOS based on national patient cohort data (binary variable LOS of 3 days was based on the	Primary TKA patients from US between 2013 and 2020. Extracted	~268,000	4 ML models: ANN, RF, histogram-based gradient boosting, KNN.	Recursive feature elimination based on a rudimentary RF model was done; the most relevant features were age, BMI, ethnicity, pre-operative transfusion, white blood cell, hematocrit, platelet count, operation time, anaesthesia type, and diabetes.	Pre- and perioperative	The best performing model was ANN. Its KPIs were AUC (0.71), calibration plot (slope: 0.82, intercept: 0.03), Brier score (0.089).	Prolonged LOS defined as exceeding the 75th percentile of all LOSs, which was more than 3 days; No revision surgery, thus only primary TKA; removes outliers in features; 80:20 train/test split.

	75th percentile of all LOS).	from ACS-NSQIP.						
(Chen et al., 2023b)	Develop ML models with national-scale data set and test their performance in predicting prolonged LOS.	Primary THA from US between 2013 and 2020. Extracted from ACS-NSQIP.	~246,000	4 ML models: ANN, RF, histogram-based gradient boosting, KNN.	Recursive feature elimination based on a rudimentary RF model was done. The most relevant variables for predicting were age, transfusion after surgery, operation time, BMI, platelet count, white blood cell count, and anaesthesia type.	Pre- and perioperative	The best performing model was ANN. Its KPIs were AUC (0.73), calibration plot (intercept: -0.01, slope: 0.99), Brier score (best: 0.185).	Prolonged LOS defined as exceeding the 75th percentile of all LOSs, which was more than 3 days; 80:20 train test split; mentions that many patient variables were made binary.
(El-Othmani et al., 2022)	Develop, test, and compare ML models to predict postoperative parameters like LOS.	Primary THA from US between 2016 and 2017. Extracted from NIS.	~177,000	4 ML models: LSVM, RF, ANN, XGBoost.	15 variables: 8 patient-specific and 7 hospital-specific variables. Variables most effective for LOS prediction were age, total number of diagnoses, APRDRG mortality, APRDRG severity, hospital division, sex, hospital bed size, race, hospital control, primary payer.	Preoperative	KPIs were AUC (0.744 of LSVM) and accuracy (72.21% of LSVM).	LOS is binary with a cutoff of 2 or less days, which is based on the average of the patients in the study; SMOTE was applied to deal with data imbalance; Uses three data subsets (train, test, validate); Patients with deficit information were removed; 80:20 train test split; Similar authors like Zalikha et al. (2023).
(Li et al., 2022)	Develop a predictive model for LOS	TKA between 2013 and 2014 from a single Singapore centre	~1800	2 ML models: XGBoost, logistic regression	Univariate analysis ($p<0.05$ were selected) is used to identify predictive variables, which were: age, ASA score, diabetes, ischemic heart disease, congestive heart failure, general anaesthesia, operation duration, cerebrovascular accident, creatinine level, race, gender, BMI, hemoglobin, and smoking.	Pre- and perioperative	XGBoost model (AUC: 0.738) performed better than logistic regression (AUC: 0.639) because AUC was higher	Normal and prolonged LOS were differentiated with a binary variable. Long LOS is 6 or more days.
(Wei et al., 2021)	Develop ANN model to determine pre- and perioperative variables to	Primary TKA from 2018 in US. Extracted	~29,000	2 ML models: logistic regression and ANN	Variables with $p<0.2$ (using chi-squared and independent samples t-test) were used for logistic regression, while stepwise logistic regression was also used for filtering, which resulted in: age, race, BMI, dyspnoea status,	Pre- and perioperative	KPI was AUC. The ANN (AUC: 0.801) performed slightly better than the logistic regression	Patients with missing data were excluded; prediction was about whether patient was discharged the same day or not, normal discharge was two to four

	predict same-date discharge	from ACS-NSQIP			functional status, anaesthesia type, operating time, preoperative INR, preoperative sodium, sex, ASA score, hypertension, chronic steroid use, and preoperative haematocrit. Variables used for ANN were the first 9 variables, as well as COPD status and anaemia status		model (AUC: 0.796)	days LOS, and same day discharge was a LOS of 0 days. Thus, LOS was modelled as a binary variable; 60/40 train/test split
(Ramkumar et al., 2019a)	Develop an ANN that predicts the LOS	Primary TKA from 2009 to 2013 in US. Extracted from NIS, as well as an institutional database	~175,000	1 ML model: ANN	15 variables: age, gender, ethnicity, race, APR risk of mortality, APR severity of illness, number of chronic diseases and diagnoses, comorbidity status, type of admission, whether the admission was from the emergency department, whether the admission was during the weekend, hospital type, patient income quartile, and whether patient was transferred from an outside hospital	Preoperative	KPIs were AUC and accuracy. The ANN for predicting the LOS had an internal AUC of 74.8%, an internal accuracy of 75.3%, an external accuracy of 80%, and external AUC of 83.2%	Patients with missing data were excluded, which was 2.5% of the data; LOS determination does not check discharge disposition; LOS outcome variable had two categories: short LOS (1 to 3 days), and long LOS (4 or more days)
(Ramkumar et al., 2019c)	Develop an ANN that predicts the LOS	Primary THA from 2009 to 2011 in US. Extracted from NIS	~78,000	1 ML model: ANN	15 variables: age, gender, ethnicity, race, APR risk of mortality, APR severity of illness, number of chronic diseases and diagnoses, comorbidity status, type of admission, whether the admission was from the emergency department, whether the admission was during the weekend, hospital type, patient income quartile, and whether patient was transferred from an outside hospital	Preoperative	KPIs were AUC and accuracy. The ANN for predicting the LOS had an internal AUC of 82%, an internal accuracy of 75%, an external accuracy of 75.6%, and external AUC of 80.3%	Patients with large amount of missing data were excluded. LOS outcome variable was binary, and threshold was based on the median (4 or more days was long LOS); 90:10 train/test split, stratified k fold (k=10) was used. Five categories of AUC quality.
(Gabriel et al., 2019)	Develop predictive model for identifying patients who will not require a long LOS	Primary THA in US from 2014 to 2016. Extracted from single institution	~1000	3 ML models: Logistic regression with ridge regression, logistic regression with Lasso, RF	Univariable logistic regression to assess associations of variables with LOS. Final model contained 9 variables: age, opioid use, metabolic equivalents score, sex, anemia, chronic obstructive pulmonary disease, hypertension, obesity, primary anaesthesia type.	Preoperative	KPI was AUC. AUC for the best performing model was 0.761 (logistic regression with ridge regression)	Short LOS is defined at 3 or less days. Train/test split is 66:33. Backward stepwise model selection based on AIC was used to construct final logistic regression model.
(Ramkumar et al., 2019b)	Develop ML model to	Primary THA in US from 2012	~122,000	1 ML model: Naive Bayesian model	Final variables: age, race, gender, and comorbidity scores (APR risk of morbidity and APR risk of illness).	Preoperative	KPIs: AUC (0.87) and accuracy (0.83)	LOS categories: 1-2, 3-5, 6+ days.

	predict LOS with big data	to 2016. Extracted from SPARCS						
(Navarro et al., 2018)	Using preoperative big data to predict the LOS	Primary TKA in US from 2009 to 2016. Extracted from SPARCS	~141,000	1 ML model: Naive Bayesian model	Final variables: age, race, gender, and comorbidity scores (APR risk of morbidity and APR risk of illness).	Preoperative	KPIs: AUC (0.7822) and accuracy (0.874)	3:1 train/test split; LOS categories: 1-3, 4-5, 5+ days.

In the following section, we explain the concepts named in Table 2.1. Machine learning (ML) particularly is a recurring term in the literature. It is a scientific field where the goal is to learn a computer how to perform desirable behaviour without giving detailed instructions on how to behave (Colliot, 2023). Essentially, it is about training computers based on input data, so that they can perform helpful tasks when new input data is provided. ML is a part of artificial intelligence (AI). AI is a scientific domain with the goal of making computers perform similar tasks that can be done by animal or human intelligence. Examples of such tasks are recognising images or words in speech. The concept of AI and ML emerged around the 1940s and 1950s. In 1943, scientists devised an artificial neural model. This model is based on the concept of a biological neuron. In 1958, a scientist created the first perceptron, which is an artificial neuron, which was able to recognise images. It uses a set of input data, combines them, transforms them with a mathematical function, and delivers an output. We explain more on the concept of artificial neurons in at the section concerning artificial neural networks in this chapter. ML methods continued to develop throughout the 1980s and 1990s. During this time, scientists worked on developing and improving decision trees and support vector machines. We expand more on these latter concepts later in this chapter. ML can be classified in three main categories, which are supervised learning, unsupervised learning, and reinforcement learning. Figure 2.1 shows an overview of these categories (Hossain et al., 2020, p. 78060). Any other ML methods are variation on these three categories.

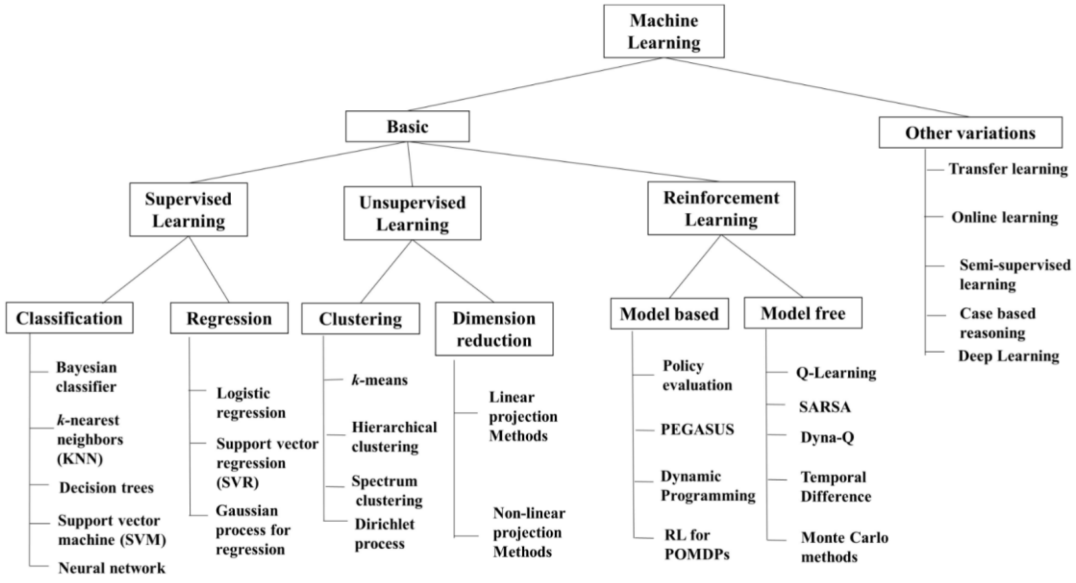


Figure 2.1: Taxonomy of ML (Hossain et al.,2020, p. 78060).

Supervised learning refers to making predictions for a certain output based on given inputs, while unsupervised learning refers to finding structure and relationships when only the input is given, and no specific output variable has to be predicted. The data to train the supervised learning model is the train data, and the data used to make predictions on to test the model is called the test data. Reinforcement learning is characterised by the existence of a decision maker which reacts to a current state, executes an action, and learns from the resulting reward. In reinforcement learning, the decision maker constantly interacts with the environment.

As can be seen in Table 2.1, the majority of the ML methods used are supervised learning. Examples are naive Bayes classifier, logistic regression, and gradient boosting. It makes sense that the literature uses supervised learning methods, because the goal of this research is to predict LOS, which can be achieved with training a model on training data. Figure 2.2 shows a flowchart of how supervised learning works (Osisanwo et al., 2017, p. 129). After the user inputs data, the program processes the data and uses it to train a supervised learning algorithm. The program tests the model and tunes parameters to improve it. Afterwards, the model can be used for classification purposes.

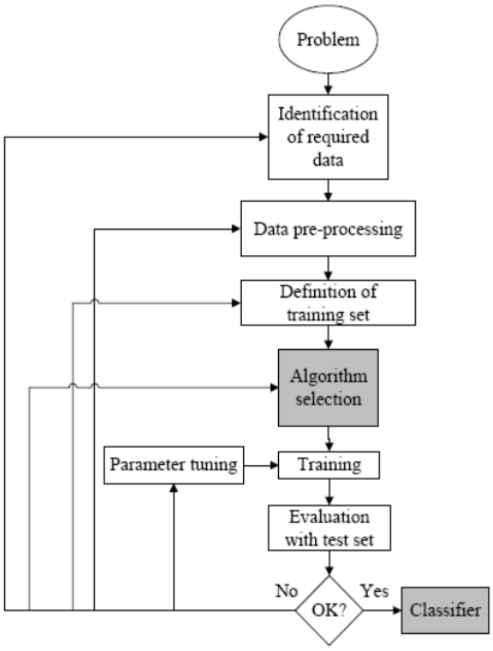


Figure 2.2: General structure of supervised learning (Osisanwo et al., 2017, p. 129).

We found promising supervised ML methods in the literature review, as shown in Table 2.1. We select ML methods who are used in at least two different sources with significantly different research teams. Certain sources have significant overlap in researchers and mainly differ in the fact that one source analyses THA patients and the other analyses TKA patients. We view these sources as one type of research. This means that we consider the following pairs as one type of research:

- Zalikha et al. (2023) and Nham et al. (2023).
- Chen et al. (2023a) and Chen et al. (2023b).
- Ramkumar et al. (2019a) and Ramkumar et al. (2019c).
- Ramkumar et al. (2019b) and Navarro et al. (2018).

Additionally, we also include relatively simple ML methods according to literature, such as k-nearest neighbour and naive Bayes classifier. In the end, we select the following ML methods for this research:

- Logistic regression (LOGR)
- Naive Bayes classifier (NBC)
- K-nearest neighbour (KNN)
- Support vector machine (SVM)
  - Linear support vector machine (LSVM)
  - Support vector machine with radial basis function kernel (SVMR)
- Random forest (RF)

- Extreme gradient boosting (XGB)
- Artificial neural network (ANN)

Logistic regression (LOGR)

Logistic regression is one of the most used methods for predicting a binary outcome variable with a given input data (Fitzmaurice & Laird, 2015). It uses equation [1] where X is a certain input variable value, β_0 is the intercept, β_1 is the coefficient for variable X, and p(X) is the probability that the outcome variable is 1. Determining whether p(X) is high enough to conclude whether the outcome variable is 1 depends on a certain prediction probability cutoff value. For example, if a cutoff value of 0.5 is used, then this means that a value of p(X) or higher will mean that the outcome variable is predicted to be 1. Equation [1] is the formula for logistic regression if only one input variable is used, and it could be extended with multiple input variables and coefficients. Using more than one input variable for logistic regression is called multiple linear regression.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \qquad [1]$$

Naive Bayes classifier (NBC)

The naive Bayes classifier is a relatively simple but powerful ML algorithm (Berrar, 2018). It uses the formulation of the Bayes' theorem to calculate conditional probabilities $p(y_j|\boldsymbol{x_i})$, which is the probability that outcome variable value y_j will be present when a row of input features $\boldsymbol{x_i}$ of instance i is present. Equation [2] shows the corresponding formula. The naive Bayes classifier can be used to predict outcome variable values. If the value of the conditional probability $p(y_j|\boldsymbol{x_i})$ is higher than a certain chosen threshold, then the classifier predicts value y_j.

$$P(y_j|\boldsymbol{x_i}) = \frac{P(\boldsymbol{x_i}|y_j) * P(y_j)}{P(\boldsymbol{x_i})} \qquad [2]$$

K-nearest neighbour (KNN)

The k-nearest neighbour algorithm is a relatively simple ML method which can be used to classify data (Zhang, 2016). The way KNN works is that the training data is mapped as data points in a space, and the result of the outcome variable of each data point is also being registered. Every axis in the space represents a feature. When a prediction is done for a data point from the test data, the KNN algorithm will check the Euclidean distance from the newly added data point to the closest data points from the training data. The amount of training data points to be checked equals parameter k. A KNN algorithm chooses the outcome for a new data point from the test dataset based on the highest number of closest data points from the training dataset. The k parameter affects how the algorithm works significantly. A lower k risks ignoring small patterns, even though these patterns could still be significant. A higher k reduces the impact of random errors on the prediction quality, but this could risk missing small but significant patterns in the data. Figure 2.3 illustrates how the data points are put in a space, where colours are the outcome of a data point, and the axes are features (Zhang, 2016, p. 3). The triangles, which is the train data, are plotted first. Afterwards, the circles are plotted, and their outcome is predicted based on the closest neighbouring train data.

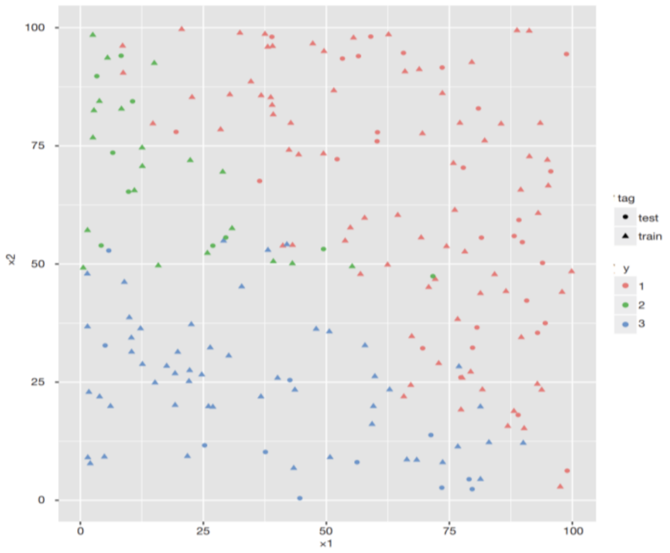


Figure 2.3: Illustration of how predictions of test data is made with KNN (Zhang, 2016, p. 3).

Support vector machine (SVM)

Support vector machine is a ML method which tries to distinguish different outcomes based on features (Pathak, 2020). The method creates a hyperplane and predicts the outcome of data points based on their relative location to this hyperplane, or decision surface. Figure 2.4 illustrates this concept (Mountrakis et al., 2011, p. 248). Data points which fall on one side of the decision surface are predicted to have a certain outcome, while data points on the other side are predicted to have another outcome.

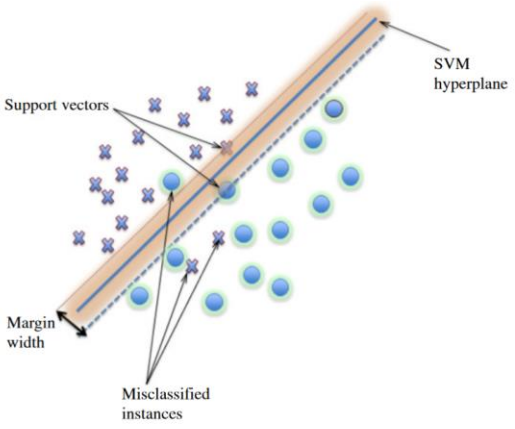


Figure 2.4: Illustration of LSVM example (Mountrakis et al., 2011, p. 248).

The SVM method tries to create an optimal hyperplane, which is based on maximising the distance of the hyperplane to the closest data points on either side of the hyperplane. This distance is called the margin, and the closest data points to the decision surface are called the support vectors. Applying a linear SVM assumes that the data is linearly separable. There are also non-linear SVMs, such as an SVM which uses the radial basis function kernel (SVMR). Such SVMs help classify data which is not linearly separable.

Random forest (RF)

Random forest is a method which can deal well with noise and outliers, is accurate in classification, and avoids overfitting (Liu et al., 2012). Random forest is an ensemble method, which is a method that uses multiple learning algorithms to create a combination of decision trees. A decision tree is a structure that contains nodes which split in other nodes. These nodes have a criterium, and the direction to resume in the decision tree is governed by whether the input data satisfies this criterium or not. Figure 2.5 shows an example of a decision tree where the criterium of the first split is whether the input data has a value higher than 0.62625 for variable LTC4S, and the second split at node 2 is governed by the criterium whether the variable CA2 has a value higher than 0.90977 (Myles et al., 2004). The final nodes, called leaf nodes, contain the prediction of the outcome variable, which is in Figure 2.5 either L or M.

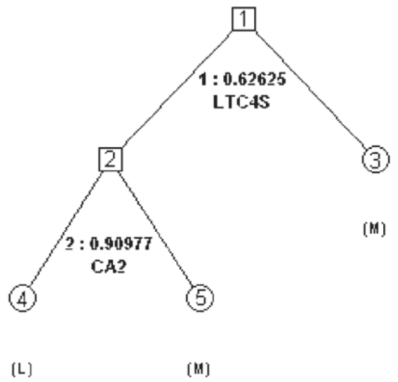


Figure 2.5: Illustration of decision tree example (Myles et al., 2004, p. 276).

A random forest creates decision trees where each one of them is built on a subset of the training data. Furthermore, for every decision tree, only a subset of the total amount of features is included. A random forest model will base its final prediction on a majority vote depending on the resulting individual predictions of all predictions of the decision trees which are part of the random forest. Figure 2.6 shows a schematic to illustrate the random forest method, where T_k signifies a subset of the training data created by sampling with replacement (Liu et al., 2012, p. 247).

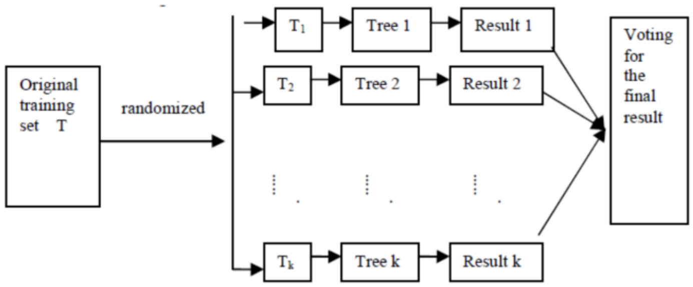


Figure 2.6: Schematic of random forest concept (Liu et al., 2012, p. 247).

Gradient boosting (XGB)

Gradient boosting is an ensemble method like random forest (Natekin & Knoll, 2013). An ensemble method uses multiple instances of a base-learner to produce better predictions. An example of base learners could be decision trees. Instead of creating different decision trees at the same time, like random forest does, gradient boosting builds the decision trees sequentially. Every new decision tree model is trained pertaining to the error of all the previous models so far, which is done to improve the prediction accuracy. A gradient-descent formulation is used to determine what the structure of the next decision tree in the sequence should be. This formulation makes sure a decision tree is created where the chosen loss function, which is based on the prediction error, descents the fastest, which is achieved by analysing the gradient of the loss function. Figure 2.7 shows a figure which illustrates how a gradient boosting machine is trained (Nhat-Duc & Van-Duc, 2023, p. 3). New decision trees are subsequentially added and aim to reduce the prediction error of the previous results. Like random forest, the final prediction of gradient boosting is based on a majority vote across all decision trees that are created.

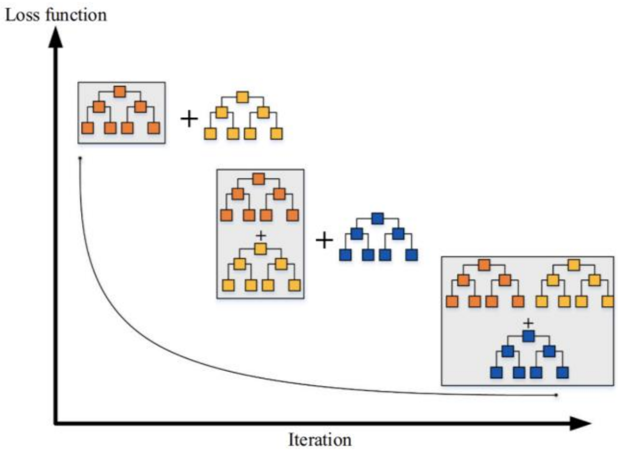


Figure 2.7: Illustration of how a gradient boosting model is trained (Nhat-Duc & Van-Duc, 2023, p. 3).

Histogram-based gradient boosting is an adjusted version of gradient boosting, where the continuous values of features are put into ranges and aggregated into separate bins. The number of bins could be set to 255 for example (Nhat-Duc & Van-Duc, 2023). The bins can be used to create a histogram to visualise the distribution of the continuous input features. The main advantage of histogram-based gradient boosting is that it reduces the computational cost of the method.

Extreme gradient boosting (XGBoost) is another adjusted version of gradient boosting (Chen & Guestrin, 2016). It introduces additional algorithms and data techniques. XGBoost handles sparse data better and makes use of approximate learning. The adjustments introduced in XGBoost has the main advantage that the computation time is significantly low compared to other similar methods.

Artificial neural network (ANN)

An artificial neural network is characterized by three types of neurons, which are input neurons, output neurons, and hidden neurons. Figure 2.8 shows an example of an ANN, where x signifies the input layer, h signifies the hidden layer, and y signifies the output layer (Bougrain, 2004, p. 348). An ANN can have multiple hidden layers. Each input node receives information of the features and passes it on to hidden nodes via connections. These connections have weights. The hidden nodes perform calculations, such as taking the weighted sum of the input values plus a value called the bias, transform this value with a function called the activation function, and pass this final value to other nodes through connections. This flow of information could pass multiple hidden nodes before it reaches the output nodes. The output nodes signify outcome variables. For example, suppose an ANN has an output node where a value of 1 signifies a long LOS and a value of 0 signifies a short LOS. If a certain collection of features is inputted in the input nodes, and the final output value is 0.8 after calculations are done in the hidden layer, then this means that the ANN predicts the likeliness of a long LOS to be 0.8, which means that the ANN suggests that the outcome is much more likely to be a long LOS. For the ANN, the choice of the weights of the connections, activation functions, and bias values should be optimised to make the predictions as good as possible.

