UNIVERSITY
OF TWENTE.

University of Twente
Faculty of Behavioural,
Management and
Social Sciences

Predicting utilization in a Multi-Provider Ambulatory Allied Healthcare Organisation

Master thesis in Industrial Engineering & Management

Author
T.A. van der Poll

June 24, 2022

Executive summary

Introduction - This study aims to develop a prediction model that is able to predict session (combination of date, practitioner, location) utilization accurately, within a time horizon of one month, such that it influences a newly proposed scheduling method in order to reduce the variance in session utilization. A reduction in session utilization variance should help the new scheduling method to reduce the waiting time for new patients. Large growth in requests for patient appointments has lead to unacceptable waiting time for new patients (62.9% within 21 days in 2019). Therefore the decision was made to analyze and alter the scheduling process, with the main goal to reduce waiting time for new patients.

Scheduling is referring to the process of online (real time) scheduling one appointment to one session. For new patients, the time horizon is as soon as possible, generally within but not limited to one month. For recurrent appointments, the horizon varies between one appointment per month to appointment per year. Appointment cancellations do occur, but since data on cancellations is not available, cancellations are not considered. Processing times are standardized per activity and therefore assumed to be deterministic. The scheduling has multiple objectives, with as main goal to minimize the maximum lateness for new patients. The newly proposed scheduling system uses the Topsis MCDA method and has both patient needs and organization needs as objective. The patient-objectives are reducing distance (3.45 km in 2019), reducing waiting time for new patients (20.7 days in 2019). The organisational needs are increasing session utilization (0.897 in 2019), reducing variance in session utilization (0.0173 in 2019) and reducing fragmentation (1.4 in 2019). In the Topsis method, the criteria carry weights on their respective importance.

Due to nonlinear relationships (for instance rescheduling appointments to earlier dates), a lack of awareness about influential factors, the non-existence of data and structurally ill-registered activities, the prediction of utilization has been inadequate, with cross-validated error of .217. We aim to improve this.

Analysis - We identify three options for the analysis of prediction modeling. Time series require a regular series of observations, which we do not have due to workdays occurring only once and often interrupted series of workdays. In literature we find only one similar study, which uses a different unit of measure. Therefore we choose to use machine learning methods. From the larger group of alternative machine learning methods, we choose to test Multivariate Adaptive Regression Splines (MARS), Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting methods (LightGBM). We believe that these models are not too complex to train and do not require too much data compared to neural networks, but also allow for the flexibility. Another advantage is that the models automatically interact predictors. Finally, the XGBoost and LightGBM models are heavily maintained and well suited for use in practice. We develop models using these techniques. The models have a set of parameters which are used to minimize overfitting.

In order to prevent leaking test data in the training data, we actually predict the utilization that is added to the session after the appointment, and construct session utilizations from added utilization. We split the obtained appointments and corresponding added utilization in 80% training and 20% testing data. We use the 80% for model development. Training the model parameters is a time consuming task, therefore we use the tree-structured parzen estimator sampling technique. Using that technique, parameters

are chosen in promising neighbourhoods based on previous iterations. We 10-fold cross-validate 500 sets of parameters for each of the methods. We find that the LightGBM model reduces the error rate on the 20% test data to .041, which is significantly ($p < .001$) lower than the .079 and .048 test error rates for the MARS and XGBoost models respectively. The set of predictors is consistent across the methods. We find that the best predictors are a combination of

- The level of utilization of the session at the time of the prediction. This predictor is intuitive as it gives a bound to the utilization that can be added. For example, if currently the utilization is .9, then at most .1 can physically be added to the session.
- The ratio of workday scheduled for other activities (e.g. regional meetings). This predictor gives a degree of irregularity of the session in the sense that usually workdays only consist of patient appointments. Sessions with a large degree of irregularity often have idle time surrounding the activity, for travelling for instance.
- The practitioner. Utilization varies over practitioners due to for instance the specialty of the practitioner. This is a convenient way of
- The location of the session. Some locations are more heavily utilized than others due to for example the capacity of the location, or the closeness to other locations.

We investigate the performance of the developed prediction method in interaction with the scheduling method. We do this by means of simulation of the online scheduling method using the Topsis MCDA method. The purpose of this is to evaluate how the LightGBM prediction method influences the scheduling method in comparison to the originally proposed logistic regression model. In the simulation, we reschedule appointments that were requested during the period 1st of July 2019 until 31st of July 2019 to the set of sessions running from the first of July 2019 to the 31st of December 2020.

We experiment with the Topsis weights and compare the implementation with the LightGBM model with the Logistic regression model constructed in the preliminary study. If all weight available in the Topsis method is put on the level of utilization, we find that the LightGBM model significantly ($p < .001$) reduces the variance in utilization to the lowest found level of 0.067 compared to 0.083 with weights corresponding to the same model with default weights and 0.072 compared to the prediction model constructed in the preliminary study, but with all weights put to the level utilization.

The experiment results in an access (waiting) time of 25.2 days, 79 km distance, and an access time violation of 1.297. The utilization level is .603 and the fragmentation is 3.213. We conclude that the LightGBM model therefore produces better predictions of utilization in practice. We also find that - when weights are set with 2 times as much weight to access time violation and days deviation from target, while putting 50% less weight on utilization - waiting time for new patients can be reduced to 11.3 days, with 3.332 km distance, 2.554 days deviation and 0.015 access time violations. The fragmentation is 1.973. The utilization is then 0.616 and variance in utilization is 0.0961, so we should note altering the weights often neglects other the criteria, for instance an increase in the variance in utilization. However, a decreased variance in utilization should not be a goal on it's own, but rather something to exploit, and due to that we observe low test errors in model development, we believe that the LightGBM model is still better and should be - carefully - tested in practice.

Recommendations - Based on our analysis we recommend a set of interventions:

1. Structural and explicit registration of session activities such as which appointment belongs to which session, especially the current level of utilization and the ratio of time spent on other activities. This requires that appointments are explicitly connected to session schedules, which is currently not the case. Beside these, we also recommend to choose among the set of predictors used to train the LightGBM model. Some predictors contribute more than others, therefore one should prioritize based on the predictor gain for instance.

2. Structural registration of data on cancellations, rescheduled appointment and no-show appointments. We expect these factors to have predictive quality.

3. The implementation of the new scheduling method in a test tool and environment, in which service agents can get accustomed to the system, in which bugs can be detected and in which patient experience can be measured. The organization is providing actual patient care. A mediocre implementation can come at the expense of patient quality perception. Special attention should be given to the interaction of the scheduling tool with service agents. In the simulation, an appointment from a group of five appointments is randomly chosen, but this might not include an appointment which suffices a patient's requirements. As a backup, the currently functioning system should still be available.

4. Setting the weights of the criteria based on the test environment. The simulation suggests that lowering the waiting time can come at the cost of distance in one experiment, and at the cost of increase variance in utilization in another experiment. The combination of weights should therefore be carefully tested in practice.

5. The scheduling method has a clear ranking in the options it provides. The best option is on top. We recommend to store the number corresponding to the chosen option to evaluate the number of appointments required to show. We suggest to start with a larger group of options and scale down over time. This also gives an option to investigate patient preferences.

6. The LightGBM method should be implemented as part of the scheduling method. The method has