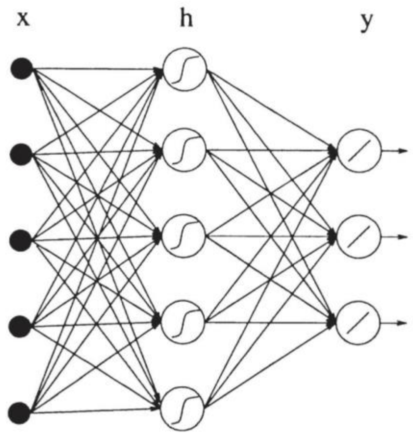


Figure 2.8: ANN illustration (Bougrain, 2004, p. 348).

Feature selection methods (FSM)

The variables used in the literature as input for the ML models differ across the literature, as shown in Table 2.1. To make sure the promising prediction variables are included in the ML models used in this thesis, we aim to incorporate as much variables as possible. Furthermore, it could be the case that the input data of the research of this thesis contains variables which are useful for LOS prediction but are not extensively researched in the literature, which further highlights the advantages of considering as much input variables as possible. However, using too many features in a prediction model could make predictions worse and could cause overfitting (Jović et al., 2015). Furthermore, having too many variables could be computationally expensive. Having many variables is also less convenient from a customer's perspective, as this would require extensive variable collection. Feature selection methods (FSMs) could help filter down the most promising variables before we train the ML models. Figure 2.9 shows that feature selection methods can be classified in the following four categories: filter methods, wrapper methods, embedded methods, and hybrid methods (Abiodun et al., 2021, p. 15095).

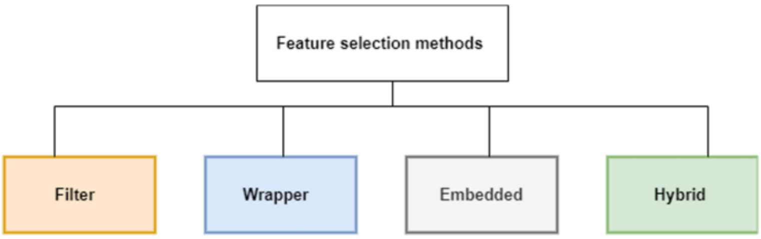


Figure 2.9: Categories of feature selection methods (Abiodun et al., 2021, p. 15095).

For filter methods, variables are selected based on a certain performance measure, which is not influenced by the used ML modelling method. Only the variables which satisfy certain criteria are included for modelling. These criteria can be statistical measures. An example of this would be Pearson correlation coefficients. High correlation between the features and the outcome variable is desirable, while high correlation between the features themselves should be avoided

(Witten et al., 2016). Similarly, chi-square tests can also be used to select features, where features which are significantly dependent on the outcome variables would be selected (Jović et al., 2015).

Wrapper methods create subsets of the data by only including a part of the features and then test the predictive performance of these subsets on the used machine learning modelling algorithm. This also means that the feature subsets become biased towards the modelling algorithm used, which is undesirable. Still, wrapper methods generally yield better feature subsets than filter methods, but wrapper methods are significantly slower. Examples of wrapper methods are sequential algorithms, such as forward or backward sequential feature selection (Abiodun et al., 2021).

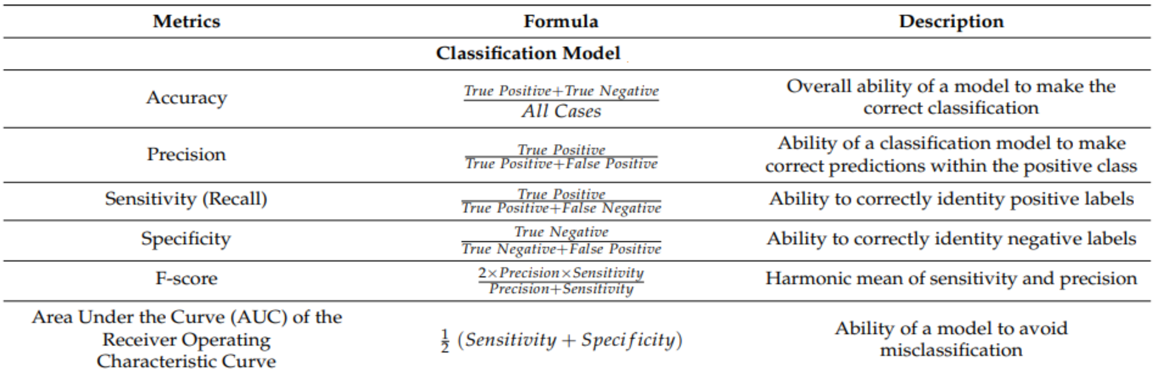
Embedded methods execute the feature selection method while the modelling algorithm is running (Jović et al., 2015). Certain embedded methods utilise regularisation, which minimises the fitting errors while making sure that feature coefficients are close to zero. An example of this is as the Lasso method. Another example of an embedded feature selection method is ranking the importance of all features and then selecting the features with an importance index higher than a certain threshold (Genuer et al., 2010).

Hybrid methods combine the most beneficial properties of both wrapper and filter methods (Jović et al., 2015). The filter method is executed first to reduce the total amount of features. Afterwards, the wrapper method is executed so that the best feature subset is selected. Hybrid methods often achieve the high efficiency of filter methods, while also attaining the high accuracy of wrapper methods.

Key performance indicators (KPIs)

Key performance indicators can be used to measure the performance of a ML prediction model. Table 2.2 proposes such KPIs (Lee et al., 2021). Classification models refers to models whose outcome variable are categorical, and non-classification models refers to models with a continuous outcome variable. As mentioned before, the LOS is modelled as a categorical variable in the literature, which implies that the KPIs for classification models are suitable for our research. This is in line with what the literature in Table 2.1 shows, as the AUC and accuracy are often used as KPIs. For these KPIs, higher values mean better model performance.

Table 2.2: KPIs for classification models (Lee et al., 2021, p. 7).



Metrics	Formula	Description
	Classification Model	
Accuracy	$\frac{True\ Positive + True\ Negative}{All\ Cases}$	Overall ability of a model to make the correct classification
Precision	$\frac{True\ Positive}{True\ Positive + False\ Positive}$	Ability of a classification model to make correct predictions within the positive class
Sensitivity (Recall)	$\frac{True\ Positive}{True\ Positive + False\ Negative}$	Ability to correctly identity positive labels
Specificity	$\frac{True\ Negative}{True\ Negative + False\ Positive}$	Ability to correctly identity negative labels
F-score	$\frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity}$	Harmonic mean of sensitivity and precision
Area Under the Curve (AUC) of the Receiver Operating Characteristic Curve	$\frac{1}{2}\,(Sensitivity + Specificity)$	Ability of a model to avoid misclassification

The AUC is a KPI which is measure as the area under the curve of a receiver operator characteristic (ROC) curve. The ROC curve (Hastie et al., 2021) is a graphic which visualises the sensitivity and specificity of a tested prediction model depending on the prediction probability cutoff value used. Every point in the ROC curve corresponds to a certain cutoff value. Every cutoff value also has a different accuracy, which makes it relatively difficult to assess a model's performance with accuracy. To evaluate the whole ROC curve, it is more favourable to use the AUC instead (Bradley, 1997). Thus, we use the AUC as main KPI for our research. An example of an ROC curve is shown in Figure 2.10, where the true positive rate is the sensitivity, and the false positive rate equals one minus the specificity (Hastie et al., 2021, p. 151). The AUC values in literature range from 0.69 to 0.83 for TKA patients, and from 0.73 to 0.87 for THA patients.

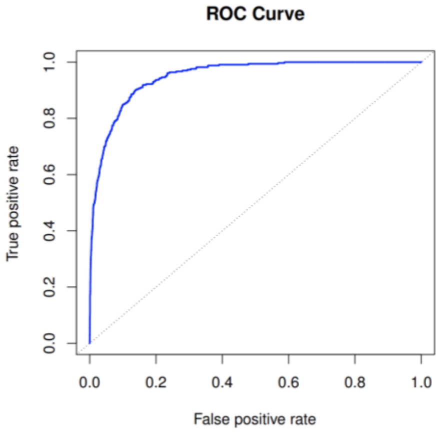


Figure 2.10: ROC curve illustration (Hastie et al., 2021, p. 151).

In this research, we define the occurrence of a long LOS as a positive label. The sensitivity is defined as the probability of a ML model to predict a long LOS given that the patient experiences a long LOS. For example, a high sensitivity of 0.8 means that 80% of the patients who experience a long LOS are predicted by the model to experience a long LOS. This would mean that the model is relatively good at identifying long LOS patients. In contrast, the specificity is the probability that the model predicts a short LOS given that the patient experiences a short LOS. A high specificity means that the model is relatively good at identifying patients who experience a short LOS.

2.2 Summary

In Chapter 2, we review the academic literature which performed LOS predictions for primary THA/TKA patients. The literature trains ML models on training data and then tests their models on separate data to measure the performance. We select eight ML models. We also select three FSMs which help reduce the required number of input features for the ML models. We use AUC as the main performance metric for the models. We have a selection of ML models to use for our research, and the next step is to gather data which can be used to train and test our models.

Chapter 3: Data collection and analysis

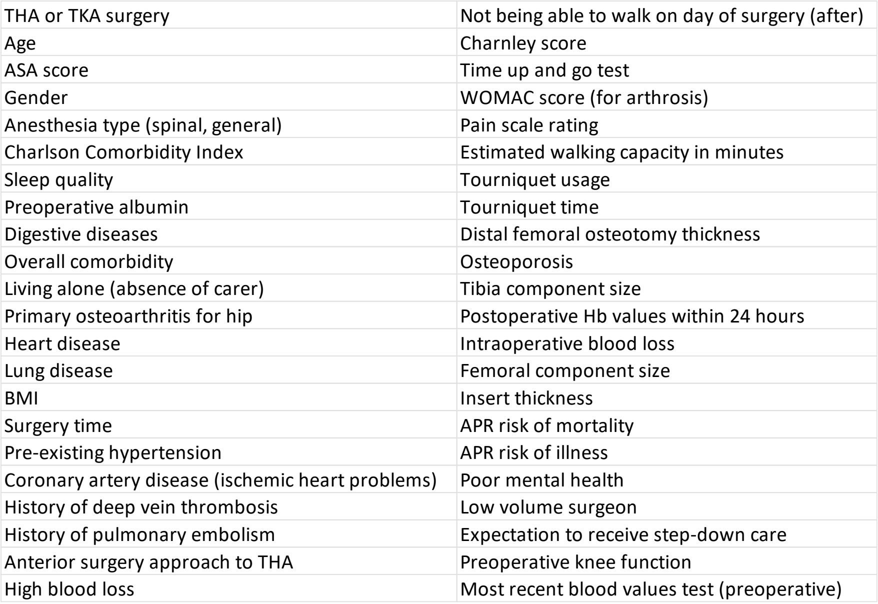
In this chapter, we answer the research questions related to the collection, analysis and processing of the data which is required as input for ML models. We answer the following main research question: *What input is required for LOS prediction methods?* In Section 3.1, we explain how we prepare for the data request from the HiX database. We discuss and interpret the selected variables, the received variables, and the received datasets. In Section 3.2, we prepare, process, and clean the data. This consists of formatting data and deleting parts of the data. In Section 3.3, we visualise certain variables and analyse them to gain initial insight in the distributions of the variables. Finally, in Section 3.4, we perform data imputation to make sure the data contains no missing values. This makes sure the data is suitable to be used as input for ML models, which will be done in Chapter 4.

3.1. Data collection

To gather data, which is necessary for training ML models, we require patient-related data from the HiX database. One important aspect which determines whether data is suitable for a potential data extraction, is whether the data is registered in the form of *written text*, or in the form of *structured data*. The written text refers to data which is typed in a text section without any extra structure from HiX itself. The structured data refers to data which has a data box dedicated to specifically one type of data, and thus has a restricted type of data registration. An example of this are numbers, as well as answers with a finite number of possible values, such as *yes/no* answers. Interpreting and structuring data in the form of written text is a significantly large workload, and there is no guarantee whether these variables are useful. Thus, analysing data in the form of written text is outside of the scope of this research.

For the HiX data request, we collect relevant variables which are suitable for predicting the LOS for primary TKA/THA patients. We select variables based on literature. In addition to the sources in Table 2.1, we extract relevant variables from Ding et al. (2022), Ding et al. (2019), Elings et al. (2014), Han et al. (2021), Johannesdottir (2022), Lakomkin et al. (2017), Ong & Pua (2013), Sibia et al. (2016), Sibia et al. (2017), and Van der Sluis (2018). Table 3.1 shows the overview of variables selected from literature. Unfortunately, not all these selected variables are available in HiX, and thus we are forced to exclude a part of these variables from the data request. Furthermore, plenty of variables are registered in HiX as written text, and thus we also exclude these variables. Our selection of variables mainly consists of preoperative variables. However, it also contains some perioperative variables, even though the aim of this research is to only make predictions by using preoperative variables. In case we do not receive enough preoperative variables from the HiX data request, we could also use perioperative variables to predict the LOS. However, this is undesirable as the goal of the research is to make LOS predictions before the surgery, and not during or after the surgery.

Table 3.1: Overview of variables deemed important for LOS prediction according to literature.



THA or TKA surgery	Not being able to walk on day of surgery (after)
Age	Charnley score
ASA score	Time up and go test
Gender	WOMAC score (for arthrosis)
Anesthesia type (spinal, general)	Pain scale rating
Charlson Comorbidity Index	Estimated walking capacity in minutes
Sleep quality	Tourniquet usage
Preoperative albumin	Tourniquet time
Digestive diseases	Distal femoral osteotomy thickness
Overall comorbidity	Osteoporosis
Living alone (absence of carer)	Tibia component size
Primary osteoarthritis for hip	Postoperative Hb values within 24 hours
Heart disease	Intraoperative blood loss
Lung disease	Femoral component size
BMI	Insert thickness
Surgery time	APR risk of mortality
Pre-existing hypertension	APR risk of illness
Coronary artery disease (ischemic heart problems)	Poor mental health
History of deep vein thrombosis	Low volume surgeon
History of pulmonary embolism	Expectation to receive step-down care
Anterior surgery approach to THA	Preoperative knee function
High blood loss	Most recent blood values test (preoperative)

While investigating which variables can be extracted from HiX, we also found variables which are in the form of structured data instead of written text and decided to also add these variables to the data request. Primarily questionnaires, such as the anamnesis questionnaire, contain a high amount of structured data, which also includes relevant variables such as the presence of comorbidities. An overview of the requested variables is shown in Table 3.2. The total number of requested variables is 219. The green variables are the variables which we received in the data request, which are 39 variables in total.

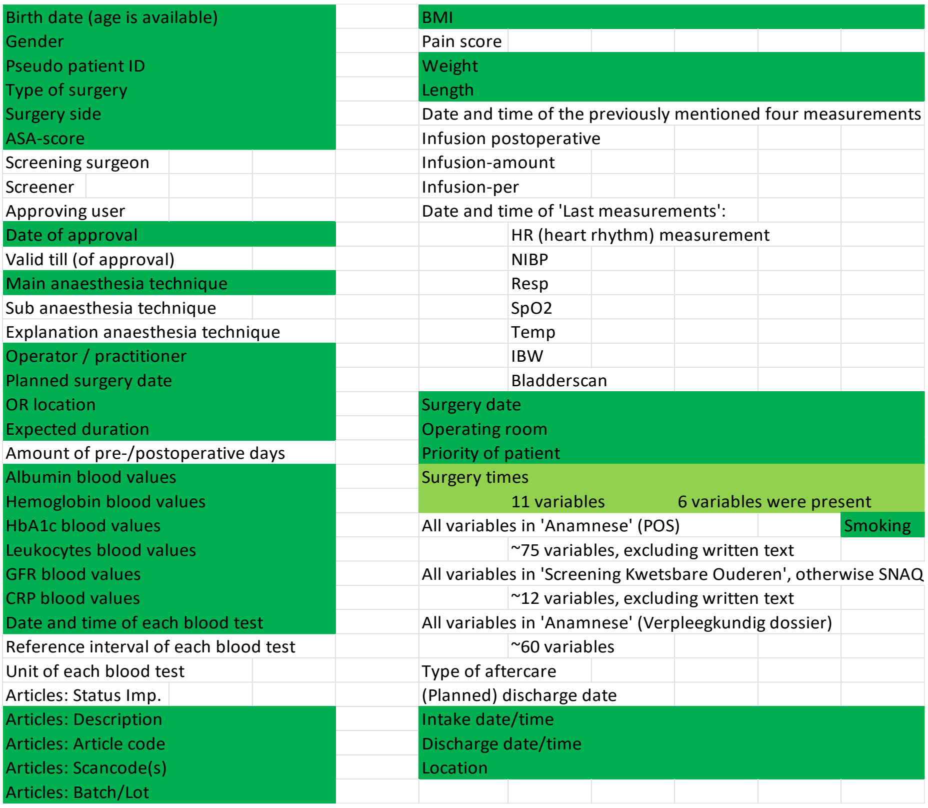
Once the data of the data request arrived, we gained access to two datasets:

- A dataset with various variables which we requested contains 200 columns and consists of:
  - A part containing data from HiX.
  - A part with data from an operating room overview database.
  - A part with data from LROI.
- A dataset containing the variables related to blood tests consisted of 60 columns with information about:
  - Blood values.
  - Blood measurement dates.
  - Medical personnel who performed the measurements.

After combining both datasets, the size of the total dataset is 260 columns/variables and 5391 rows/surgeries, which are the names of the variables. Of the 260 variables we received, we requested 99 columns. However, 51 columns all referred to the type of prosthesis used in the surgery, which concerns the *Articles* variable as shown in Table 3.2. Furthermore, most of the 260 variables turned out to be post- or perioperative variables, which is suboptimal because the purpose of this research is to make a preoperative prediction of the LOS by using preoperative variables only.

There are also a few variables present in the data request which are useful for predicting the LOS according to academic literature but were not requested in the HiX data request. These variables are the surgical approach and the Charnley score, which are variables that could not be extracted from HiX but are present in the LROI part of the received dataset.

Table 3.2: Overview of variables requested for HiX data request, where the green variables are present in the received datasets.



🟩 Birth date (age is available)	🟩 BMI			
🟩 Gender	Pain score			
🟩 Pseudo patient ID	🟩 Weight			
🟩 Type of surgery	🟩 Length			
🟩 Surgery side	Date and time of the previously mentioned four measurements			
🟩 ASA-score	Infusion postoperative			
Screening surgeon	Infusion-amount			
Screener	Infusion-per			
Approving user	Date and time of 'Last measurements':			
🟩 Date of approval		HR (heart rhythm) measurement		
Valid till (of approval)		NIBP		
🟩 Main anaesthesia technique		Resp		
Sub anaesthesia technique		SpO2		
Explanation anaesthesia technique		Temp		
🟩 Operator / practitioner		IBW		
🟩 Planned surgery date		Bladderscan		
🟩 OR location	🟩 Surgery date			
🟩 Expected duration	🟩 Operating room			
Amount of pre-/postoperative days	🟩 Priority of patient			
🟩 Albumin blood values	🟩 Surgery times			
🟩 Hemoglobin blood values		🟩 11 variables	🟩 6 variables were present	
🟩 HbA1c blood values	All variables in 'Anamnese' (POS)			🟩 Smoking
🟩 Leukocytes blood values		~75 variables, excluding written text		
🟩 GFR blood values	All variables in 'Screening Kwetsbare Ouderen', otherwise SNAQ			
🟩 CRP blood values		~12 variables, excluding written text		
🟩 Date and time of each blood test	All variables in 'Anamnese' (Verpleegkundig dossier)			
Reference interval of each blood test		~60 variables		
Unit of each blood test	Type of aftercare			
Articles: Status Imp.	(Planned) discharge date			
🟩 Articles: Description	🟩 Intake date/time			
🟩 Articles: Article code	🟩 Discharge date/time			
🟩 Articles: Scancode(s)	🟩 Location			
🟩 Articles: Batch/Lot				

3.2. Data preparation

After collecting the data, the next step is to process the data so that it can be used for machine learning models. The main goal is to make sure that the dataset has no missing values, without sacrificing too much data. This can be done by either deleting rows or columns with missing data, or by data imputation. Rows and columns can only be used for machine learning if they contain no missing values. A downside of data imputation is that it creates bias (Lee & Huber, 2021), and a downside of deleting rows or columns is that we lose useful data. We start with the combined dataset of 5391 rows and 260 columns. We refer to the missing value percentage as NA%. The initial NA% of the data is 49%. We import the dataset to the programming language R, and to prepare the data we execute the following steps:

1. Remove variables with only missing values, which are *NA* values.
2. Remove variables which do not appear to have an impact on the LOS according to the expert opinion of an OCON anaesthesiologist.
3. Remove post- and perioperative variables. We are left with 5391 rows and 45 columns. The new NA% becomes 30%.
4. Combine columns which are about the same variable (e.g. ASA scores, Charnley scores, and surgeons are scattered across columns), and calculate the LOS and LongLOS variables for each surgery. We are left with 5391 rows and 38 columns.
5. Remove patients which do not seem to be primary TKA/THA patients or have clear errors in their data. We are left with 5379 rows and 38 columns.

6. Remove patients where LOS calculation is impossible (missing discharge date/time), which is approximately 26% of the data. We are left with 3974 rows and 38 columns. The new NA% becomes 22%. We make a boxplot of the remaining data where we plot the NA fraction of each patient, which Figure 3.1A shows.
7. Calculate the NA% of each remaining variable. Table 3.3 shows these values.
8. Data imputation could introduce bias (Lee & Huber, 2021). In literature, there is no clear threshold when there is too much missing data to use data imputation, but we use a threshold of 95% as a starting point. We consider variables with an NA% of higher than 95% too high and we remove them. We are left with 3974 rows and 30 columns. The new NA% of the total dataset becomes 1.3%.
9. Calculate the NA% of each surgery and make a boxplot, as Figure 3.1B shows. The boxplot shows that there are outliers with a significantly higher NA% than the rest of the patients. However, we keep these patients in the data because imputation generally outperforms ignoring data (Van der Heijden et al., 2006).
10. Plot a selection of intuitive variables and leave out variables such as blood values.
11. Remove postoperative variables which we used for LOS calculation, such as the time to bed department, discharge date, and discharge time. We also remove the surgery date and pseudo_id variables from the dataset. We are left with 3974 rows and 25 columns. The new NA% of the total dataset becomes 1.6%.
12. Impute the remaining missing data with the MICE data imputation method, which we do in Section 3.4.

The initial NA% of the combined dataset of 5391 rows/surgeries and 260 columns/variables is 49%. After removing variables with only missing values, removing peri- and postoperative variables, and the variables which are not deemed important according to expert opinion, the NA% becomes 30%. Furthermore, we categorise GFR blood values in four categories, which are in line with literature (Neuen et al., 2018; Wanner et al., 2018). We consider missing values as data that has been measured but not registered in the database. The only exception to this rule is HbA1c blood results. According to expert opinion, medical staff only measure HbA1c if the patient has diabetes. Thus, we assume that a missing value for HbA1c means that the patient does not have diabetes, while a filled in value means that the patient has diabetes.

In step 5, we remove patients with clear errors in the data and patients who seem to have undergone a different surgical procedure than a primary THA/TKA. Various columns in the received dataset contain information in the form of written text with information about the surgical procedure, and we use these variables to determine whether a different surgery was performed.

In step 6, we remove patients where LOS calculation is impossible, and the new NA% becomes 22%. We calculate the LOS of patients by analysing the time difference between their postoperative arrival at the bed department and their discharge timing. We remove patients who contain faulty or missing data concerning these times. Afterwards, we make a boxplot of the NA fraction of each patient, as is shown in Figure 3.1A. The NA fraction is a decimal value where a value of 1 means 100%. The boxplot contains whiskers which are 1.5 times the interquartile range. However, the interquartile range, as well as the whiskers are all situated at an NA fraction of 0.21, which shows that the vast majority of the patients have this NA fraction value.

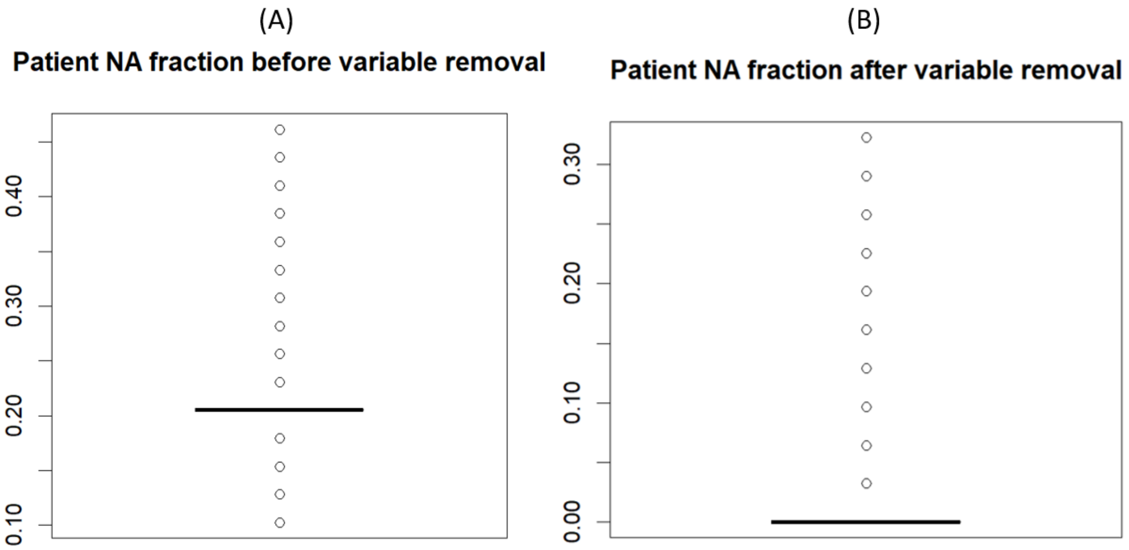


Figure 3.1: (A) Boxplot of NA fraction of patients in step 6, and (B) boxplot of NA fraction of patients in step 9.

We calculate the NA% per variable in step 7. We do this to analyse whether missing values are clustered around certain variables. If this is the case, then we could remove a relatively small number of variables while reducing the NA% significantly. The sorted table with the NA% of each of the 38 variables in this step is shown in Table 3.3. Only eight variables have an NA% of 95% of more, while all other variables have an NA% of approximately 5% or less. This means that a large part of the missing values in the dataset is clustered around the variables with the highest NA%. Since data imputation introduces bias (Lee & Huber, 2021), we put a threshold of 95% for when a variable has too much missing data. We remove variables with an NA% of 95% or more, and the new mean NA% becomes 1.3%. According to literature, most of the removed variables in this step do not seem to be very important for LOS prediction, except for the albumin blood levels. Certain blood values, such as cholesterol, albumin, leukocytes, and CRP, have high NA% values because OCON does not measure every blood value for each patient.

Table 3.3: NA fraction of each variable in step 7, where the orange variables are variables with an NA% of 95% or more.

Variable	NA fraction		Variable	NA fraction		Variable	NA fraction
Buitenlandse.Postcode	1.000		UITSLAG_Natrium	0.048		Elective_patient	0.000
Andere.voorgaande.operaties	1.000		UITSLAG_Hemoglobine	0.043		Surgery.Outside.OfficeHours	0.000
UITSLAG_Cholesterol	0.999		Prosthesis_type	0.028		Time_to_bed_department	0.000
UITSLAG_Albumine	0.995		ASA_score	0.016		Discharge_date	0.000
UITSLAG_Leukocyten	0.985		Planned_anaesthesia	0.002		Discharge_time	0.000
Lengte	0.958		BMI	0.002		pseudo_id	0.000
Gewicht	0.958		Smoking	0.001		Surgery_date	0.000
UITSLAG_CRP	0.952		Gender	0.000		Age	0.000
UITSLAG_Trombocyten	0.055		Charnley.score	0.000		UITSLAG_HBA1C	0.000
UITSLAG_Ureum	0.054		Surgery_side	0.000		Surgeon	0.000
UITSLAG_Kreatinine	0.050		Surgical_approach	0.000		LOS	0.000
UITSLAG_GFR	0.049		Diagnosis	0.000		LongLOS	0.000
UITSLAG_Kalium	0.048		Type.Surgery	0.000			

In step 9, we again plot a boxplot of the NA fraction of the patients to see the impact of removing the variables in step 8. The boxplot is shown in Figure 3.1B. The median, quartiles, and the whiskers all are situated at an NA% of 0%, while patients with a higher NA% are indicated as outliers. Approximately 10% of the patients have an NA% of more than 0%, which means in this case that 10% of the patients are outliers. Imputation generally is better than ignoring data (Van der Heijden et al., 2006), which means that it would be more beneficial to use imputation on the data instead of deleting the patients. Thus, we leave these patients in the data.

3.3. Data analysis

To recap, we started with 5391 surgeries and 260 variables with an NA% of 49%. The remaining dataset consists of 3974 surgeries and 30 variables with an NA% of 1.3%. In Section 3.3, we execute step 10, and we analyse the raw data and visualise the distributions of variables in three categories: patient data, LOS data, and procedure data.

3.3.1. Patient data analysis

The most intuitive variables concerning the patient itself are plotted in this section. Figure 3.2 shows the bar charts or boxplot of the patients' BMI, gender, age, type of surgery, Charnley score, and ASA score. The Charnley score classifies patients based on how much they are affected by arthrosis or other joint-related diseases.

The median of the BMI is 27.4 and 50% of the patients lie between a BMI of 24.8 and 30.7. There are quite some outliers with a BMI higher than the upper whisker value of the box plot. The patients are mostly female, while approximately 40% is male. The median of the age is 70, while 50% of the patients have an age between 64 and 76 years old. The age has mainly outliers of patients who are younger than the lower whisker value, and barely outliers older than the upper whisker value. The performed surgeries are mainly THAs, which is 56% of the surgeries. Most patients have a Charnley score of *A*, which indicates that one joint is affected by arthrosis. More severe Charnley scores have less patients compared to less severe Charnley scores. Finally, an ASA score of 2 is most common, whereas an ASA score of 1 is the least common.

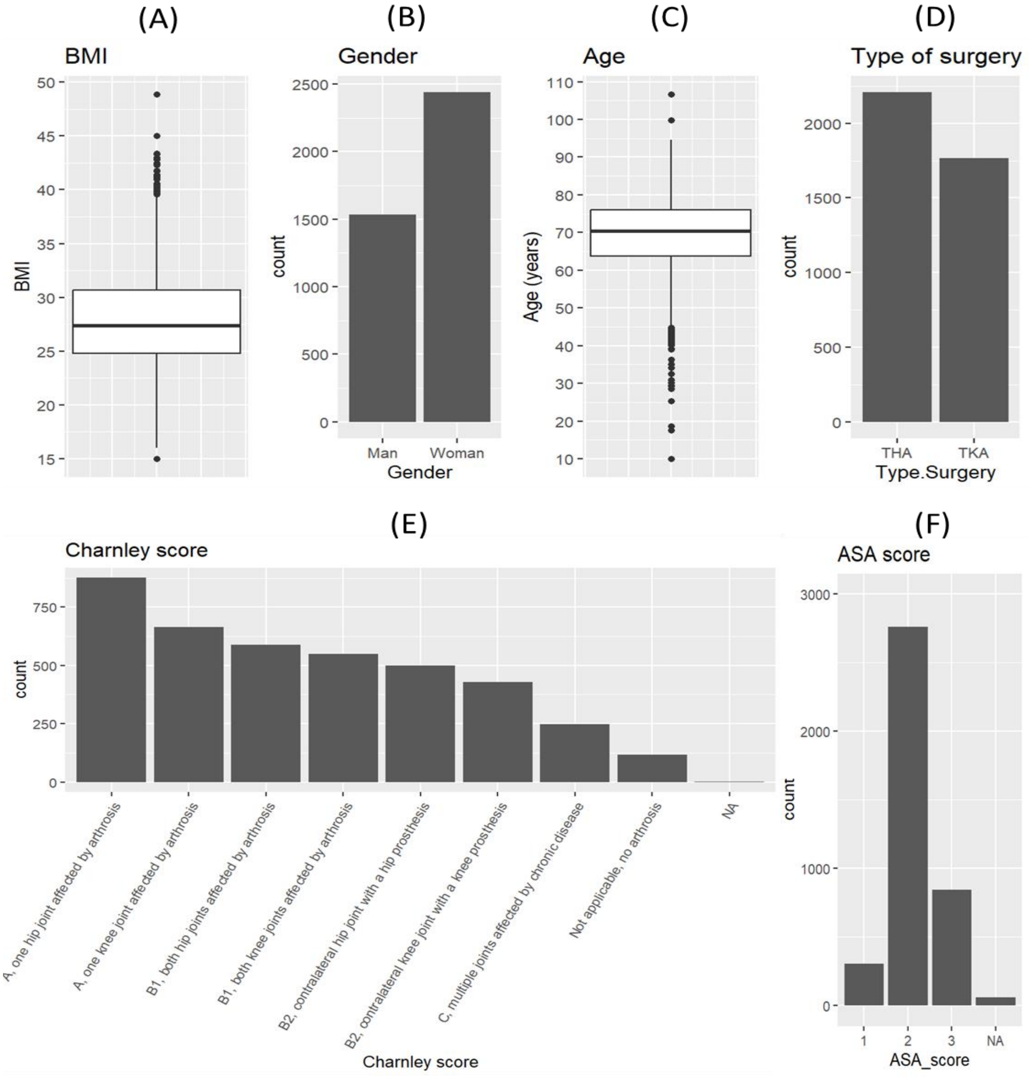


Figure 3.2: Patient data of (A) BMI boxplot, (B) gender bar chart, (C) age boxplot, (D) type of surgery bar chart, (E) Charnley score bar chart, and (F) ASA score bar chart.

To gain more insight in the correlation between certain variables, we make combination plots by using variables such as BMI, age, gender, type of surgery. Figure 3.3 shows these plots. There does not seem to be a lot of difference in the median between men who underwent a THA or TKA. However, this BMI difference becomes bigger for women who underwent a THA or TKA. THA patients seem to have more outliers with an extra high BMI compared to TKA patients. The age distribution does not seem to differentiate a lot between genders and type of surgery. However, there seems to be more outliers, which are younger patients, for THA patients than TKA patients. Concerning Figure 3.3C, there does not seem to be a clear correlation between BMI, age, and type of surgery.

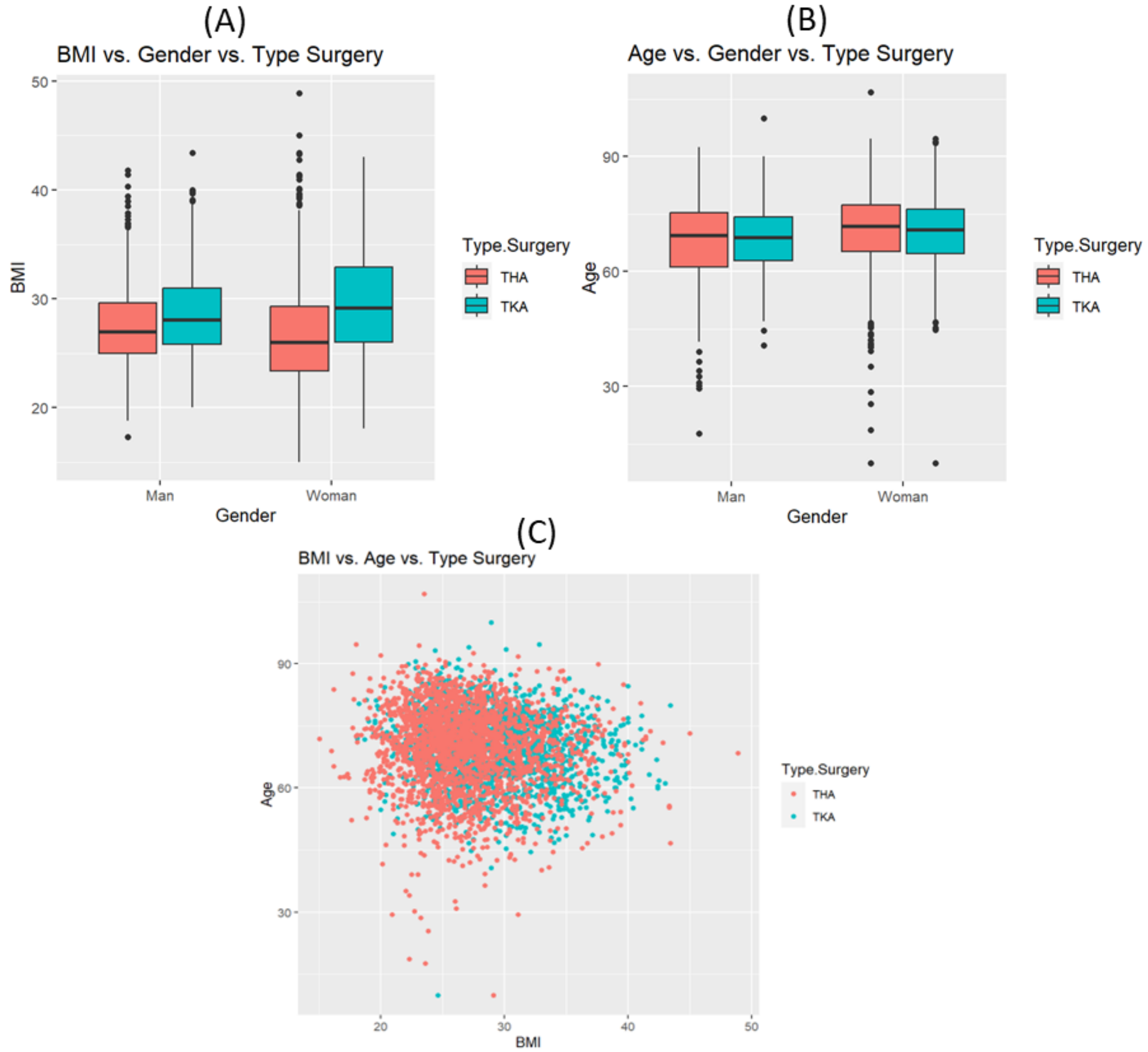


Figure 3.3: Combination plots of: (A) boxplots that show the correlation between BMI, gender, and type of surgery; (B) boxplots that show the correlation between age, gender, and type of surgery; and (C) scatter plot which shows the correlation between BMI, age, and type of surgery.

3.3.2. LOS data analysis

This section is about variables related to the LOS. As mentioned before, the LOS is calculated as the difference between the postoperative arrival time of a patient at the bed department, and the discharge timing. For example, if a patient arrives at the bed department at 11:00 on Monday, and is discharged Wednesday 17:00, then the LOS is 2 days plus 6 hours, which is 2.25 days. Figure 3.4 shows boxplots of the distribution of the LOS, as well as the LOS distribution when viewing type of surgeries separately. The median of the LOS is around 2 days, while 50% of the data lies between approximately 1.5 and 3 days. However, there are plenty of outliers with a much higher LOS. The LOS does not seem to differ by a lot between type of surgeries. Their median values are approximately 2 days. However, the interquartile range is slightly different. TKA patients are unlikely to experience a LOS less than 2 days, because 75% of the TKA patients have a LOS of at least 2 days. In contrast, 75% of the THA patients have a LOS of at least 1.25 days, which shows that THA patients tend to recover faster than TKA patients.

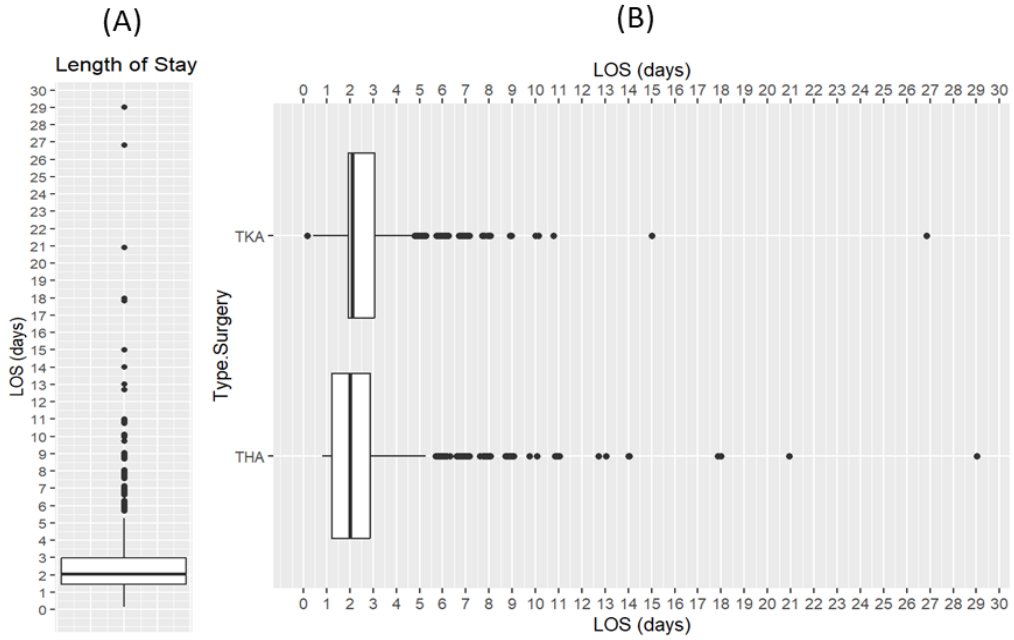


Figure 3.4: Data visualisations of (A) boxplot of length of stay (LOS), and (B) boxplots to show the correlation between LOS and type of surgery.

As mentioned in Chapter 2, academic literature consistently tries to predict a categorical long LOS variable, instead of an exact numerical LOS variable. To determine whether a patient experienced a long LOS, the patient should have a LOS above a certain threshold. With the help of an OCON anaesthesiologist, we choose a long LOS threshold based on the 75th percentile value of the LOS data, which is 3 days. This threshold is also in line with literature (Chen et al., 2023a; Chen et al., 2023b). We depict the long LOS as a binary variable LongLOS, where 1 signifies a long LOS, which means that a patient experienced a LOS of at least 3 days. Figure 3.5 indicates that the majority of the patients experience a LOS of less than 3 days. As expected, 75% of the patients experience a short LOS because the threshold value is based on the 75th percentile value. The proportion of patients experiencing a long LOS compared to patients who experience a short LOS differs per surgery type. This proportion is 28% for TKA patients, and it is 23% for THA patients. This means that TKA patients are more likely to experience a LOS of at least 3 days.

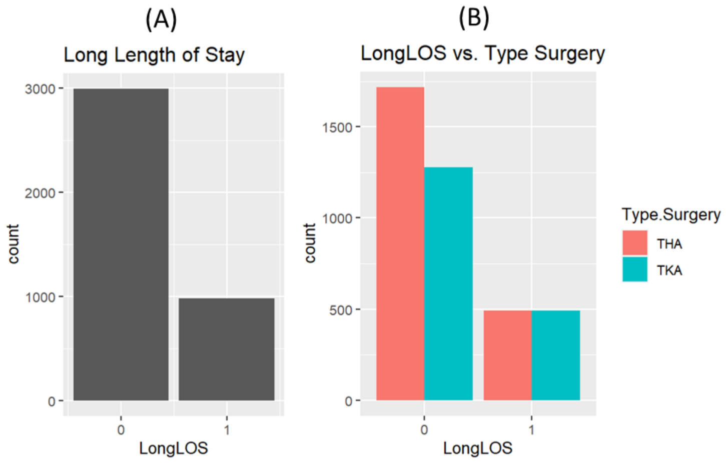


Figure 3.5: Data visualisations of (A) bar chart of whether a patient experienced a long length of stay, and (B) bar chart which shows the correlation between type of surgery and whether a patient experienced a long LOS.

3.3.3. Procedure data analysis

In this section, we cover the variables which concerns the surgical procedure. The graphs for these variables are shown in Figure 3.6. The surgeries performed per surgeon are not the same for all surgeons. Certain surgeons perform more surgeries in total than others, and certain surgeons perform more THA than TKA surgeries and vice versa. In our remaining data, there are 14 surgeons in total. The top three surgeons with the most primary THA/TKA surgeries done performed 37% of the total amount of primary THA/TKA surgeries. The surgeons ranked fourth and fifth in terms of the number of surgeries performed, perform only THA surgeries. Most of the surgeries use spinal anaesthesia, which is 92% of all surgeries. The surgery side does not differ much between patients. Approximately 52% of the surgeries are performed on the right side of a patient's body.

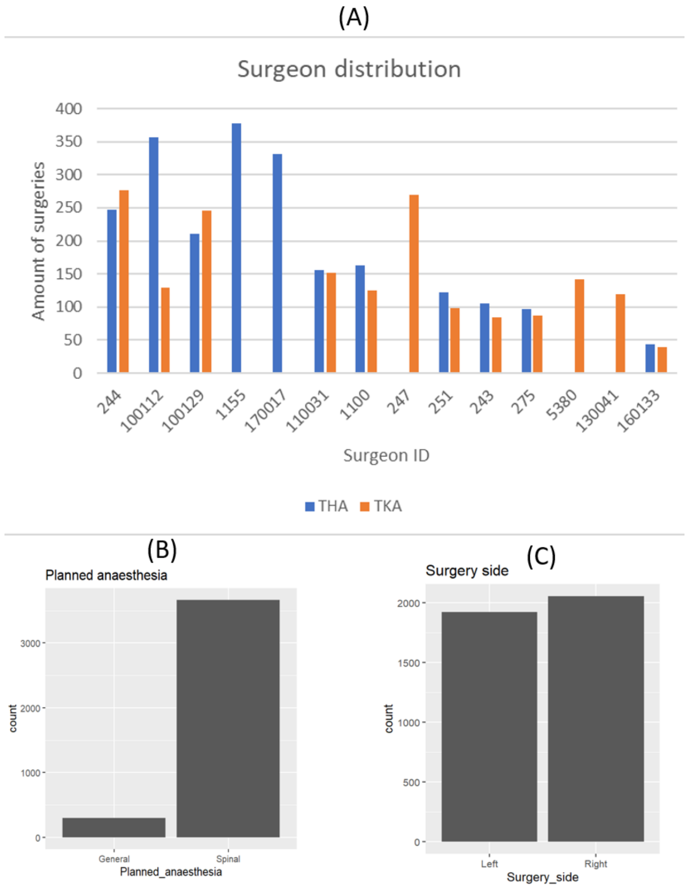


Figure 3.6: Bar chart data visualisations of (A) surgery distribution of each surgeon, (B) planned anaesthesia, and (C) surgery side.

3.4. Data imputation

In step 11, we remove the postoperative variables which we used to calculate the LOS with. We also remove the pseudo_id and surgery date variables, which were used for identifying surgeries. We are left with a dataset of 3974 rows and 25 columns. Two of these columns are the LOS and LongLOS variable, and the remaining 23 variables are features. The current NA% is 1.5%. The next step is to fill in the missing values of the remaining data, which is necessary to use the columns and rows of the corresponding cell. The missing data pattern for our data is shown in Figure 3.7. Every row indicates a certain scenario, which is a combination of red and blue blocks which describe whether each variable has missing values. A red block means that there are missing values for the variables mentioned at the upper axis, and a blue block means that there are no missing values for this variable. The left row counts the number of scenarios present in the dataset. For example, the fourth row means that there are 6 patients who have missing values for the urea and thrombocytes blood test result. The right axis shows the number of variables with missing values for a certain scenario. This value is 2 for the fourth row, because the urea and thrombocytes blood test results are missing. The variables on the right of the upper axis are the variables with the most missing values. Figure 3.7 gives a visual representation of where the missing values are situated in the dataset.

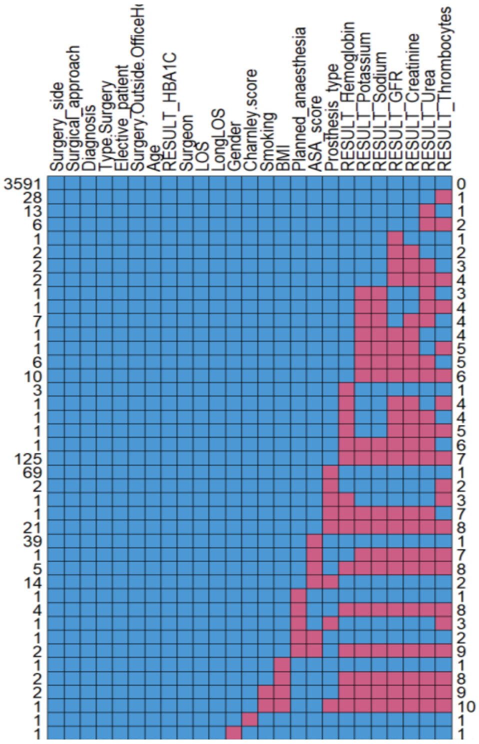


Figure 3.7: Missing data patterns of the data.

Before data imputation, we split the dataset based on whether a patient undergoes a THA or TKA surgery, which is in line with the literature described in Chapter 2. Thus, we perform the data imputation of each of these datasets separately. This means that each of the new subsets do not contain the feature about the type of surgery performed anymore, and thus the total amount of features of each of these subsets becomes 24. The THA dataset contains 2208 surgeries and the TKA dataset contains 1766 surgeries.

We perform data imputation, which is the process of filling in missing data. One data imputation method is MICE, or Multiple Imputation by Chained Equations (Van Buuren & Groothuis-Oudshoorn, 2011). MICE has been widely used before for machine learning in a healthcare-related setting (Polo Friz et al., 2022). We use the 'mice' function in R to apply data imputation. Single imputation is when only one value is filled in for each missing value in the original dataset. Multiple imputation creates copies of the original dataset and fills in different values for each of those datasets. When the amount of missing data is low, such as 5% or less, the performance of multiple imputation is approximately as good as single imputation (Van der Heijden et al., 2006). Since our remaining data has less than 5%, we choose to perform single imputation. We use the imputed dataset for training and testing the machine learning models in

Chapter 4. The imputation is based on the training data, which means that no knowledge is used from the test data during the data imputation process.

Finally, we split each dataset into a train and test subset. We use 80% of the data for training, which is in line with the literature in Chapter 2. This equals 1766 surgeries for the THA dataset and 1412 surgeries for the TKA dataset.

3.5 Summary

In Chapter 3, we elaborate on what input is required for the ML methods for LOS prediction to work. We require data where the features are properly structured, and the data contains no missing values. We collect this data by filing a data request for primary TKA/THA patients. We request variables based on academic literature. We perform various steps which combine columns into single features, remove patients where no LOS calculation is possible, and resolve mistakes in the data. We also remove features with a very high percentage of missing values. For features with a relatively low missing value percentage, we perform data imputation to fill in these missing values. We split the data based on whether a patient undergoes a TKA or THA, and we split each subset again based on an 80:20 train test split.

We started Chapter 3 with one dataset of 260 columns/variables and 5391 rows/surgeries, and we end up with one THA dataset of 2208 surgeries and one TKA dataset of 1766 surgeries. Both datasets contain 24 variables, which is a mix of numerical and categorical variables. Table 3.4 shows an overview of all remaining variables. The next step is to use the prepared data as input for creating the ML models, which we execute in Chapter 4.

Table 3.4: Overview of all remaining variables after the data preparation.

Variable	**Class**	**Variable**	**Class**
Gender	*character*	Age	*numeric*
Smoking	*character*	UITSLAG_GFR	*character*
BMI	*numeric*	UITSLAG_Hemoglobine	*numeric*
Surgery_side	*character*	UITSLAG_Kalium	*numeric*
Surgical_approach	*character*	UITSLAG_Kreatinine	*numeric*
Prosthesis_type	*character*	UITSLAG_Natrium	*numeric*
Diagnosis	*character*	UITSLAG_Trombocyten	*numeric*
Charnley.score	*character*	UITSLAG_Ureum	*numeric*
Elective_patient	*character*	UITSLAG_HBA1C	*character*
Surgery.Outside.OfficeHours	*character*	Surgeon	*character*
Planned_anaesthesia	*character*	LOS	*numeric*
ASA_score	*character*	LongLOS	*character*

Chapter 4: Machine learning methods

In Chapter 3, we collect relevant features which should be suitable for predicting the LOS for primary TKA/THA patients. We now have two imputed datasets of 2208 surgeries for the THA dataset and 1766 surgeries for the TKA dataset. Both datasets contain 24 variables. Two of these variables are the LOS and LongLOS variables. The other 22 variables are features. In Chapter 4, we answer the research question: *How do we configure the LOS prediction methods to improve their performance?* In Section 4.1 we discuss feature selection methods which help reduce the number of input features required for the ML models. In Section 4.2 we describe the details concerning the execution of the machine learning (ML) models, such as parameter tuning and data preprocessing.

As mentioned in Section 3.4, we divide the datasets in train and test data. The THA train dataset contains 1766 surgeries and the TKA train dataset contains 1412 surgeries. From this point on, we only use the train datasets to perform feature selection methods and create ML models. We use the test datasets in Chapter 5 to test the performance of the models.

4.1 Feature selection methods

Using too many features in a model or code is computationally demanding, and it could also lead to overfitting in certain models and result in worse predictions (Jović et al., 2015; Witten et al., 2016). Also, having a high number of features could make it cumbersome for OCON to collect data. Thus, we use feature selection methods for these purposes, which we explain later in this section. As mentioned in Chapter 2, we name four categories of feature selection methods, namely filter methods, wrapper methods, embedded methods, and hybrid methods. For this research, we will be applying the first three feature selection methods. We also use a fourth subset of features where no feature selection method is applied, using all 22 remaining features which are left after Chapter 3. We do this to compare and test the practicality of using feature selection methods.

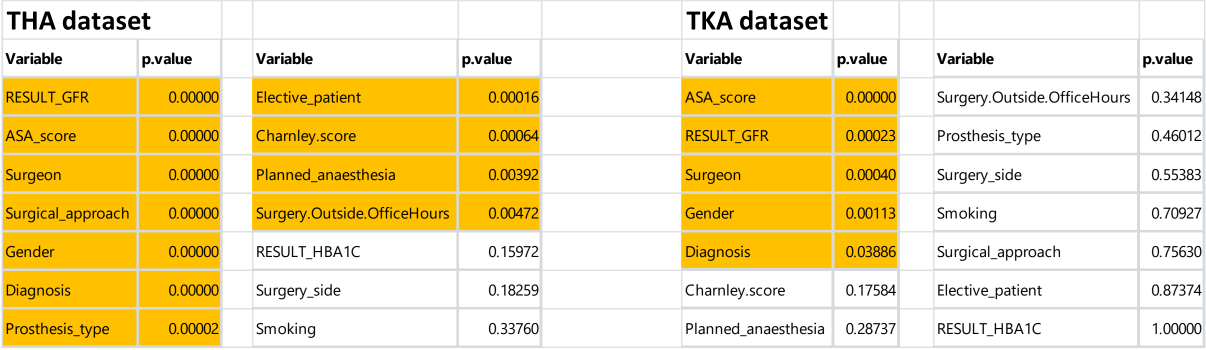
OCON states that predicting whether a patient experiences a long LOS is sufficient, and predicting the exact LOS is not necessary. The literature in Chapter 2 also uses the categorical variable LongLOS as a dependent variable, instead of the LOS variable. Thus, we consider the LongLOS as the target variable.

4.1.1 Filter method

The first feature selection method we use is a filter method, it is a combination of two methods. We select categorical features based on Pearson's chi-squared tests with the LongLOS variable (Witten et al., 2016), and we select numerical features by using a one-way ANOVA F-test (Elssied et al., 2014) between the features and the LongLOS. For both tests, we use a significance level of 0.05 (Thaseen & Kumar, 2017). Furthermore, we remove highly correlated numerical features to get rid of multicollinearity (Mallampati et al., 2023; Liu et al., 2020). We select these features based on their Pearson correlation with each other, and we consider an absolute value of 0.6 or higher as a significant correlation (Liu et al., 2020). For this calculation we use the LOS variable instead of the LongLOS variable because numerical variables are required for calculating the Pearson correlation.

First, we perform chi-squared tests between the LongLOS and the categorical features and calculate the p-values. The rounded p-values are shown in Table 4.1 for each training dataset. The orange features have a p-value lower than the significance level of 0.05, which means that we consider these features as significant and select them. We discard the remaining features and are left with 11 categorical features for the THA dataset and five categorical features for the TKA dataset.

Table 4.1: Rounded p-values from chi-squared tests between categorical features and LongLOS for both training datasets.



THA dataset				TKA dataset			
Variable	p.value	Variable	p.value	Variable	p.value	Variable	p.value
RESULT_GFR	0.00000	Elective_patient	0.00016	ASA_score	0.00000	Surgery.Outside.OfficeHours	0.34148
ASA_score	0.00000	Charnley.score	0.00064	RESULT_GFR	0.00023	Prosthesis_type	0.46012
Surgeon	0.00000	Planned_anaesthesia	0.00392	Surgeon	0.00040	Surgery_side	0.55383
Surgical_approach	0.00000	Surgery.Outside.OfficeHours	0.00472	Gender	0.00113	Smoking	0.70927
Gender	0.00000	RESULT_HBA1C	0.15972	Diagnosis	0.03886	Surgical_approach	0.75630
Diagnosis	0.00000	Surgery_side	0.18259	Charnley.score	0.17584	Elective_patient	0.87374
Prosthesis_type	0.00002	Smoking	0.33760	Planned_anaesthesia	0.28737	RESULT_HBA1C	1.00000

To illustrate the significance of chosen features compared to ignored features, we visualise the importance of the chosen features as follows. In the following examples, we use the THA training dataset. Table 4.1 shows that the ASA score is a feature with a significantly low p-value, which means that it should be a relevant feature. Figure 4.1A shows the bar chart of the ASA score for a short or long LOS. We barely see patients with an ASA score of 1 experiencing a long LOS, even though there are plenty of such patients who experience a short LOS. This highlights the importance of the ASA score as a feature. In contrast, the Smoking feature has a high p-value, which indicates that it is not a useful feature. Figure 4.1B shows the bar chart of the Smoking feature for a short or long LOS. There is barely any difference in proportion of patients who smoke. For patients who experienced a long LOS, 7.5% of the patients smoked. For patients who experienced a short LOS, 9% of the patients smoked. This highlights further why the Smoking feature is not a promising feature to use in the ML models.

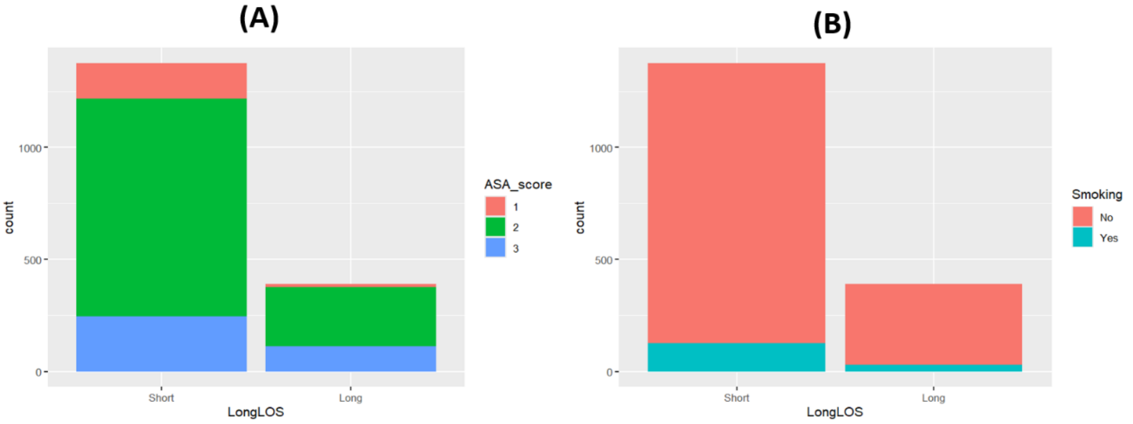


Figure 4.1: Bar charts of the THA training dataset for comparing the LongLOS with (A) the ASA score and (B) the Smoking feature.

The performance of a model could improve when highly correlated features are removed (Mallampati et al., 2023). Other feature selection methods can also benefit from removing correlated features. We discard numerical features by analysing their Pearson correlation coefficient with other numerical features. Academic literature tends to use a correlation threshold of at least 0.6 (Liu et al., 2020), which is what we also use. This means we view a correlation coefficient of more than 0.6 or less than -0.6 as a significant correlation, and thus we discard one of the correlated features in such case.

Figure 4.2 shows the correlation coefficients between the numerical features of each training dataset. None of the features are significantly correlated with each other. Thus, we do not remove any numerical features for either dataset in this stage.

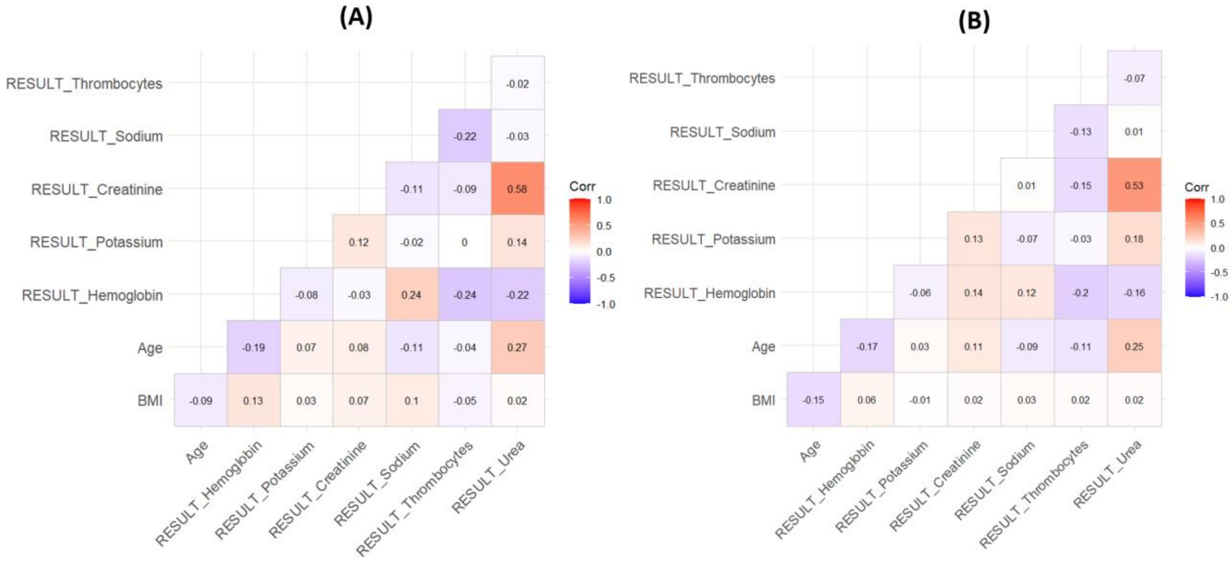


Figure 4.2: Pearson correlation plot between numerical features for the train dataset of (A) THA and (B) TKA.

Next, we select relevant numerical features with the help of a one-way ANOVA F-test (Elssied et al., 2014). We perform the test between the numerical features and the categorical variable LongLOS. After running the ANOVA F-test, we analyse the p-value based on the F statistic. We select the features with a p-value lower than our chosen significance level of 0.05. The p-value of every remaining numerical feature is shown in Table 4.2 for both training datasets. We select features with at least a p-value smaller than 0.05. The corresponding features are coloured orange. For the THA dataset, we are left with seven numerical features, and for the TKA dataset, we are left with three numerical features.

Table 4.2: Rounded p-values from ANOVA test between numerical features and LongLOS for each training dataset.

THA dataset			**TKA dataset**	
Features	**p_value**		**Features**	**p_value**
Age	0.00000		Age	0.00000
RESULT_Hemoglobin	0.00000		RESULT_Urea	0.00007
RESULT_Urea	0.00000		RESULT_Hemoglobin	0.00010
RESULT_Sodium	0.00002		BMI	0.45579
RESULT_Thrombocytes	0.00470		RESULT_Sodium	0.63961
RESULT_Creatinine	0.01145		RESULT_Thrombocytes	0.65376
RESULT_Potassium	0.02654		RESULT_Creatinine	0.69706
BMI	0.43595		RESULT_Potassium	0.93674

Finally, we combine the selected features from each training dataset. To summarise, after applying the filter feature selection method, our THA training data consists of 1766 surgeries and 18 features columns, and our TKA training data consists of 1412 surgeries and eight features. An overview is shown in Table 4.3.

Table 4.3: Selected features of each dataset after applying the filter feature selection method.

THA dataset		TKA dataset
Gender	Age	Gender
Surgical_approach	RESULT_GFR	Diagnosis
Prosthesis_type	RESULT_Hemoglobin	ASA_score
Diagnosis	RESULT_Potassium	Age
Charnley.score	RESULT_Creatinine	RESULT_GFR
Elective_patient	RESULT_Sodium	RESULT_Hemoglobin
Surgery.Outside.OfficeHours	RESULT_Thrombocytes	RESULT_Urea
Planned_anaesthesia	RESULT_Urea	Surgeon
ASA_score	Surgeon	

4.1.2 Wrapper method

Filter methods are totally separate from the machine learning model which is used for modelling. In contrast, wrapper methods incorporate the machine learning model into the feature selection process. They iteratively use a part of the features, create a machine learning model, test the performance of such a model, and evaluate which subset of features yields the best predictive performance. Forward feature selection may be more computationally efficient, but it tends to find weaker subsets than backward feature selection (Kumar, 2014). Since we prioritise performance over running time, we perform backward sequential feature selection. This method starts with all features, and iteratively reduces the number of features based on a certain criterium. This criterium can be, for example, the AUC of the created model with the selected subset of features after validating the model. Since a wrapper method is based on the ML algorithm used, the selected features also differ per algorithm. One of the inputs for backward feature selection is the number of features to evaluate in each iteration. From this point on, the categorical features are transformed into multiple dummy features. We explain the concept of dummy features in Section 4.2. For backward feature selection, we use feature subset sizes of 5 till 45 while using steps of 5. This means that our wrapper method evaluates batches of five features in each iteration. Using smaller steps could severely increase the running time and risks overfitting (Kohavi & Sommerfield, 1995).

We test each subset of features by using k-fold cross-validation on the resulting model. The k-fold cross-validation method divides the data into k groups of approximately equal size (Hastie et al., 2021). Consequently, the chosen machine learning model is trained on all data except one of the k groups. The k groups are used as validation datasets to test how well the trained machine learning model performs. This procedure is executed k times, where the validation dataset differs every time. Figure 4.3 shows a schematic of k-fold cross-validation. The numbers in Figure 4.3 are the indices for each observation in a dataset. The index of the last observation is n. Each coloured bar represents an iteration in k-fold cross-validation, where the blue bar is the training dataset and the orange bar the validation dataset. Usually, academic literature chooses values of 5 or 10 for k in k-fold cross-validation (Hastie et al., 2021). We apply 10-fold cross-validation on the training data to measure the performance of our subset of features after every iteration. We use AUC value of the ROC curve as a performance metric.

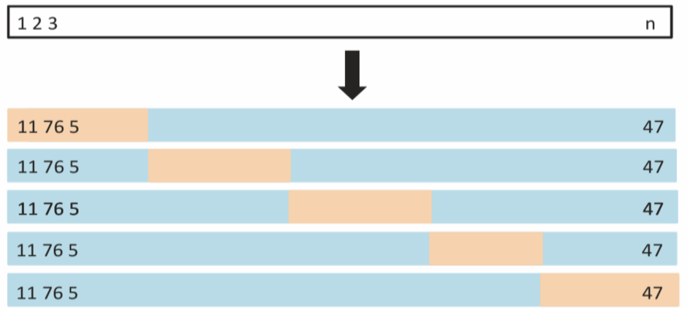


Figure 4.3: K-fold cross-validation, where the numbers are indices of observations of a dataset (Hastie et al., 2021, p. 203).

4.1.3 Embedded method

The final feature selection method is the embedded method. We use two approaches. One approach is Lasso for the logistic regression. Lasso is a method which minimises a formula based on the prediction error and the coefficients used (Hastie et al., 2021). This makes sure that the prediction error as well as the number of features used is minimised. Lasso is not applicable on the other ML models we use in this research, and that is why we use another approach. This second approach consists of making use of the varImp function of the caret library in R. This function calculates a numerical feature importance of a ML model, and the selected feature importance methods depend on the ML model used (*VarImp Function - RDocumentation*, n.d.). Certain ML methods are not appropriate for the varImp function, namely KNN, NBC, LSVM and SVMR. Instead, the feature importance of these four ML methods will be calculated with the filterVarImp function (*FilterVarImp Function - RDocumentation*, n.d.). Since filterVarImp function is not ML model specific, all these four ML methods have the same feature importance. Technically, this means that selecting features with the help of filterVarImp is not an embedded feature selection. However, we use this method as a substitute for the ML methods that do not have a unique feature importance ranking process in the varImp function. We refer to features selected based on feature importance as embedded features.

To illustrate how the varImp function works, we cover two examples in Figure 4.4, namely the LOGR and RF methods which are used on the THA train data. We run both functions with the dataset that contains all remaining features and plot the feature importance graphs in Figure 4.4.

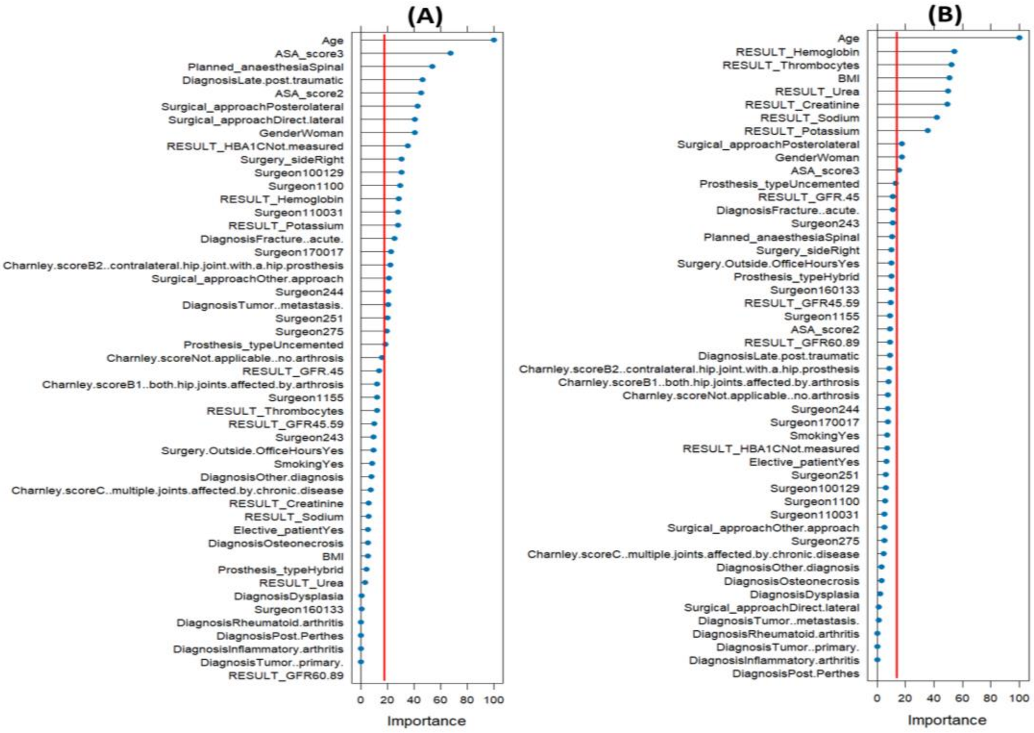


Figure 4.4: Importance plots of (A) logistic regression (LOGR), and (B) random forest (RF).

We determine which features are important based on visual analysis. This is different per ML method, and thus the cutoff value to determine which features to include in the final model are also different among ML methods. We choose a cutoff of 18 for Figure 4.4A because one could argue that the features above this cutoff are clustered together, as this is demonstrated with a bigger gap in importance. We choose a cutoff of 14 for Figure 4.4B while applying the same logic. With this feature selection method, we select the upper 24 features until Prosthesis_typeUncemented for the LOGR method, and we select the upper 11 features until ASA_score3 for the RF method.

4.2 Machine learning methods

In this section, we explain the details regarding the execution of the machine learning methods selected in Chapter 2. The ML models are:

1. Logistic regression (LOGR)
2. Naive Bayes classifier (NBC)
3. K-nearest neighbour (KNN)
4. Linear support vector machine (LSVM)
5. Support vector machine with radial basis function kernel (SVMR)
6. Random forest (RF)
7. Extreme gradient boosting (XGB)
8. Artificial neural network (ANN)

We first use one of the feature selection methods described in Section 4.1 on the train datasets, and then we use the remaining data to train the ML models. When training the models, we use 10-fold repeated cross-validation to ensure a proper level of flexibility in the models, which can reduce overfitting (Hastie et al., 2021). We repeat the cross-validation three times to take randomness into account. We only apply cross-validation once for the wrapper method instead of three times to reduce severely long running times. After training the ML models, we test the models on the test datasets and calculate the AUC for the ROC curve of each model. We cover these results in Chapter 5.

Various machine learning methods do not handle categorical features well and require adjusting, such as support vector machines and KNN (Pagan et al., 2023; Edwards & Raskutti, 2004). LOGR also requires adjusting of the categorical features before being able to use them, as categorical features transform into dummy features when creating the model. Dummy features are binary features which signify whether one of the many possible values for a categorical feature is present. For example, the categorical feature ASA score is split up into dummy features ASA_score2 and ASA_score3. Only one of these dummy features can have a value of 1, which indicates the corresponding ASA score. When both dummy features equal 0, the ASA score of the corresponding patient is 1. To make sure the input data is applicable on all ML models; we transform the categorical features into dummy features. The process of transforming categorical features into dummy features is called one-hot encoding (Hastie et al., 2021). In total, we end up with 50 features for the THA train dataset and with 46 features for the TKA train dataset. The difference in features between these datasets is because certain categorical variables, such as the variable about the surgical approach, have different values for each dataset.

Certain features cover a larger range of numerical values than others. Machine learning methods tend to view features with a large range of numerical values as more important in modelling compared to features which cover a smaller range (Ozsahin et al., 2022). To make sure all features are in the same scale, we preprocess the data with standardisation. This method subtracts the mean of the data from the original numerical data and divides it by its standard deviation. According to literature, feature scaling methods like standardisation either improve ML models or barely affect them. We apply standardisation on all ML methods.

Various ML methods have parameters, such as the analysed number of neighbours in the KNN method or the number of features to consider at each node in the RF method. The tuning of these parameters is done by the train function from the caret library in R. The function creates the model and finds the most suitable parameters by testing various parameters and uses the AUC metric to find the best performing model.

Initial testing of the LSVM machine learning model suggests that the model do not seem to be performing particularly well. Thus, we add another but similar model, which is the support vector machine with radial basis function kernel (SVMR) (Park et al., 2023).

The ANN methods applied in Ramkumar et al. (2019a) and Ramkumar et al. (2019c) use multiple hidden layers and multiple neurons per layer. We work with the nnet library in R, which works with one hidden layer. We create an ANN with all remaining features of the THA train dataset, and the caret train function tunes the size and weight decay parameters by using cross-validation. The size parameter signifies the number of neurons in the hidden layer, and the weight decay parameter reduces overfitting by lowering the weights in the ANN. Initially, the function tunes the model with a size parameter of 1, 3, and 5, while the decay parameters of 0, 0.001, and 0.1 are used. When inputting the THA dataset for all remaining features, the nnet function chose a size of 1 and a decay parameter of 0.1 as the optimal parameters. The validation AUC is 0.72 and the test AUC is 0.75. Usually, neural networks are complex and contain more than one neuron (Ramkumar et al., 2019a; Ramkumar et al., 2019c). To make sure that we do not miss out on important parameters, we test more tuning values to search for a more complex and possibly better performing ANN model. For the size parameter, we try values from 1 to 18. For the decay parameter, we input values from 0 to 0.01 while making steps of 0.0002. Figure 4.5 shows the plot of the tuning process. The colours are the number of neurons in the hidden layer, and the x-axis shows the weight decay value. The y-axis shows the AUC value, which is the key performance indicator.

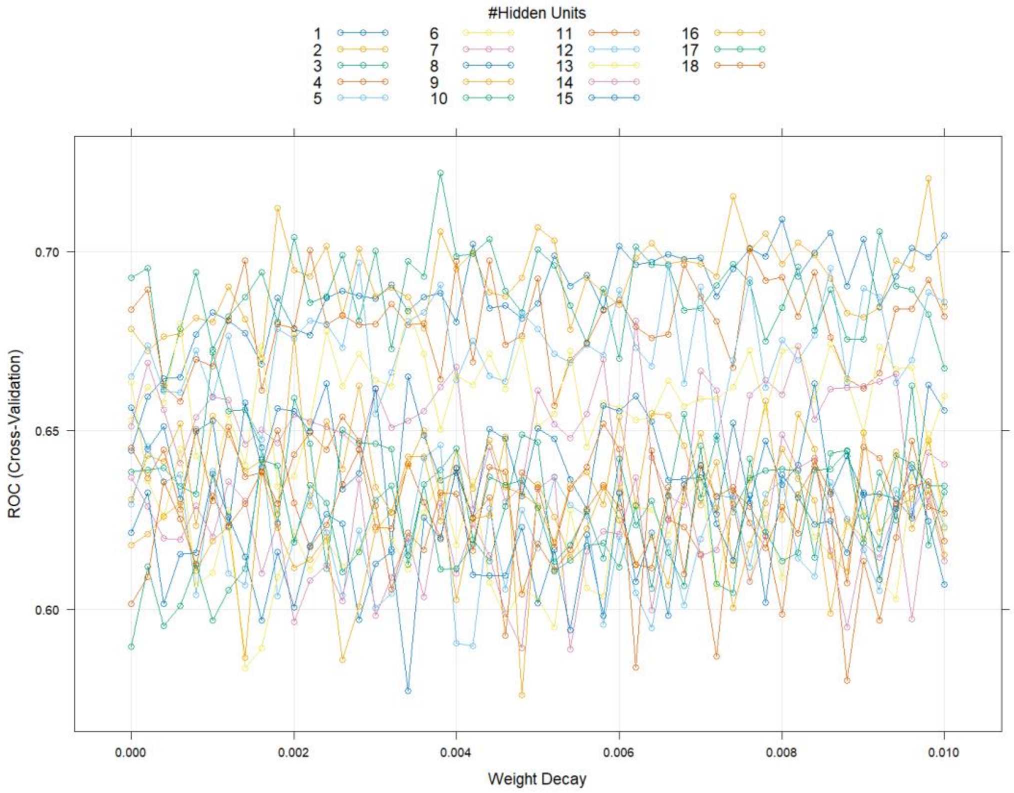
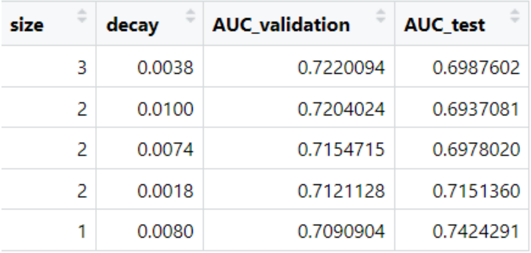


Figure 4.5: Plot of tuning process of ANN with all remaining features as input while using the THA train dataset.

The top five configurations in Figure 4.5 with the highest validation AUC values are shown in Table 4.4. We tested these configurations on the THA test data, which is shown in the fourth column (AUC_test) in Table 4.4. Even though the first four configurations got the highest validation AUC values, they all score lower than the configuration which only uses one neuron in the hidden layer and with a weight decay parameter of 0.008. This configuration has a test AUC of 0.74. This AUC value is still lower than the test AUC of the ANN which used a size parameter of 1 and a decay parameter of 0.1, which is 0.75.

Table 4.4: ANN configurations with highest AUC validation values for THA training data with all remaining features.



size	decay	AUC_validation	AUC_test
3	0.0038	0.7220094	0.6987602
2	0.0100	0.7204024	0.6937081
2	0.0074	0.7154715	0.6978020
2	0.0018	0.7121128	0.7151360
1	0.0080	0.7090904	0.7424291

Figure 4.6 shows the visualisation of the ANN that we use. On the left are the input neurons, in the middle is the hidden neuron, on the right is the output neuron, and at the top are the biases. Since this configuration is the ANN model with the highest test AUC value, it gives a strong indication that the default tuning parameters of the train function for the ANN are sufficient. The fact that certain configurations have a higher validation AUC value than the configuration with a size parameter of 1 and decay parameter of 0.1 shows that these configurations are vulnerable to overfitting for the used dataset. So, we use one hidden layer, one neuron, and a decay parameter of 0.1 for our THA dataset. For the TKA dataset, we find that this configuration is also the most suitable one, and we conclude that the default tuning parameters are also sufficient for this dataset.

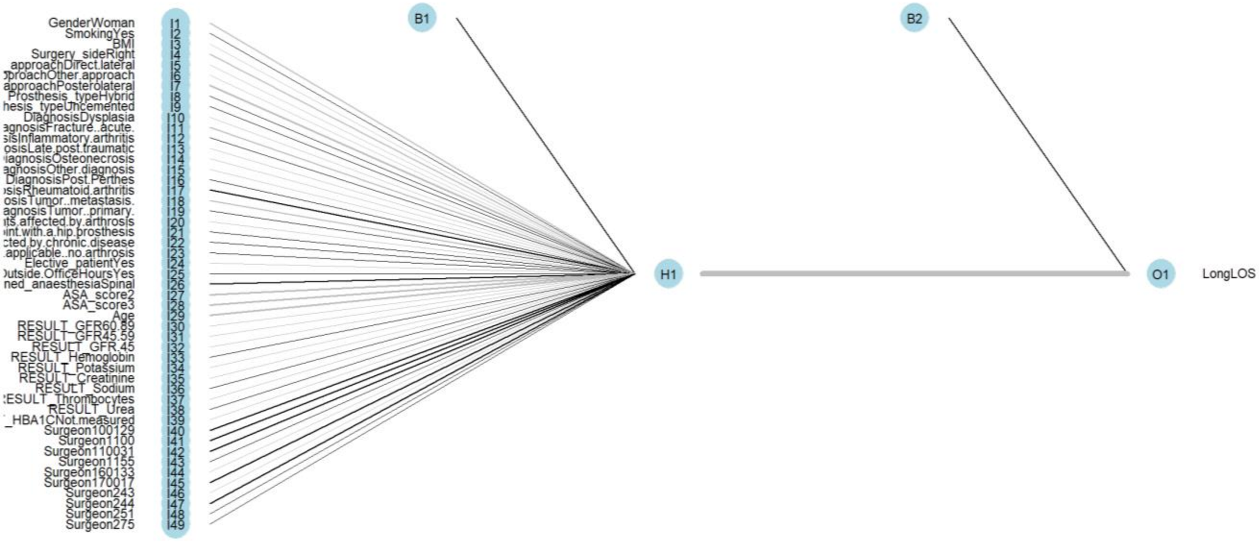


Figure 4.6: Visualisation of the ANN with all remaining features of the THA train data.

4.3 Summary

To recap, we perform three types of features selection methods, create subsets of features, and apply eight ML methods on the remaining data. The filter method uses chi-squared tests, one-way ANOVA F-tests, and Pearson correlation coefficients to select the most promising features. The wrapper method uses the backward selection method to select its features while the corresponding ML method is also executed. The embedded method uses a feature importance metric to select the most promising features based on varying cutoff values for the importance values. Additionally, as an extra embedded method, we apply Lasso for logistic regression. We also apply the ML methods for THA/TKA dataset where we do not perform any feature selection method (FSM). With one-hot encoding, we transform the categorical features in dummy features. Furthermore, we preprocess all features by scaling them to make sure the features are treated equally in the modelling. We train the ML models with a train function from the caret library. We program it to use cross-validation and automatically tune the parameters so that it chooses the most suitable parameters for the corresponding models. For each of the two training datasets (THA/TKA), we have 26 models based on combinations of FSMs and ML methods and eight more ML models for using all remaining features with no FSM. In total, we have 34 ML models for each of the THA and TKA datasets. In Chapter 5, we show the results of each 68 ML methods after being tested on the test datasets.

Chapter 5: Results

In this chapter, we cover the performance of each FSM and ML method described in Chapter 4. We answer the following research questions: *Which LOS prediction methods perform the best?* and *How can we implement the LOS prediction methods in practice?* In Section 5.1, we test the resulting models on the test data, create the ROC curves, and calculate the AUC values. The test datasets are separate for the THA and TKA data. In Section 5.2, we visualise the learning curves of ML methods to see how well the models learn with the training data (Meek et al., 2002). In Section 5.3, we analyse the performance of ML models and select the most promising ones. In Section 5.4, we create a prediction tool based on the most promising ML models and their corresponding features. The tool is able to analyse an incoming new patient and predict whether this patient will experience a long LOS or not. In Section 5.5, for each of these ML models, we compare their predictions for the test data with the actual results.

5.1 Modelling results

Figure 5.1 shows the ROC curves and the AUC values for each FSM and ML methods applied for the THA dataset. Figure 5.2 shows the same information but for the TKA dataset. Each used ML method and their AUC values have different colours. For example, in Figure 5.1A we compare all remaining features without any FSM. The RF model with an AUC of 0.755 seems to be the best model to fit, because it has the highest AUC. The number of features mentioned in the plots of the ROC curves of the Lasso features includes the dummy features as described in Section 4.2. We elaborate further on the best performing models in Section 5.3.

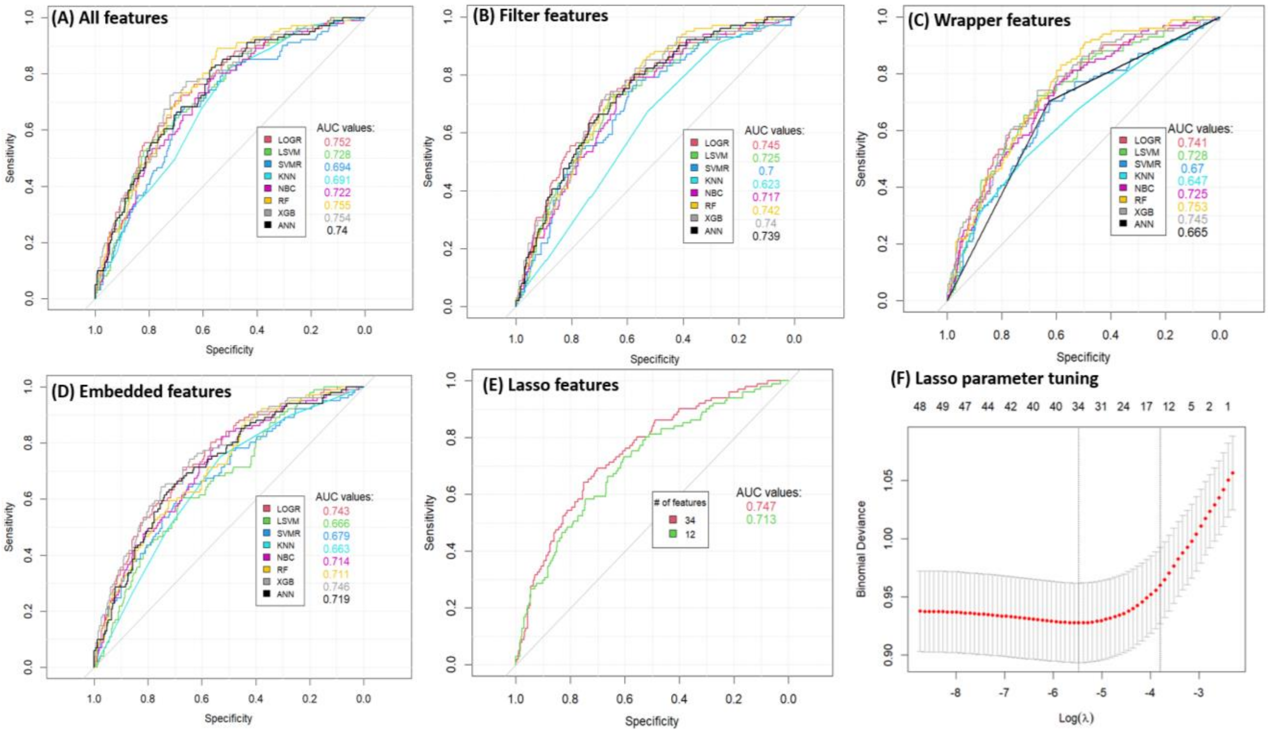


Figure 5.1: ROC curves and AUC values for each FSM and ML method for the THA dataset. The plots are for: (A) all remaining features, (B) the features after the filter FSM is applied, (C) the features after the wrapper FSM is applied, (D) the features after the embedded FSM using the feature importance is applied, (E) the features selected with the Lasso method, and (F) a plot for the Lasso parameter tuning which is used for feature selection.

Figure 5.1F and Figure 5.2F also include a plot which shows the tuning process for the lambda parameter for the Lasso method. This lambda parameter makes sure that using more features is penalised. This means that a higher lambda value ensures fewer features, but this also means that the overall performance of the corresponding model could deteriorate. We select two subsets of features with the Lasso method. The first selection is the feature subset which has the lowest binomial deviance when we performed cross-validation while tuning the lambda parameter and thus performs the best. The second selection of features has a binomial deviance value which is within one standard deviation of the best performing feature subset. This means that the second selection of features has a somewhat similar model performance to the best performing feature subset, while containing less features.

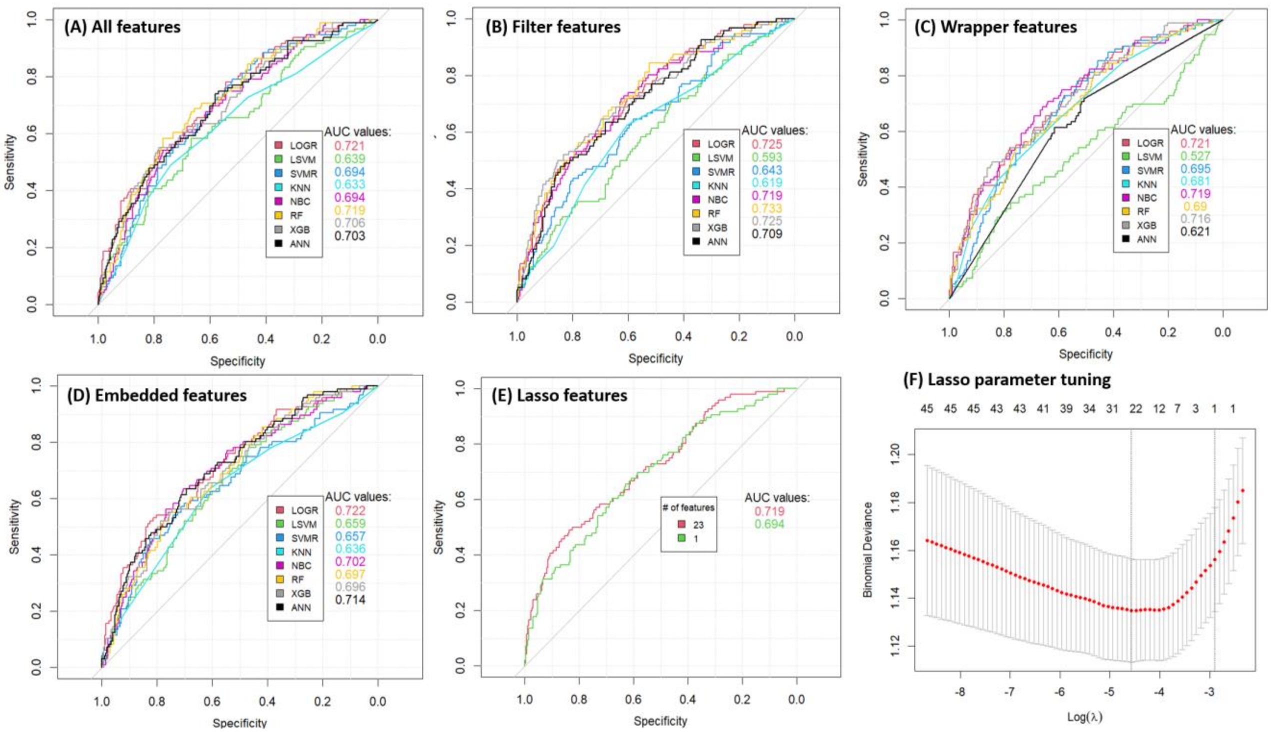


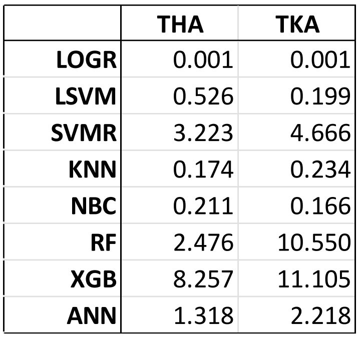
Figure 5.2: ROC curves and AUC values for each FSM and ML method for the TKA dataset. The plots are for: (A) all remaining features, (B) the features after the filter FSM is applied, (C) the features after the wrapper FSM is applied, (D) the features after the embedded FSM using the feature importance is applied, (E) the features selected with the Lasso method, and (F) a plot for the Lasso parameter tuning which is used for feature selection.

We run the code for the FSMs and ML methods on a laptop with the specifications as shown in Table 5.1. We calculate the running time of the ML models. We can neglect the running times of all remaining features with no FSM, filter FSM, and embedded FSM since each of those (eight ML models per each) are finished within 30 minutes. Table 5.2 shows the running time in hours of each ML method during the execution of the wrapper FSM. The wrapper FSM has by far the longest running time of all FSMs, which is in line with literature (Jović et al., 2015). Even though the THA training dataset has more rows (1766 rows) than the TKA training dataset (1412 rows), the running time is longer for all ML methods except LSVM and NBC for the TKA data. The running time of the wrapper method for both the THA and TKA datasets is approximately 2 days. Running the whole R script for both datasets, which consists of data extraction, data preparation, data imputation, FSM execution, ML model training and testing, and plotting learning curves, takes approximately four to 5 days.

Table 5.1: Relevant specifications of the device where we run the R code on.

Central processing unit	12th Gen Intel Core i7 Processor (14-core) i7-12800H, 1.8 GHz with Turbo Boost up to 4.8 GHz, with 24 MB of Cache
Graphics processing unit	NVIDIA® GeForce RTX™ 3060 (6GB GDDR6 VRAM)
Random Access Memory	16 GB DDR5-4800 MHz
Operating system	Windows 11 Home, Version 23H2

Table 5.2: Running time in hours of each ML method which ran for the wrapper FSM.



	THA	TKA
LOGR	0.001	0.001
LSVM	0.526	0.199
SVMR	3.223	4.666
KNN	0.174	0.234
NBC	0.211	0.166
RF	2.476	10.550
XGB	8.257	11.105
ANN	1.318	2.218

5.2 Learning curves

We plot learning curves to analyse how well the models are trained on the training data (Meek et al., 2002). Learning curves can give an indication whether too much or too little training data is used, or how well the model is performing on the test data compared to the training data. We measure a model's performance with their AUC value. Figure 5.3 shows the learning curves using all remaining features for the THA dataset, and Figure 5.4 shows the learning curves using all remaining features for the TKA dataset. The x-axis is the number of training rows used and the y-axis is the AUC value of the model which is trained on this number of rows. The blue lines are the AUC values when the model is tested on the training data, and the orange lines are the AUC values when it is tested on the test data. If AUC values of both lines are increasing when more training data is used, then it means that having more training data is beneficial. The XGB model in Figure 5.3 is a clear example of this. When an orange line starts to increase, but eventually starts to decrease, it means that the model is overfitting, and that less data would be better for the corresponding model's performance. An example of this is the LSVM for the TKA data, which is shown in Figure 5.4. When there is a large gap between the orange and blue lines, it means that the model has a significantly different performance on the training data than the test data. Adding more data could lower such gaps. Both RF models in Figure 5.3 and Figure 5.4 show significant gaps between the training and testing AUC values. We perform any further analysis of relevant learning curves in Section 5.3.

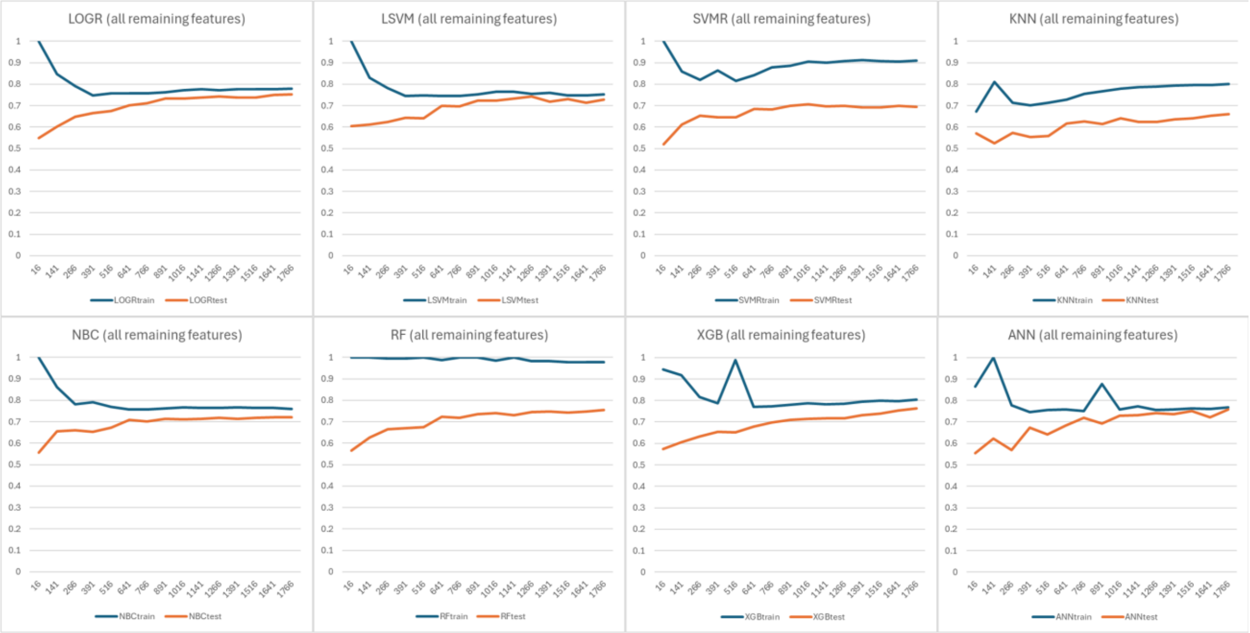


Figure 5.3: Learning curves for the ML models using all remaining features for the THA dataset.

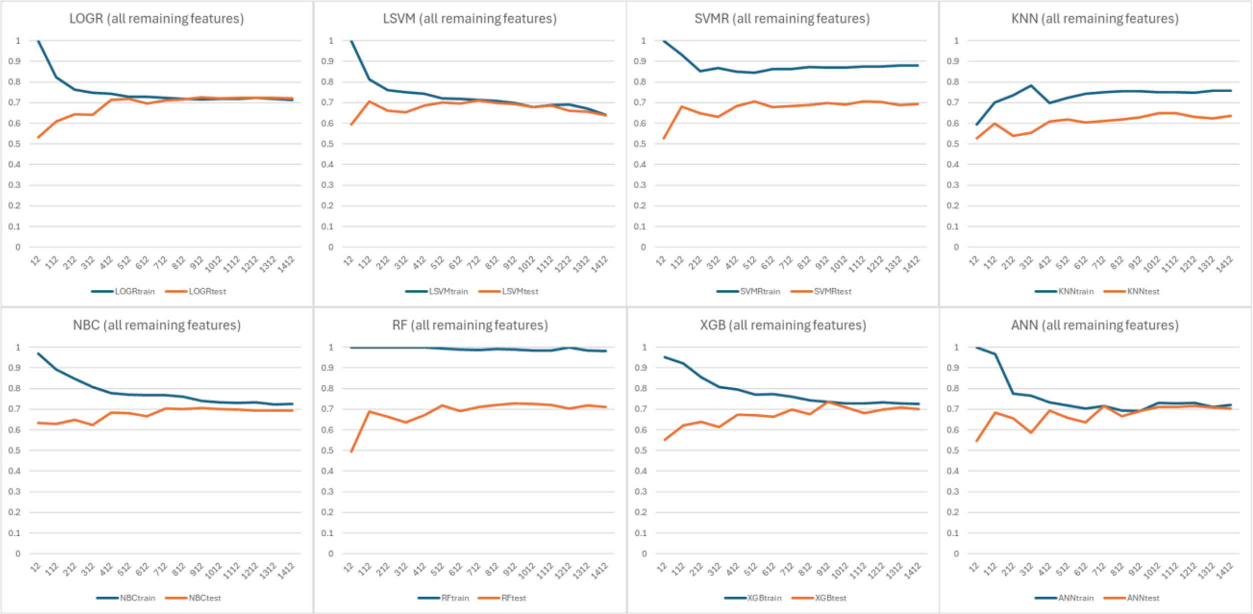


Figure 5.4: Learning curves for the ML models using all remaining features for the TKA dataset.

5.3 Analysis of results

In this section, we analyse which models are the most promising and which ones we will use in our prediction tool which is mentioned in Section 5.4. We first analyse AUC values (Section 5.3.1) and number of features used for each ML model. Afterwards, we analyse the learning curves (Section 5.3.2) and feature importance (Section 5.3.3) for the most promising models.

5.3.1 Model performance

The main KPI we use for defining the best models is the AUC value, but we also analyse the number of features used. Table 5.3 shows the overview of AUC values for each FSM and ML method for the THA dataset, while Table 5.4 shows the same information about the TKA dataset. The rows are the ML methods, and the columns are the FSMs. Both tables also show the number of features used for the corresponding models. We do not count dummy features here. In Table 5.3 and Table 5.4, the bold and underlined numbers are the top 10 ML models with the highest AUC values either for the THA or TKA datasets.

The Lasso FSM is an embedded FSM. Since we also used another embedded FSM based on feature importance, we added the Lasso results at the bottom of the column about the embedded FSM features. LassoOpt is the feature subset with the lowest binomial deviance when we performed Lasso, while LassoStd is the feature subset with a binomial deviance which is within one standard deviation of the lowest binomial deviance.

Table 5.3: AUC values and number of features overview of the models trained and tested on the THA data.

AUC	All	Filter	Wrapper	Embedded	# of features	All	Filter	Wrapper	Embedded
LOGR	***0.7517***	***0.7454***	0.7407	***0.7431***	LOGR	***22***	***18***	13	***13***
LSVM	0.7283	0.7250	0.7284	0.6659	LSVM	22	18	22	20
SVMR	0.6941	0.7001	0.6705	0.6791	SVMR	22	18	22	20
KNN	0.6905	0.6229	0.6468	0.6626	KNN	22	18	5	20
NBC	0.7220	0.7171	0.7251	0.7141	NBC	22	18	20	20
RF	***0.7551***	***0.7422***	***0.7531***	0.7112	RF	***22***	***18***	***22***	11
XGB	***0.7538***	0.7405	***0.7454***	***0.7456***	XGB	***22***	18	***19***	***12***
ANN	0.7398	0.7390	0.6653	0.7190	ANN	22	18	15	8
LassoOpt				***0.7471***	LassoOpt				***17***
LassoStd				0.7126	LassoStd				10

Table 5.4: AUC values and number of features overview of the models trained and tested on the TKA data.

AUC	All	Filter	Wrapper	Embedded	# of features	All	Filter	Wrapper	Embedded
LOGR	***0.7211***	***0.7251***	***0.7206***	***0.7440***	LOGR	***22***	***8***	***18***	***13***
LSVM	0.6388	0.5929	0.5271	0.6742	LSVM	22	8	8	20
SVMR	0.6936	0.6435	0.6946	0.6708	SVMR	22	8	22	20
KNN	0.6331	0.6194	0.6806	0.6520	KNN	22	8	8	20
NBC	0.6943	***0.7194***	0.7189	0.6859	NBC	22	***8***	5	20
RF	***0.7190***	***0.7327***	0.6901	0.7101	RF	***22***	***8***	21	12
XGB	0.7057	***0.7254***	0.7158	0.7061	XGB	22	***8***	14	12
ANN	0.7033	0.7092	0.6211	***0.7234***	ANN	22	8	8	***15***
LassoOpt				***0.7192***	LassoOpt				***23***
LassoStd				0.6938	LassoStd				1

With Table 5.3 and Table 5.4, we select a few ML models for each dataset to implement in the prediction tool in Section 5.4. The ML models with the highest AUC values do not necessarily have to be the best ML models, as the number of features is also important. Having fewer features is more desirable because this requires less work from the user of the prediction tool, and it also reduces the probability of human error. Preferably, we do not want to use ML methods which use all remaining features, as these models use the highest number of features of all FSMs. The selected models which are used in prediction tool are coloured green in Table 5.3 and Table 5.4.

For the THA data in Table 5.3, the RF model with all remaining features has the highest AUC value of 0.7551. The RF model with the wrapper features has only a slightly lower AUC value, but the number of features is the same. Thus, we add the RF model which uses all remaining features to our prediction tool. The model with the lowest number of features but still with a good AUC value within the top 10 highest AUC values is the XGB with embedded features.

This model only required 12 features but still managed to achieve a high AUC value of 0.7456. We also add this model to our prediction tool. Similarly, the LOGR model which uses embedded features only uses 13 features and still achieves a high AUC value. This is the third model that we add to our prediction tool. One model which has a decent AUC value but uses only eight features is the ANN ML model which uses embedded features. Its AUC value is not in the top 10 highest AUC values, but an AUC value of 0.72 is still sufficient. This is the final model that we add to the prediction tool for the THA data.

For the TKA data in Table 5.4, the model with the highest AUC value of 0.744 is the LOGR model with the embedded features. It uses 13 features. Due to the high AUC and the relatively low number of features, we select this model for the prediction tool. The RF model with filter features is also a promising model as it only needs 8 features and has a relatively high AUC. We add the RF model to our selection of models for the prediction tool as well. No other models score significantly better in AUC value or use significantly less features. Thus, we will only add two models from the TKA data to the prediction tool.

Not all models perform better when they use more features. Particularly, for the TKA data, the filter FSM which only has eight features performs relatively well. The filter FSM for the TKA data uses one of lowest number of features but still has four models which are in the top 10 models with the highest AUC. In contrast, the model with the highest AUC for the THA data uses all 22 features. For the TKA data, the RF model with filter features has a better AUC than the RF model with all remaining features. This could be explained by the concept of overfitting. Since the TKA data is even smaller than the THA data, it is harder for the models trained on the TKA training data to generalise to test data. Using too many features in this case could cause a model to overfit to the training data, which makes the model worse at generalising (Jović et al., 2015). This means that the models could perform better if we use less features. This is indeed the case because the filter FSM performs better for the RF model. However, whether less features yields a higher AUC may depend on the ML method used, because using less features does not always improve the AUC as can be seen in Table 5.4.

According to literature, wrapper FSM usually should yield better results than the filter FSM (Jović et al., 2015). However, despite the long running time as shown in Table 5.2, for the TKA dataset, the feature subset from the filter FSM trained ML models with higher AUC values than the models trained with the wrapper FSM features. Academic literature mentions that wrapper FSM could face overfitting problems (Kohavi & Sommerfield, 1995) and the feature subsets of the wrapper FSM are biased towards the ML method used (Jović et al., 2015). It could be the case that the ML models who use the features of the wrapper FSM are overfitting on the training data and thus perform worse than the relatively simple filter FSM.

5.3.2 Learning curves of promising models

The learning curves for the selected models for the THA data and TKA data are shown in Figure 5.5 and Figure 5.6, respectively. The x-axis represents the number of rows used in the training data and the y-axis represents the AUC values of the corresponding models. In Figure 5.5, all four learning curves have increasing lines for the test data when more data is added. When all training data is used, Figure 5.5B, Figure 5.5C and Figure 5.5D have small gaps between the training and test AUC values. This is desirable because this means the model has similar performance for the train and test data. In contrast, Figure 5.5A has a large gap between the training and test AUC values. Since the training AUC is much higher than the testing AUC, this could be an indication of overfitting. This could be caused by the fact that the training data is relatively small compared to the literature in Chapter 2. For Figure 5.5A, the training AUC is slowly decreasing while the testing AUC is slowly increasing. This indicates that both AUC values could come closer to each other if there would be more data. Since the AUC values are increasing for all learning curves in Figure 5.5, having more training data could improve the corresponding models.

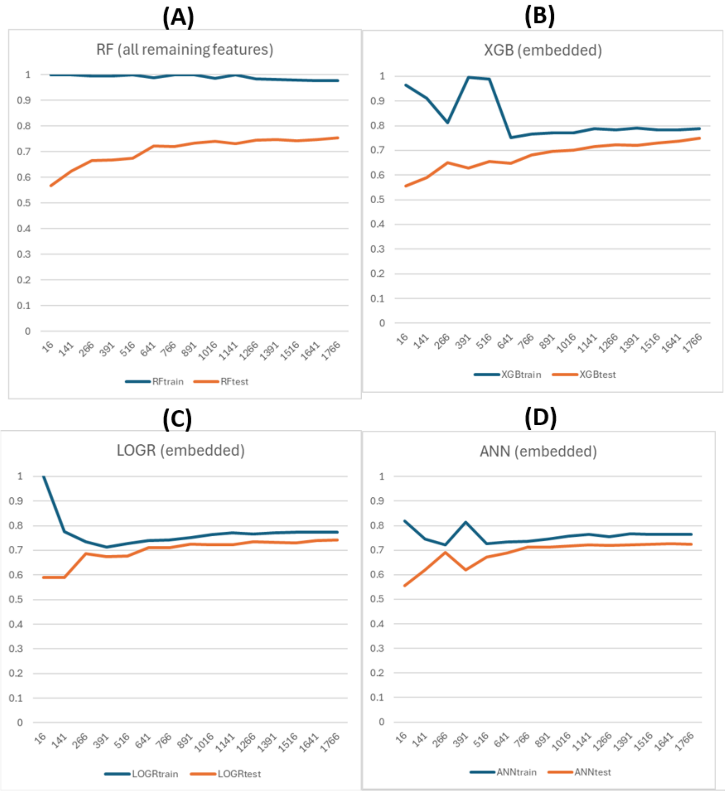


Figure 5.5: Learning curves of selected models for the THA data which are: (A) RF with all remaining features, (B) XGB with embedded features, (C) LOGR with embedded features, and (D) ANN with embedded features.

The two learning curves in Figure 5.6 show a somewhat different pattern than the learning curves in Figure 5.5. Figure 5.6A has a decreasing training AUC and an increasing testing AUC. However, the testing AUC seems to stop increasing when all the training data is used for modelling. Since there is still a significant gap between the training and testing AUC, more data is needed to draw conclusions on whether the testing AUC can increase further. Figure 5.6B has a testing AUC which increases above the training AUC. This could mean that the testing data is easier to predict for the LOGR model with embedded features than the training data. This phenomenon can happen when the test data is similar to parts of the training data which occur frequently in the training data. In this case, the corresponding model is better in predicting the test data than the training data because the training data also contains less represented rows which are harder to make long LOS predictions for. A way to make sure the training and testing AUC are closer together, is to increase the total amount of data. In short, more data is beneficial for both the THA and TKA ML models. We expect the TKA models especially to benefit from more data, since the TKA training dataset is even smaller (1412 rows) than the THA training dataset (1766 rows).

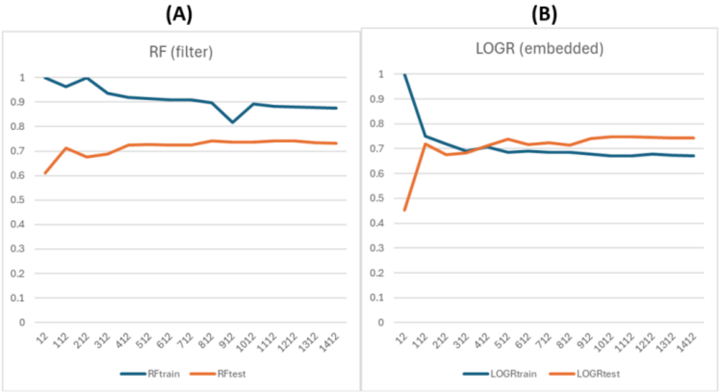


Figure 5.6: Learning curves of selected models for the TKA data which are: (A) RF with filter features, and (B) LOGR with embedded features.

5.3.3 Feature importance of promising models

To visualise the feature importance of the selected models, we calculate the scaled feature importance with the varImp function from the caret library in R. Figure 5.7 shows the feature importance of each model that we add to the prediction tool based on the THA data, and Figure 5.8 shows the same but for the models based on the TKA data. The feature lists include the dummy features. Features with a high feature importance affect the predictive ability of the corresponding model the most.

In Figure 5.7, the Age feature is consistently the feature with the highest importance value. Especially the XGB model with embedded features (Figure 5.7B) deems the Age feature as much more important than the rest. The feature importance values of the RF model with all remaining features (Figure 5.7A) have a very unequal distribution. The top eight features have a significantly higher feature importance than the rest of the features. Figure 5.7C and Figure 5.7D have a slightly more balanced distribution of feature importance. Overall, the most important features for the THA data seem to be Age, ASA score, Surgical_approach, and RESULT_Hemoglobin.

Concerning Figure 5.8, both models that are trained on the TKA data have the Age feature as most important feature. The ML model of Figure 5.8A primarily benefits from its top three most important features, while the rest of the features are significantly less important. In contrast, the feature importance of the model of Figure 5.8B is more evenly distributed. Overall, for the TKA data, the most important features seem to be Age, Surgeon, and RESULT_Urea.

For both Figure 5.7 and Figure 5.8, the feature importance ranking has significant differences among their models. One reason for this is that each ML model works different. For example, even though the training THA data is the same for all models in Figure 5.7, there are significant differences in feature ranking across the models. Another reason could be that the data of both the THA and TKA data is relatively small compared to literature. A small dataset may not be very representative of reality. This comes with the risk that ML models may consider random fluctuations or noise as important. Certain features may be statistically relevant for the training data, but do not have to be relevant in practice. To overcome risking such issues, collecting more data could help.

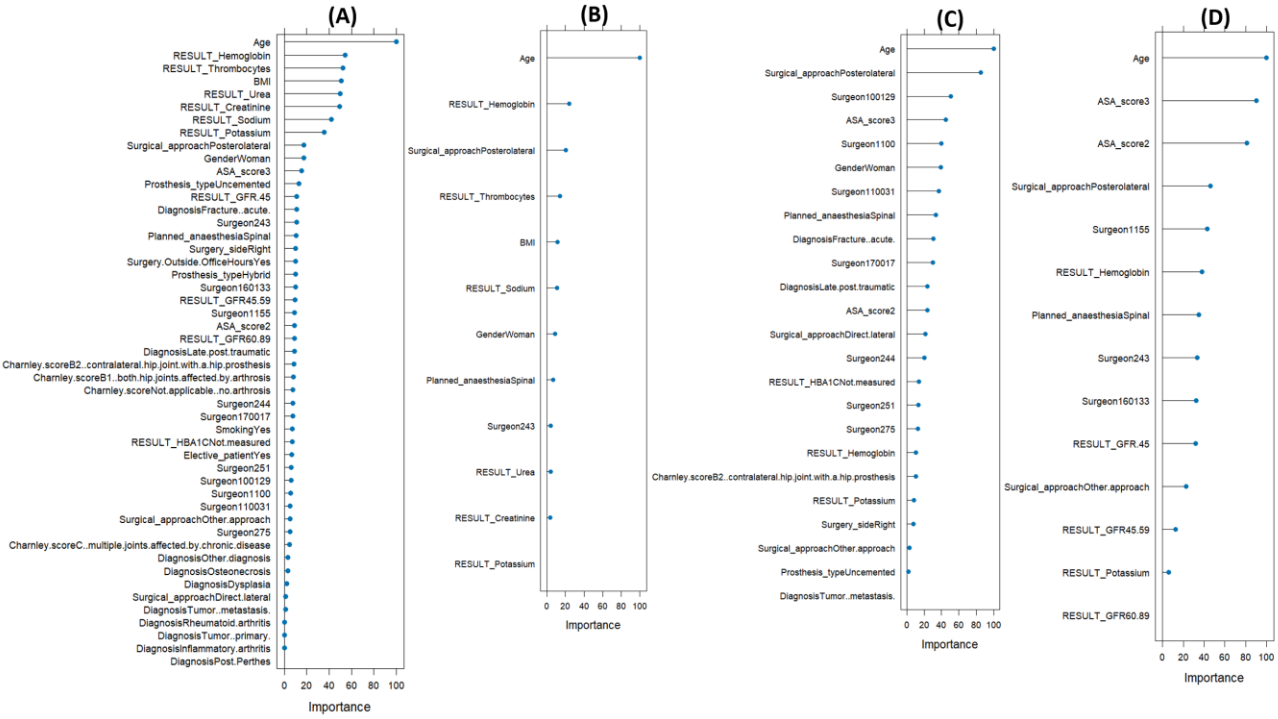


Figure 5.7: Scaled feature importance of selected ML models trained on THA data. The models are: (A) RF with all remaining features, (B) XGB with embedded features, (C) LOGR with embedded features, and (D) ANN with embedded features.

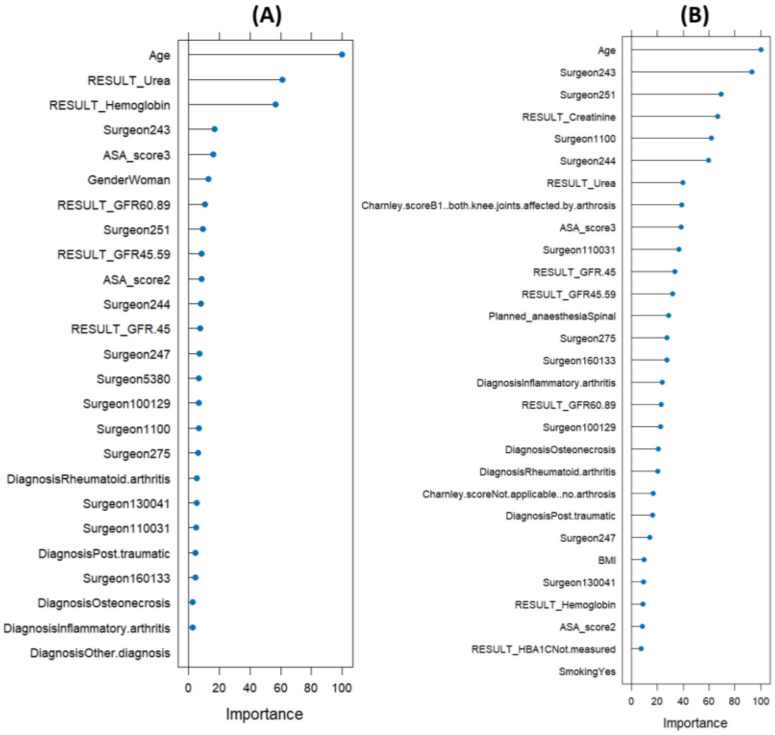


Figure 5.8: Scaled feature importance of selected ML models trained on TKA data. The models are: (A) RF with filter features, and (B) LOGR with embedded features.

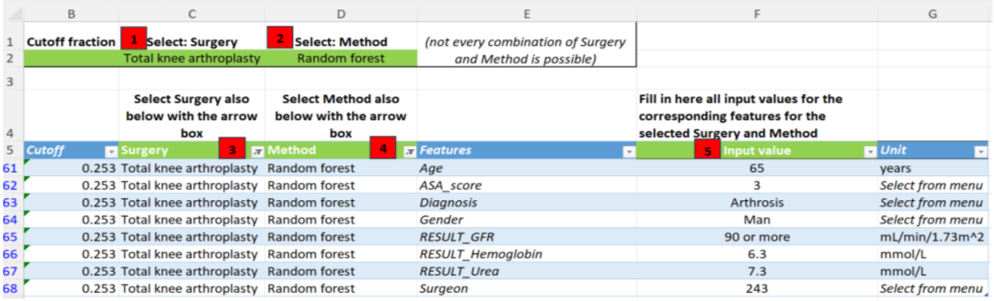
5.4 Prediction tool

To make sure that OCON can use the best performing ML models to make LOS predictions, we devise a prediction tool. We describe the tool in Section 5.4.1. In Section 5.4.2, we discuss the validation of the tool by asking OCON staff to fill in a user experience survey. We update the tool based on the feedback from the survey results in Section 5.4.3.

5.4.1 Prediction tool overview

The prediction tool consists of an interface in Excel where the user can input data, and a script in R which uses this data to make a prediction. To use the tool, it is required that the user has Excel and R installed on their computer. The tool can predict whether a certain new patient will experience a long or short LOS based on values inputted for the features by the user. The features differ per ML model used. It is important that the user makes sure all input values for each required feature is filled in, as this is required for the corresponding ML model. One of the input values is which combination of ML method and FSM should be used. Another one of the input values is the cutoff value. Figure 5.9 shows a screenshot of the Excel user interface. The red blocks show the steps that the user must perform to use the Excel file for the tool. The steps are as follows, and the cell locations are written between parentheses:

1. Fill in the Surgery (C2), which is either THA or TKA.
2. Fill in a machine learning method at the Method (D2).
3. Use the arrows to filter the chosen Surgery (C5).
4. Use the arrows to filter the chosen Method (D5).
5. Input the values for the corresponding features in column F. Categorical features can be selected from a dropdown menu.
6. The user must save the Excel file, such as by pressing Ctrl+S.

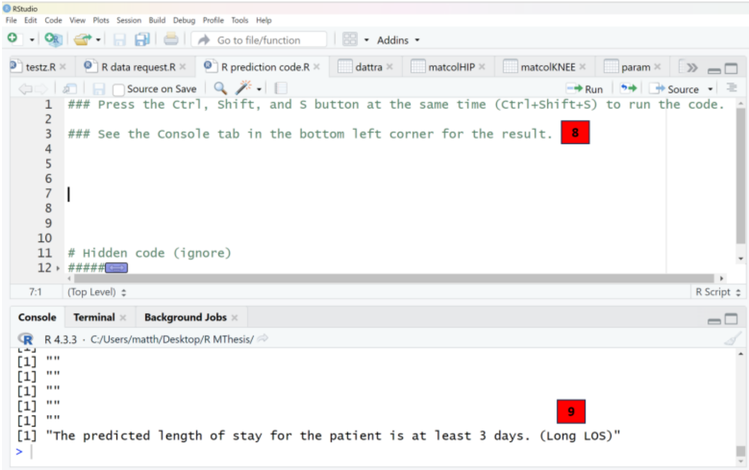


	B	C	D	E	F	G
1	Cutoff fraction	1 Select: Surgery	2 Select: Method	(not every combination of Surgery		
2		Total knee arthroplasty	Random forest	and Method is possible)		
3						
4		Select Surgery also below with the arrow box	Select Method also below with the arrow box		Fill in here all input values for the corresponding features for the selected Surgery and Method	
5	Cutoff	Surgery 3	Method 4	Features	5 Input value	Unit
61	0.253	Total knee arthroplasty	Random forest	Age	65	years
62	0.253	Total knee arthroplasty	Random forest	ASA_score	3	Select from menu
63	0.253	Total knee arthroplasty	Random forest	Diagnosis	Arthrosis	Select from menu
64	0.253	Total knee arthroplasty	Random forest	Gender	Man	Select from menu
65	0.253	Total knee arthroplasty	Random forest	RESULT_GFR	90 or more	mL/min/1.73m^2
66	0.253	Total knee arthroplasty	Random forest	RESULT_Hemoglobin	6.3	mmol/L
67	0.253	Total knee arthroplasty	Random forest	RESULT_Urea	7.3	mmol/L
68	0.253	Total knee arthroplasty	Random forest	Surgeon	243	Select from menu

Figure 5.9: Interface of prediction tool in Excel.

The next steps should be performed in the R script. Figure 5.10 shows a screenshot of the R script, and the steps are as follows:

7. Open the R file called: 'R prediction code'.
8. In the R file, press Ctrl+Shift+S to run the script, or press the Source button.
9. In the R file, in the bottom left corner, check the Console tab for the resulting prediction. This is either long LOS of at least 3 days, or a short LOS of less than 3 days.



```
### Press the Ctrl, Shift, and S button at the same time (Ctrl+Shift+S) to run the code.

### See the Console tab in the bottom left corner for the result.








# Hidden code (ignore)
#####
```

```
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] "The predicted length of stay for the patient is at least 3 days. (Long LOS)"
>
```

Figure 5.10: Code for prediction tool in R.

Whenever a ML model makes a prediction, it gives a fraction from 0 to 1 as output. The higher the value, the more likely it is according to the model that the corresponding patient will experience a long LOS. When this fraction is above a certain cutoff threshold, it predicts that the patient will experience long LOS. Decreasing this cutoff value makes the model more likely to predict a long LOS, while increasing the cutoff value does the opposite. The default cutoff value for each model used in the prediction tool is the cutoff value in the corresponding ROC curve with the highest sum of the sensitivity and the specificity. The ROC curves with the corresponding chosen cutoff values are shown in Figure 5.11 for the THA data and Figure 5.12 for the TKA data. The values between parentheses are the specificity and sensitivity of the cutoff value respectively. For example, in Figure 5.11, the cutoff value for the RF model with all remaining features (Figure 5.11A) is 0.191, and its specificity and sensitivity are 0.554 and 0.861, respectively. Suppose the model predicts a prediction fraction of 0.3. The value of 0.3 is higher than the used cutoff value of 0.191, which means that the model predicts a long LOS. In Figure 5.9, in cell location B2, the user of the prediction tool has the option to adjust this cutoff value manually to make the tool more or less likely to predict a long LOS. We also add a sheet called Information to Excel, which contains an overview about the available ML models, the number of features they require, their AUC, default cutoff value, sensitivity, and specificity.

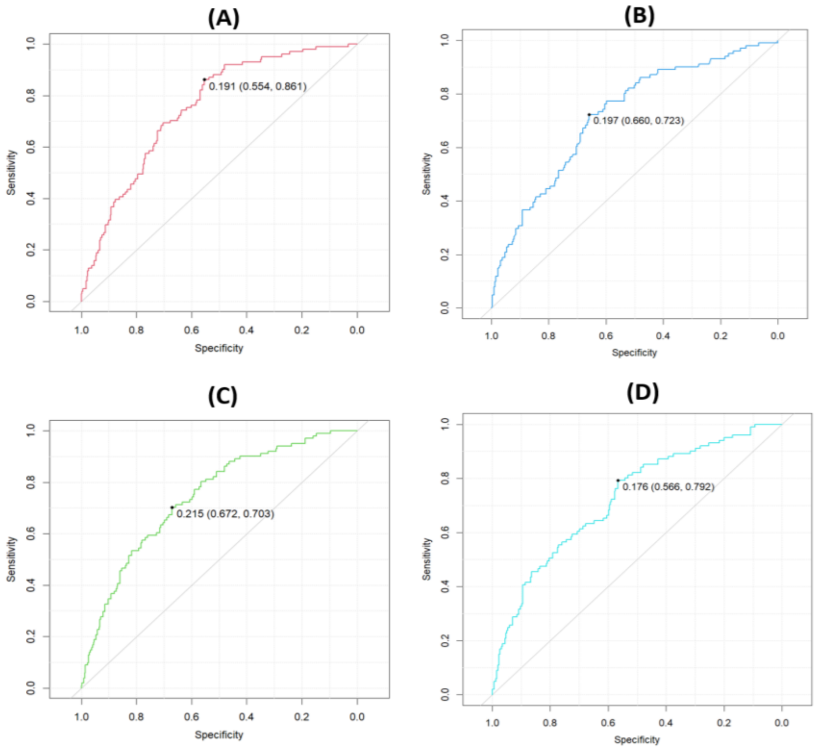


Figure 5.11: ROC curves of selected models for the THA data along with the best cutoff value and its corresponding specificity and sensitivity. The models for each plot are: (A) RF with all remaining features, (B) XGB with embedded features, (C) LOGR with embedded features, and (D) ANN with embedded features.

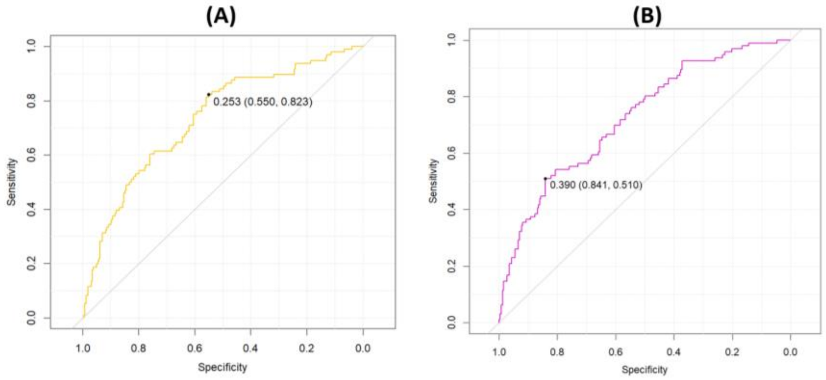


Figure 5.12: ROC curves of selected models for the TKA data along with the best cutoff value and its corresponding specificity and sensitivity. The models for each plot are: (A) RF with filter features, and (B) LOGR with embedded features.

5.4.2 Tool validation

We validate our prediction tool based on a user experience survey, namely the User Experience Questionnaire (UEQ, 2018). Appendix A shows the questionnaire. These questions of the survey are divided into six categories, namely:

- Attractiveness: a user’s impression of the tool.
- Perspicuity: how easy it is for the user to work with the tool.
- Efficiency: whether the user can use the tool without having to do needless tasks.
- Dependability: whether the user experiences being in control of the process.
- Stimulation: whether the tool stimulates or motivates the user to use it.
- Novelty: how innovative the tool feels to the user.

We asked two categories of OCON staff to fill in the questionnaire, which are medical staff who are potential users of the tool and non-medical staff who also work in the hospital. Figure 5.13 shows the graphs of the medical staff, and Figure 5.14 shows the graphs of the non-medical staff. Both figures also show plots for three grouped categories, which are: attractiveness, pragmatic quality, and hedonic quality. The pragmatic quality category consists of Perspicuity, Efficiency, and Dependability. This category refers to task-related aspects. The hedonic quality category refers to non-task-related aspects and consists of Stimulation and Novelty.

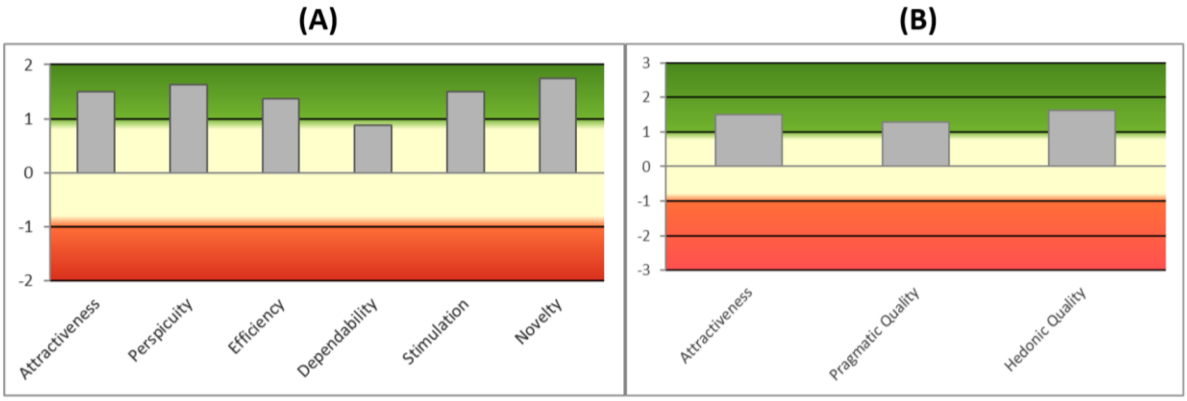


Figure 5.13: Graphs concerning the User Experience Questionnaire results of medical staff. The bar charts are: (A) mean scores for each of the six categories and (B) mean scores of three grouped categories.

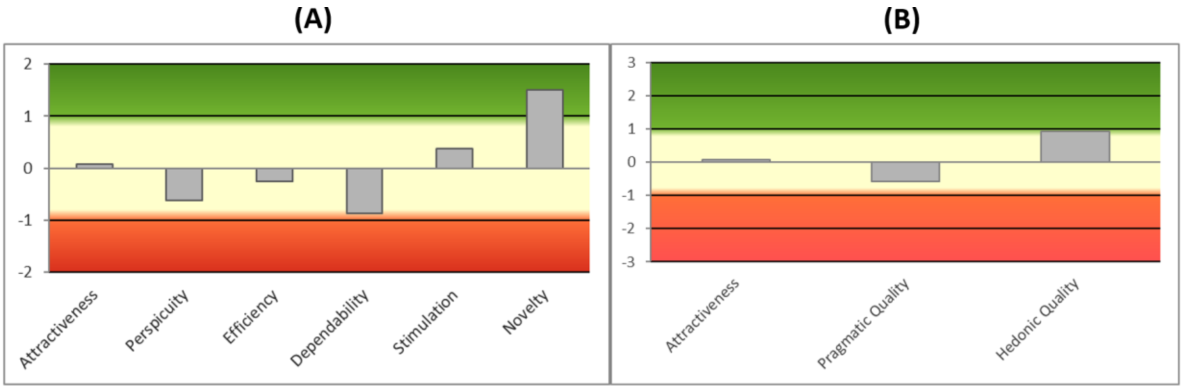


Figure 5.14: Graphs concerning the User Experience Questionnaire results of non-medical staff. The bar charts are: (A) mean scores for each of the six categories and (B) mean scores of three grouped categories.

The medical staff filled in significantly higher scores than the non-medical staff. Novelty is the highest scoring category, while Dependability scored the lowest across all categories. This gives the indication that the respondents think the tool is innovative, but they are not familiar with using it and may require some time to get used to it. Still, the medical staff assign good scores for the Dependability category. In contrast, the non-medical staff fill in lower scores in this category. Another category with stark contrasts between the medical and non-medical staff is the Perspicuity category. Essentially, the non-medical staff views the prediction tool as not very easy to work with and expects medical staff to struggle with using the tool. On the contrary, the medical staff do view it as easy to work with.

One non-medical respondent claims that the user would want to trust the tool that it already selects the best settings, and that potential users do not know what ML methods are and do not know the impact of choosing one ML method over another. This respondent also states that it is undesirable to work in two programs, and that it would be ideal for the user to only work with one program with one interface. This latter point could be an opportunity for further research. Another non-medical respondent thinks that certain aspects are too complex for potential users, such as selecting the ML model. This respondent also states that the Excel interface for inputting the features is well organised and that it is nice that the tool gives one clear prediction. The respondent also wonders is there is a way to show how reliable each prediction is. Despite the lower scores from the non-medical respondents, the hedonic quality still has a fair mean score, which indicates that these respondents also acknowledge the non-task-related qualities of the tool, and especially its novelty.

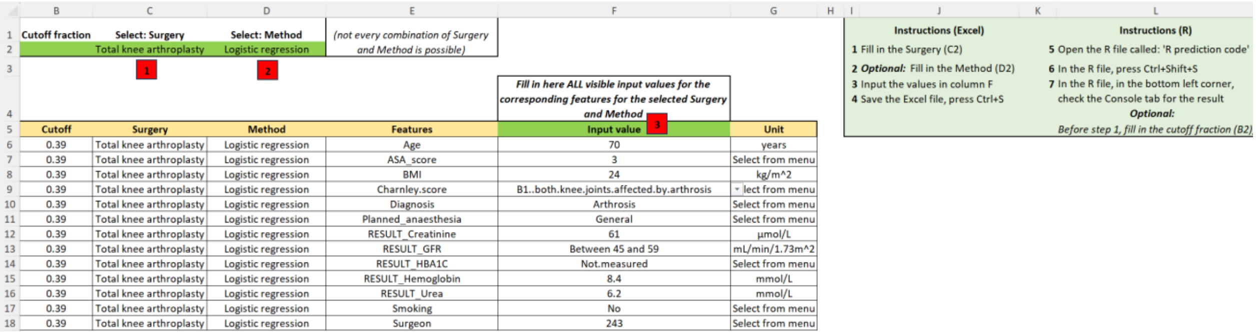
The medical respondents are positive about the tool and filled in good scores for all categories of the survey. One medical respondent states that the tool is very innovative and impressive. Another medical respondent states that the tool has nice explanations for the helpful method in the Excel interface.

5.4.3 Tool adjustments based on feedback

To improve the prediction tool, we incorporate the feedback given from the respondents. One respondent states that it would be beneficial if the tool already selects the best settings. What is defined as the best settings is subjective, because one user might value the AUC of a model more than the number of features to input or vice versa. Even though we offer the user freedom in selecting which ML method to use, we make sure that the tool uses a default cutoff value and a default ML method. The chosen default ML model for both THA and TKA is LOGR, because of its relatively low number of input features and its good AUC.

Another respondent asks whether it is possible to say something about how reliable the predictions by the tool are. We introduce an additional performance indicator to measure this. This is a number which is calculated as the absolute difference between the prediction fraction and the chosen cutoff value. A higher number indicates that the prediction fraction is further away from the cutoff value. Furthermore, we present this number as a percentage where 100% is the maximum difference of the prediction fraction with the cutoff value. A high percentage means that the given prediction is less likely to change if the user would input a slightly different cutoff value. This prediction score can be used as an indication of how certain the predictions are, where higher values indicate a higher certainty. A value below 10% indicates a relatively low certainty. Additionally, we make some adjustments to make sure the tool is easier to work with and feels more intuitive to the user. Figure 5.15 shows the updated Excel interface, and Figure 5.16 shows the updated R code.

- The selection of ML methods in step 2 is based on the selected surgery type.
- The THA and TKA settings each have a default ML method which is based on a good AUC and a low number of features. Changing the ML method is optional.
- Filtering the features works automatically based on selected ML method and surgery type. Thus, we remove step 3 and 4 as described in Section 5.4.1.
- We add a Feature Importance sheet with plots from Figure 5.7 and Figure 5.8 to give the user more insight in the importance of features for each ML model.
- Protect cells of the Information and Feature Importance sheets to reduce the risk of human error.



Cutoff fraction	Select: Surgery	Select: Method	(not every combination of Surgery and Method is possible)		
	Total knee arthroplasty	Logistic regression			
				Fill in here ALL visible input values for the corresponding features for the selected Surgery and Method	
Cutoff	**Surgery**	**Method**	**Features**	**Input value**	**Unit**
0.39	Total knee arthroplasty	Logistic regression	Age	70	years
0.39	Total knee arthroplasty	Logistic regression	ASA_score	3	Select from menu
0.39	Total knee arthroplasty	Logistic regression	BMI	24	kg/m^2
0.39	Total knee arthroplasty	Logistic regression	Charnley.score	B1..both.knee.joints.affected.by.arthrosis	lect from menu
0.39	Total knee arthroplasty	Logistic regression	Diagnosis	Arthrosis	Select from menu
0.39	Total knee arthroplasty	Logistic regression	Planned_anaesthesia	General	Select from menu
0.39	Total knee arthroplasty	Logistic regression	RESULT_Creatinine	61	µmol/L
0.39	Total knee arthroplasty	Logistic regression	RESULT_GFR	Between 45 and 59	mL/min/1.73m^2
0.39	Total knee arthroplasty	Logistic regression	RESULT_HBA1C	Not.measured	Select from menu
0.39	Total knee arthroplasty	Logistic regression	RESULT_Hemoglobin	8.4	mmol/L
0.39	Total knee arthroplasty	Logistic regression	RESULT_Urea	6.2	mmol/L
0.39	Total knee arthroplasty	Logistic regression	Smoking	No	Select from menu
0.39	Total knee arthroplasty	Logistic regression	Surgeon	243	Select from menu

Instructions (Excel)

1 Fill in the Surgery (C2)

2 ***Optional:*** Fill in the Method (D2)

3 Input the values in column F

4 Save the Excel file, press Ctrl+S

Instructions (R)

5 Open the R file called: 'R prediction code'

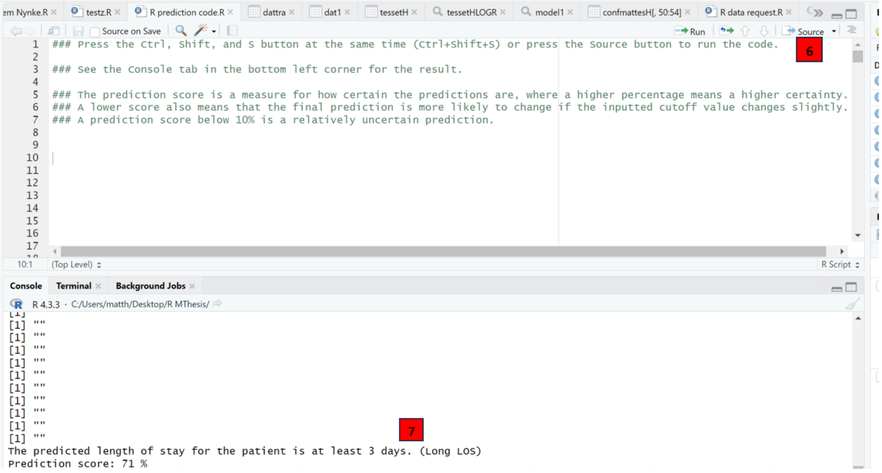
6 In the R file, press Ctrl+Shift+S

7 In the R file, in the bottom left corner, check the Console tab for the result

Optional:

Before step 1, fill in the cutoff fraction (B2)

Figure 5.15: Updated interface of prediction tool in Excel based on feedback.



```
### Press the Ctrl, Shift, and S button at the same time (Ctrl+Shift+S) or press the Source button to run the code.

### See the Console tab in the bottom left corner for the result.

### The prediction score is a measure for how certain the predictions are, where a higher percentage means a higher certainty.
### A lower score also means that the final prediction is more likely to change if the inputted cutoff value changes slightly.
### A prediction score below 10% is a relatively uncertain prediction.
```

```
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
[1] ""
The predicted length of stay for the patient is at least 3 days. (Long LOS)
Prediction score: 71 %
```

Figure 5.16: Updated R code of prediction tool based on feedback.

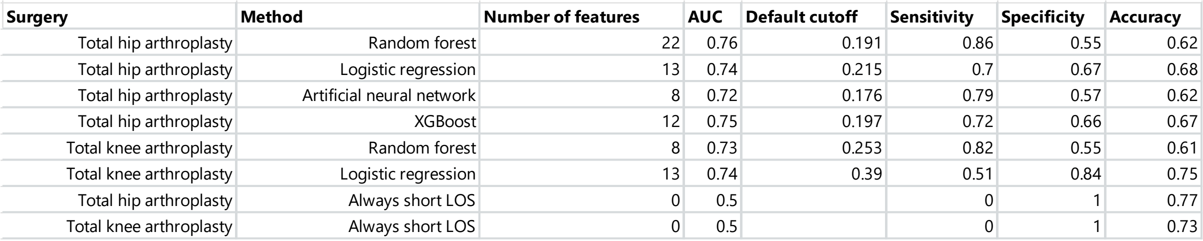
5.5 Predicted versus actual values

We supply each ML model in the prediction tool with default cutoff values, which are necessary for the tool to decide whether it predicts a patient to have a long LOS or not. We test the ML models with the default cutoff values on the test data. This means the models make predictions based on the features in the test data. To compare the predicted LOS values with the actual values in the test data, we measure the accuracy, which is the fraction of correctly predicted outcomes. Figure 5.11 and Figure 5.12 show the used default cutoff value for each ML model. Table 5.5 compares the predictions with the actual values for the LongLOS, and it shows the frequency of each combination of predicted and actual values for each ML method. It also shows the results when a model always predicts a short LOS, which occurs significantly more often in the data than long LOS instances. The values between brackets are the fractions of the total amount of instances for the corresponding frequencies. Table 5.6 shows the accuracy of each model described in Table 5.5, as well as their sensitivity and specificity.

Table 5.5: Distribution of predictions versus actual LongLOS values for each ML method.

Prediction	Actual	RFHIP_Freq	LOGRHIP_Freq	ANNHIP_Freq	XGBHIP_Freq	RFKNEE_Freq	LOGRKNEE_Freq	Always Short LOS (THA)	Always Short LOS (TKA)
Short LOS	Short LOS	192 (0.43)	229 (0.52)	193 (0.44)	224 (0.51)	141 (0.40)	217 (0.61)	341 (0.77)	258 (0.73)
Long LOS	Short LOS	149 (0.34)	112 (0.25)	148 (0.33)	117 (0.26)	117 (0.33)	41 (0.12)	0	0
Short LOS	Long LOS	20 (0.05)	30 (0.07)	22 (0.05)	28 (0.06)	20 (0.06)	47 (0.13)	101 (0.23)	96 (0.27)
Long LOS	Long LOS	81 (0.18)	71 (0.16)	79 (0.18)	73 (0.17)	76 (0.21)	49 (0.14)	0	0

Table 5.6: Performance values for each ML method.



Surgery	Method	Number of features	AUC	Default cutoff	Sensitivity	Specificity	Accuracy
Total hip arthroplasty	Random forest	22	0.76	0.191	0.86	0.55	0.62
Total hip arthroplasty	Logistic regression	13	0.74	0.215	0.7	0.67	0.68
Total hip arthroplasty	Artificial neural network	8	0.72	0.176	0.79	0.57	0.62
Total hip arthroplasty	XGBoost	12	0.75	0.197	0.72	0.66	0.67
Total knee arthroplasty	Random forest	8	0.73	0.253	0.82	0.55	0.61
Total knee arthroplasty	Logistic regression	13	0.74	0.39	0.51	0.84	0.75
Total hip arthroplasty	Always short LOS	0	0.5		0	1	0.77
Total knee arthroplasty	Always short LOS	0	0.5		0	1	0.73

In Table 5.6, the method with the highest accuracy (0.77) for the THA data is when a model always predicts a short LOS. The accuracy equals the fraction of the test data which are short LOS patients. Even though the accuracy is the highest, the AUC value is the lowest (0.5). A model which always predicts a short LOS has a sensitivity of 0 because it is unable to predict long LOS patients, which is undesirable. This shows that accuracy would be an unsuitable performance indicator for measuring the quality of a prediction model in this case. The model which always predicts a short LOS has a lower accuracy for the TKA data, because the fraction of short LOS patients in this dataset is small than the THA data. We add the accuracy of each ML model to the Information sheet so that the user gets more insight into the performance of each ML model.

5.6 Summary

In Chapter 5, we test the 68 ML models described in Chapter 4 on the test data. We select the most promising ML models based on their AUC and number of features. We select four ML models for the THA data and two models for the TKA data. The selected models are:

- For the THA dataset:
    - Random forest with all remaining features.
    - Logistic regression with embedded features.
    - XGBoost with embedded features.
    - Artificial neural network with embedded features.
- For the TKA dataset:
    - Random forest with filter features.
    - Logistic regression with embedded features.

Overall, the most important features for the THA data seem to be Age, ASA score, Surgical_approach, and RESULT_Hemoglobin. For the TKA dataset, the most important features seem to be the Age, Surgeon, and RESULT_Urea.

We incorporate these selected models into a prediction tool, which consists of an Excel interface and an R code. In the Excel interface, users select the type of surgery and input feature values. The R code plugs the data into the selected ML model and makes a prediction concerning whether a patient is expected to experience a long LOS. We ask OCON staff to fill in the User Experience Questionnaire to validate the tool, and we adjust our tool based on the received feedback. By using learning curves, we conclude that more input data could improve the performance of the ML models. We elaborate on the discussion of our research in Chapter 6.

Chapter 6: Discussion

In this chapter, we discuss the implications of our research. In Section 6.1, we conclude our research and provide a short summary. Section 6.2 discusses the limitations that occurred throughout the research, and Section 6.3 mentions the theoretical and practical contributions. We mention our recommendations to OCON in Section 6.4, and we finalise this chapter with Section 6.5, where we discuss opportunities for further research.

6.1 Conclusion

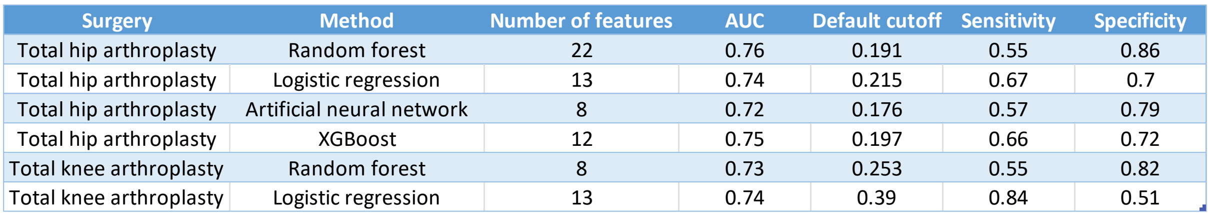
In Chapter 1, we analysed the problem that OCON is facing. We interviewed OCON staff and created a problem cluster based on the findings, as is shown in Figure 1.1. The resulting research problem is:

> *"A method should be devised which is able to make better predictions for the postoperative length of stay for patients who underwent a primary TKA/THA."*

OCON provides a patient registration database called HiX which contains patient-related data. We made a selection of relevant features in HiX for predicting the length of stay (LOS) for primary total knee arthroplasty (TKA) and primary total hip arthroplasty (THA) surgeries based on literature. We extracted details from as many surgeries as were available. We prepared the data in such a way that it can be used on feature selection methods (FSMs) and machine learning (ML) models, and we split the data based on whether a THA or TKA is performed. We used three different FSMs to make sure irrelevant features are excluded to examine if this improves ML models. Using too many features could potentially cause overfitting and make the ML models worse. We also included a fourth dataset with all remaining features where we did not use an FSM. Afterwards, we applied the four datasets on eight different ML models. The most important metric for measuring the performance of such models is the AUC of the corresponding ROC curves. The number of features for each ML model is another metric for measuring how promising a model is. We selected the six most promising models and created a prediction tool where a user can make LOS predictions of a patient based on inputted feature values. Each selected ML model uses different features. Figure 5.7 and Figure 5.8 shows their feature importance. Table 6.1 shows an overview of the final selected models and their performance metrics. Each row in this table signifies a ML model, which is about a certain type of surgery, a ML method, used default cutoff value, and about the number of used input features. Each row also contains information about the performance of the model, namely its AUC, sensitivity, and specificity. We validated the prediction tool by asking OCON staff to fill in the User Experience Questionnaire and applied feedback to improve it.

We conclude our research with the updated prediction tool which OCON staff (medical or non-medical staff) can use to estimate whether a patient is expected to experience a long LOS. The tool can be used as an advisory tool to help OCON staff make better predictions by using historical data, and so our research solves the research problem.

Table 6.1: Overview of performance metrics of each of the selected ML models for the prediction tool.



Surgery	Method	Number of features	AUC	Default cutoff	Sensitivity	Specificity
Total hip arthroplasty	Random forest	22	0.76	0.191	0.55	0.86
Total hip arthroplasty	Logistic regression	13	0.74	0.215	0.67	0.7
Total hip arthroplasty	Artificial neural network	8	0.72	0.176	0.57	0.79
Total hip arthroplasty	XGBoost	12	0.75	0.197	0.66	0.72
Total knee arthroplasty	Random forest	8	0.73	0.253	0.55	0.82
Total knee arthroplasty	Logistic regression	13	0.74	0.39	0.84	0.51

6.2 Limitations

During the execution of our research, we came across various limitations. The first limitation is that the HiX database does not contain all features which we requested. A lot of features who are used in literature for LOS prediction are not stored in HiX, and thus cannot be extracted for our research. Examples of this are the Timed Up and Go test and a patient's mental health. The second limitation is that a lot of features are stored in HiX as written text. The act of structuring written text is very challenging, as every person can write their text in a different way, and it can contain spelling errors. This means that a feature with written text has a very high number of unique values, which makes it hard for a ML model to generalise information. Thus, we limit our research to features which are not stored in a written text format but in a structured and limited format. The third data limitation is that 26% of the received data is unusable, because the corresponding surgeries contained no data which are required for calculating the LOS, such as discharge date and time. The fourth data limitation is that there are missing values for certain features. Even though we perform data imputation, the imputed values are never as good as the real values. The data imputation method imputes a value which the method expects to be there, and the real value can be different than the imputed value. The fifth data limitation is that we did not receive most of the features that we requested even though they are present in HiX. We requested 219 features which are available in HiX, but we received 39 of them in our data request. The sixth limitation is that various features contained a few wrong values, which required manual fixing or resulted in being forced to remove some surgeries from the data. Finally, the time between when we handed in our data request and when we received our final dataset took longer than expected.

6.3 Theoretical and practical contribution

The AUC values for the THA and TKA models are comparable with what the academic literature in Table 2.1. The best performing models in academic literature found AUC values between 0.69 and 0.83 for TKA data, and they found AUC values between 0.73 and 0.87 for THA data. One key difference between our research and the academic literature is that most of the sources in Table 2.1 have at least 100,000 surgeries to analyse per THA or TKA surgery type. In contrast, our total THA dataset contains 2208 surgeries and our total TKA dataset contains 1766 surgeries, which is significantly less compared to the amount of data that is used in the literature. Our datasets also contain different features than the academic literature. One key difference is that the literature often uses comorbidity scores, which are not present in our received data. An example of such a classification system used in literature is All Patients Refined Diagnosis Related Groups (APRDRG). Despite the difference in features and the significantly smaller dataset, our created ML models have AUC values which are comparable to literature. Finally, the literature described in Table 2.1 perform their research on either an American or Asian population. In contrast, our research is performed on a European population, namely the Dutch population.

To the best of our knowledge, there is no interactive prediction tool in academic literature with an adjustable cutoff input value and ML method which can make a prediction on whether a primary THA or TKA patient is expected to experience a long LOS or not. With the prediction tool's estimations, OCON staff can gain insight into knowledge gained from historical patient data, which should improve their predictions concerning whether THA/TKA patients will experience a long LOS. This can help solve the problems as defined in the problem cluster in Figure 1.1 in Chapter 1. For example, making better predictions can reduce patient uncertainty, improve bed capacity utilisation, and reduce the risk of a congested planning.

6.4 Recommendations

First and foremost, we recommend OCON staff to use the prediction tool. It can function as an advisory tool to improve predictions concerning the LOS of THA/TKA patients. We recommend using the ML models with relatively few features to make sure filling in feature values does not become a bothersome chore. We also recommend using the default cutoff value in the beginning when people start using the tool for the first time. Once users feel like they want to the tool to become better at identifying long LOS patients, they can decrease the cutoff value slightly, such as in steps of 0.05. We also recommend users to use the default ML models for each type of surgery, which should make the tool easier to work with.

In order to measure the effectiveness of the tool, we recommend that users keep track of the predictions done by the model, as well as the actual LOS. This can visualise how the tool performs compared new data. This can be useful in tuning the tool. For example, users can input a lower cutoff value than the default cutoff value if they feel like the model is favouring identifying short LOS patients over identifying long LOS patients. Lowering the cutoff value makes the tool more likely to predict a long LOS, which comes at the expense of the tool becoming worse at identifying short LOS patients. This would mean that the sensitivity of the corresponding ML model increases, while its specificity decreases.

If OCON desires to further improve the tool, then the models can be trained again on newly added data. However, retraining the models and implementing them in the prediction tool requires a significant amount of effort. Especially adding more features would be a challenging task. Adding new features is not limited to HiX only, as new features can also be added manually in an Excel sheet for example. Adding more surgeries for the features that are already being used by the models in the prediction tool would be easier, but it still requires analysing the R code and making sure it works with newly added data. For example, adding new data might make the FSMs select different features, and it also affects the feature importance.

Finally, the prediction tool interface could be improved by the IT department. Currently, the tool requires the user to use Excel as well as R. To make the tool more user-friendly, one could add macro buttons to the Excel which runs the R code. This would not require the user anymore to open R at all.

6.5 Further research

The performance of ML models highly depends on the input data. Poor input data results in poor ML models. Collecting more surgeries and more features increases the input data, and extra data can train the ML models better and improve their AUC values. An option for further research could be to make sure the R code can work with the new data and create new ML models.

One of the best performing models in the literature described in Table 2.1 is the ANN. The creation of the ANN model in our research was limited to one hidden layer. Ramkumar et al. (2019a) uses an ANN with multiple hidden layers. Exploring the effectiveness of ANN for the data of this research is also a further research opportunity.

Finally, another way to benefit from the prediction tool is by combining the predictions with the capacity planning. For example, the planning for each patient bed at the bed department can be adjusted based on the LOS predictions of the prediction tool. Exploring how LOS predictions can impact and improve scheduling problems is another opportunity for further research.

References

Abbink, R. J. C. M. (2021). *Optimizing operations at an orthopaedic hospital. Forecasting patient distributions and implementing strategies.* [Thesis]. https://essay.utwente.nl/88623/

Abiodun, E. O., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., & Alkhawaldeh, R. S. (2021). A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. In *Neural Computing and Applications* (Vol. 33, Issue 22). https://doi.org/10.1007/s00521-021-06406-8

Berrar, D. (2018). Bayes' theorem and naive bayes classifier. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3). https://doi.org/10.1016/B978-0-12-809633-8.20473-1

Bougrain, L. (2004). Practical introduction to artificial neural networks. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *37*(15). https://doi.org/10.1016/s1474-6670(17)31048-0

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7). https://doi.org/10.1016/S0031-3203(96)00142-2

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*. https://doi.org/10.1145/2939672.2939785

Chen, T. L. W., Buddhiraju, A., Seo, H. H., Shimizu, M. R., Bacevich, B. M., & Kwon, Y. M. (2023a). Can machine learning models predict prolonged length of hospital stay following primary total knee arthroplasty based on a national patient cohort data? *Archives of Orthopaedic and Trauma Surgery*. https://doi.org/10.1007/s00402-023-05013-7

Chen, T. L.-W., Buddhiraju, A., Costales, T. G., Subih, M. A., Seo, H. H., & Kwon, Y.-M. (2023b). Machine Learning Models Based on a National-Scale Cohort Identify Patients at High Risk for Prolonged Lengths of Stay Following Primary Total Hip Arthroplasty. *The Journal of Arthroplasty*. https://doi.org/10.1016/j.arth.2023.06.009

Colliot, O. (2023). A Non-technical Introduction to Machine Learning. In *Neuromethods* (Vol. 197). https://doi.org/10.1007/978-1-0716-3195-9_1

Ding, Z., Li, J., Xu, B., Cao, J., Li, H., & Zhou, Z. (2022). Preoperative High Sleep Quality Predicts Further Decrease in Length of Stay after Total Joint Arthroplasty under Enhanced Recovery Short-stay Program: Experience in 604 Patients from a Single Team. Orthopaedic Surgery. https://doi.org/10.1111/os.13382

Ding, Z., Xu, B., Liang, Z., Wang, H., Luo, Z., & Zhou, Z. (2019). Limited Influence of Comorbidities on Length of Stay after Total Hip Arthroplasty: Experience of Enhanced Recovery after Surgery. Orthopaedic Surgery, 12(1), 153–161. https://doi.org/10.1111/os.12600

Edwards, C., & Raskutti, B. (2004). The effect of attribute scaling on the performance of support vector machines. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, *3339*. https://doi.org/10.1007/978-3-540-30549-1_44

El-Othmani, M. M., Zalikha, A. K., & Shah, R. P. (2022). Comparative Analysis of the Ability of Machine Learning Models in Predicting In-hospital Postoperative Outcomes After Total Hip Arthroplasty. *Journal of the American Academy of Orthopaedic Surgeons*, *30*(20). https://doi.org/10.5435/JAAOS-D-21-00987

Elawady, Y. M. (2021). *Structureren van de patiëntenstroom van een Totale Heup- en Knie Prothese door vroegtijdige risicostratificatie : Adviesrapport over de implementatie van de "Modified Elderly Mobility Scale" in de orthopedie.* [Thesis]. https://essay.utwente.nl/88580/

Elings, J., Hoogeboom, T., van der Sluis, G., & van Meeteren, N. (2014). What preoperative patient-related factors predict inpatient recovery of physical functioning and length of stay after total hip arthroplasty? A systematic review. Clinical Rehabilitation, 29(5), 477–492. https://doi.org/10.1177/0269215514545349

Elssied, N. O. F., Ibrahim, O., & Osman, A. H. (2014). A novel feature selection based on one-way ANOVA F-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, *7*(3). https://doi.org/10.19026/rjaset.7.299

filterVarImp function - RDocumentation. (n.d.). Www.rdocumentation.org. Retrieved July 1, 2024, from https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/filterVarImp

Fitzmaurice, G. M., & Laird, N. M. (2015). Binary Response Models and Logistic Regression. In *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*. https://doi.org/10.1016/B978-0-08-097086-8.42060-X

Gabriel, R. A., Sharma, B. S., Doan, C. N., Jiang, X., Schmidt, U. H., & Vaida, F. (2019). A Predictive Model for Determining Patients Not Requiring Prolonged Hospital Length of Stay after Elective Primary Total Hip Arthroplasty. *Anesthesia and Analgesia*, *129*(1). https://doi.org/10.1213/ANE.0000000000003798

Genuer, R., Poggi, J. M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*(14). https://doi.org/10.1016/j.patrec.2010.03.014

Han C, Liu J, Wu Y, Chong Y, Chai X and Weng X (2021) To Predict the Length of Hospital Stay After Total Knee Arthroplasty in an Orthopedic Center in China: The Use of Machine Learning Algorithms. *Front. Surg.* 8:606038. doi: 10.3389/fsurg.2021.606038

Hastie, T., Tibshirani, R., James, G., & Witten, D. (2021). An introduction to Statistical Learning with Applications in R (2nd Edition). *Springer Texts*, *102*.

Hossain, M. A., Noor, R. M., Yau, K. L. A., Azzuhri, S. R., Z'Aba, M. R., & Ahmedy, I. (2020). Comprehensive survey of machine learning approaches in cognitive radio-based vehicular Ad Hoc networks. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.2989870

Johannesdottir, K. B., Henrik Kehlet, Petersen, P. B., Aasvang, E. K., Helge, & Jørgensen, C. C. (2022). Machine learning classifiers do not improve prediction of hospitalization > 2 days after fast-track hip and knee arthroplasty compared with a classical statistical risk model. Acta Orthopaedica, 117–123. https://doi.org/10.2340/17453674.2021.843

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*. https://doi.org/10.1109/MIPRO.2015.7160458

Kohavi, R., & Sommerfield, D. (1995). Feature subset selection using the wrapper method: overfltting and dynamic search space topology. *KDD 1995 - Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*.

Kumar, V. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3). https://doi.org/10.6029/smartcr.2014.03.007

Lakomkin, N., Kothari, P., Dodd, A. C., VanHouten, J. P., Yarlagadda, M., Collinge, C. A., Obremskey, W. T., & Sethi, M. K. (2017). Higher Charlson Comorbidity Index Scores Are Associated With Increased Hospital Length of Stay After Lower Extremity Orthopaedic Trauma. Journal of Orthopaedic Trauma, 31(1), 21–26. https://doi.org/10.1097/bot.0000000000000701

Lee, J. H., & Huber, J. C. (2021). Evaluation of multiple imputation with large proportions of missing data: How much is too much? *Iranian Journal of Public Health*, *50*(7). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8426774/

Lee, K. F. A., Gan, W. S., & Christopoulos, G. (2021). Biomarker-informed machine learning model of cognitive fatigue from a heart rate response perspective. In *Sensors* (Vol. 21, Issue 11). https://doi.org/10.3390/s21113843

Li, H., Jiao, J., Zhang, S., Tang, H., Qu, X., & Yue, B. (2022). Construction and Comparison of Predictive Models for Length of Stay after Total Knee Arthroplasty: Regression Model and Machine Learning Analysis Based on 1,826 Cases in a Single Singapore Center. *Journal of Knee Surgery*, *35*(1). https://doi.org/10.1055/s-0040-1710573

Liu, Y., Mu, Y., Chen, K., Li, Y., & Guo, J. (2020). Daily Activity Feature Selection in Smart Homes Based on Pearson Correlation Coefficient. *Neural Processing Letters*, *51*(2). https://doi.org/10.1007/s11063-019-10185-8

Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random forest. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7473 LNCS*. https://doi.org/10.1007/978-3-642-34062-8_32

Mallampati, S. B., Seetha, H., & Batchu, R. K. (2023). PCB-LGBM: A Hybrid Feature Selection by Pearson Correlation and Boruta-LGBM for Intrusion Detection Systems. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 163). https://doi.org/10.1007/978-981-99-0609-3_37

Meek, C., Thiesson, B., & Heckerman, D. (2002). The Learning-Curve Sampling Method Applied to Model-Based Clustering. *Journal of Machine Learning Research*, *2*(3). https://doi.org/10.1162/153244302760200678

Mountrakis, G., Im, J., & Ogole, C. (2011). Support vector machines in remote sensing: A review. In *ISPRS Journal of Photogrammetry and Remote Sensing* (Vol. 66, Issue 3). https://doi.org/10.1016/j.isprsjprs.2010.11.001

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. In *Journal of Chemometrics* (Vol. 18, Issue 6). https://doi.org/10.1002/cem.873

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*(DEC). https://doi.org/10.3389/fnbot.2013.00021

Navarro, S. M., Wang, E. Y., Haeberle, H. S., Mont, M. A., Krebs, V. E., Patterson, B. M., & Ramkumar, P. N. (2018). Machine Learning and Primary Total Knee Arthroplasty: Patient Forecasting for a Patient-Specific Payment Model. *Journal of Arthroplasty*, *33*(12). https://doi.org/10.1016/j.arth.2018.08.028

Neuen, B. L., Ohkuma, T., Neal, B., Matthews, D. R., De Zeeuw, D., Mahaffey, K. W., Fulcher, G., Desai, M., Li, Q., Deng, H., Rosenthal, N., Jardine, M. J., Bakris, G., & Perkovic, V. (2018). Cardiovascular and renal outcomes with canagliflozin according to baseline kidney function data from the CANVAS program. *Circulation*, *138*(15). https://doi.org/10.1161/CIRCULATIONAHA.118.035901

Nham, F. H., Court, T., Zalikha, A. K., El-Othmani, M. M., & Shah, R. P. (2023). Assessing the predictive capacity of machine learning models using patient-specific variables in determining in-hospital outcomes after THA. *Journal of Orthopaedics*, *41*. https://doi.org/10.1016/j.jor.2023.05.012

Nhat-Duc, H., & Van-Duc, T. (2023). Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification. *Automation in Construction*, *148*. https://doi.org/10.1016/j.autcon.2023.104767

Ong, P-H., & Pua, Y-H. (2013). A prediction model for length of stay after total and unicompartmental knee replacement. The Bone & Joint Journal, 95-B(11), 1490–1496. https://doi.org/10.1302/0301-620x.95b11.31193

Organisatie. (2024). OCON. https://www.ocon.nl/over/organisatie

Osisanwo, F. Y., Akinsola, J. E. T., Hinmikaiye, J. O., Awodele, O., Olakanmi, O., & Akinjobi, J. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, *48*(3).

Ozsahin, D. U., Taiwo Mustapha, M., Mubarak, A. S., Said Ameen, Z., & Uzun, B. (2022). Impact of feature scaling on machine learning models for the diagnosis of diabetes. *Proceedings - 2022 International Conference on Artificial Intelligence in Everything, AIE 2022*. https://doi.org/10.1109/AIE57029.2022.00024

Pagan, M., Zarlis, M., & Candra, A. (2023). Investigating the impact of data scaling on the k-nearest neighbor algorithm. *Computer Science and Information Technologies*, *4*(2). https://doi.org/10.11591/csit.v4i2.p135-142

Papalia, R., Zampogna, B., Torre, G., Papalia, G. F., Vorini, F., Bravi, M., Albo, E., De Vincentis, A., & Denaro, V. (2021). Preoperative and perioperative predictors of length of hospital stay after primary total hip arthroplasty—our experience on 743 cases. *Journal of Clinical Medicine*, *10*(21). https://doi.org/10.3390/jcm10215053

Park, J., Zhong, X., Miley, E. N., & Gray, C. F. (2023). Preoperative Prediction and Risk Factor Identification of Hospital Length of Stay for Total Joint Arthroplasty Patients Using Machine Learning. *Arthroplasty Today*, *22*. https://doi.org/10.1016/j.artd.2023.101166

Pathak, R. (2020). Support vector machines: Introduction and the dual formulation. In *Lecture Notes in Electrical Engineering* (Vol. 643). https://doi.org/10.1007/978-981-15-3125-5_57

Polo Friz, H., Esposito, V., Marano, G., Primitz, L., Bovio, A., Delgrossi, G., Bombelli, M., Grignaffini, G., Monza, G., & Boracchi, P. (2022). Machine learning and LACE index for predicting 30-day readmissions after heart failure hospitalization in elderly patients. *Internal and Emergency Medicine*. https://doi.org/10.1007/s11739-022-02996-w

Ramkumar, P. N., Karnuta, J. M., Navarro, S. M., Haeberle, H. S., Iorio, R., Mont, M. A., Patterson, B. M., & Krebs, V. E. (2019a). Preoperative Prediction of Value Metrics and a Patient-Specific Payment Model for Primary Total Hip Arthroplasty: Development and Validation of a Deep Learning Model. *Journal of Arthroplasty*, *34*(10). https://doi.org/10.1016/j.arth.2019.04.055

Ramkumar, P. N., Karnuta, J. M., Navarro, S. M., Haeberle, H. S., Scuderi, G. R., Mont, M. A., Krebs, V. E., & Patterson, B. M. (2019b). Deep Learning Preoperatively Predicts Value Metrics for Primary Total Knee Arthroplasty: Development and Validation of an Artificial Neural Network Model. *Journal of Arthroplasty*, *34*(10). https://doi.org/10.1016/j.arth.2019.05.034

Ramkumar, P. N., Navarro, S. M., Haeberle, H. S., Karnuta, J. M., Mont, M. A., Iannotti, J. P., Patterson, B. M., & Krebs, V. E. (2019c). Development and Validation of a Machine Learning Algorithm After Primary Total Hip Arthroplasty: Applications to Length of Stay and Payment Models. *Journal of Arthroplasty*, *34*(4). https://doi.org/10.1016/j.arth.2018.12.030

Rolink, A. T. (2023). *Improving OR-scheduling at an orthopaedic clinic.* [Thesis *Improving OR-scheduling at an orthopaedic clinic.*]. https://essay.utwente.nl/94212/

Sibia, U. S., MacDonald, J. H., & King, P. J. (2016). Predictors of Hospital Length of Stay in an Enhanced Recovery After Surgery Program for Primary Total Hip Arthroplasty. The Journal of Arthroplasty, 31(10), 2119–2123. https://doi.org/10.1016/j.arth.2016.02.060

Sibia, U. S., Waite, K. A., Callanan, M. A., Park, A. E., King, P. J., & MacDonald, J. H. (2017). Do shorter lengths of stay increase readmissions after total joint replacements? Arthroplasty Today, 3(1), 51–55. https://doi.org/10.1016/j.artd.2016.05.001

Sumaiya Thaseen, I., & Aswani Kumar, C. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University - Computer and Information Sciences*, *29*(4). https://doi.org/10.1016/j.jksuci.2015.12.004

UEQ. (2018). *User experience questionnaire (UEQ).* Ueq-Online.org. https://www.ueq-online.org/

Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3). https://doi.org/10.18637/jss.v045.i03

Van der Heijden, G. J. M. G., T. Donders, A. R., Stijnen, T., & Moons, K. G. M. (2006). Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: A clinical example. *Journal of Clinical Epidemiology*, *59*(10). https://doi.org/10.1016/j.jclinepi.2006.01.015

Van der Sluis, G. (2018). From me needs knee to knee needs me: transition of perioperative care for people that chose to have total knee replacement surgery by augmenting activity and personalized functional goalsetting. Zalsman Groningen BV. https://doi.org/10.26481/dis.20181109gs

varImp function - RDocumentation. (n.d.). Www.rdocumentation.org. https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/varImp

Wanner, C., Lachin, J. M., Inzucchi, S. E., Fitchett, D., Mattheus, M., George, J., Woerle, H. J., Broedl, U. C., Von Eynatten, M., & Zinman, B. (2018). Empagliflozin and clinical outcomes in patients with type 2 diabetes mellitus, established cardiovascular disease, and chronic kidney disease. *Circulation*, *137*(2). https://doi.org/10.1161/CIRCULATIONAHA.117.028268

Wei, C., Quan, T., Wang, K. Y., Gu, A., Fassihi, S. C., Kahlenberg, C. A., Malahias, M. A., Liu, J., Thakkar, S., della Valle, A. G., & Sculco, P. K. (2021). Artificial neural network prediction of same-day discharge following primary total knee arthroplasty based on preoperative and intraoperative variables. *Bone and Joint Journal*, *103-B*(8). https://doi.org/10.1302/0301-620X.103B8.BJJ-2020-1013.R2

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. In *Data Mining: Practical Machine Learning Tools and Techniques*.

Zalikha, A. K., Court, T., Nham, F., El-Othmani, M. M., & Shah, R. P. (2023). Improved performance of machine learning models in predicting length of stay, discharge disposition, and inpatient mortality after total knee arthroplasty using patient-specific variables. *Arthroplasty*, *5*(1). https://doi.org/10.1186/s42836-023-00187-2

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11). https://doi.org/10.21037/atm.2016.03.37

Appendix

Appendix A: User Experience Questionnaire

Please make your evaluation now.

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Example:

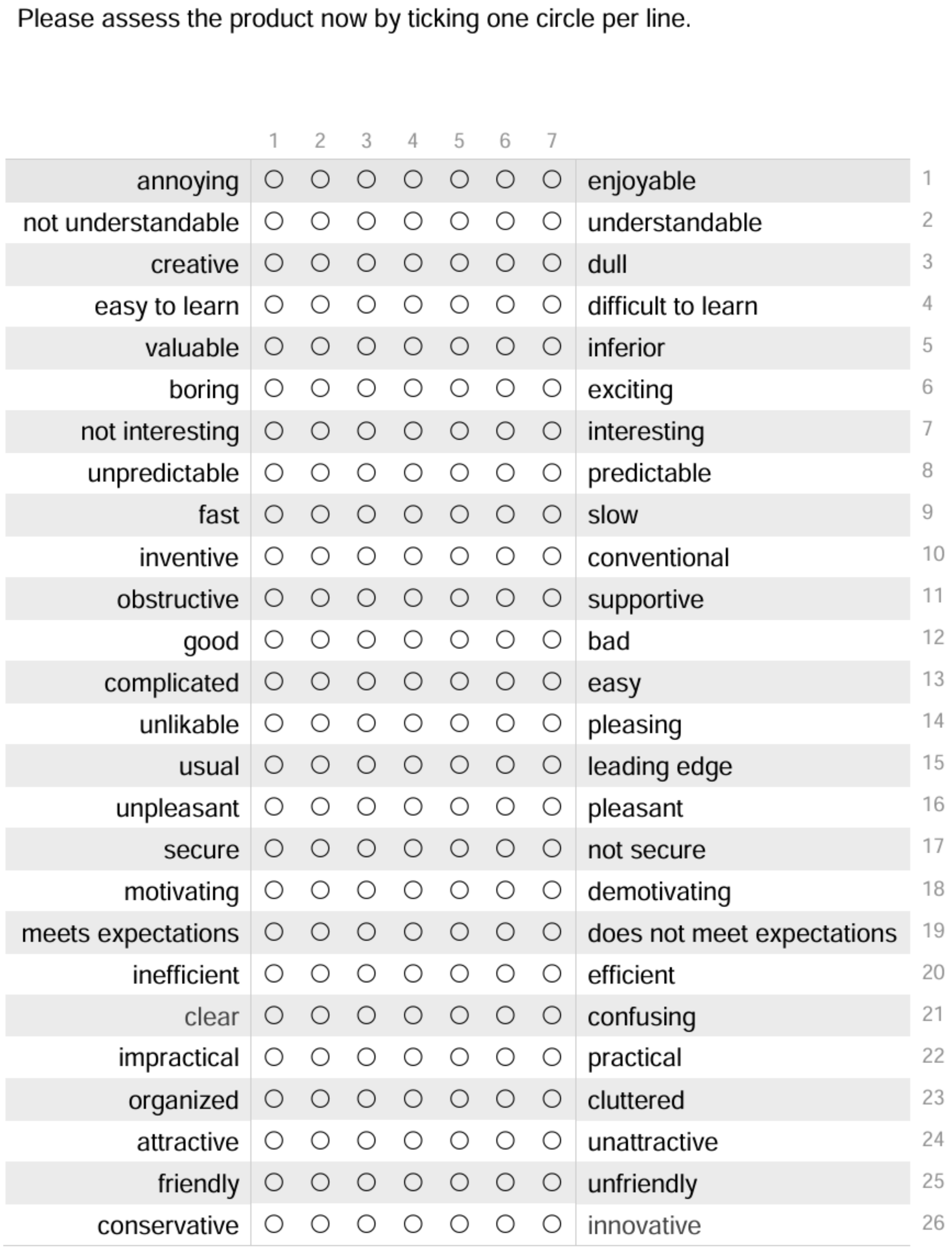
attractive	○	⊗	○	○	○	○	○	unattractive

This response would mean that you rate the application as more attractive than unattractive.

Please decide spontaneously. Don’t think too long about your decision to make sure that you convey your original impression.

Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!



Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26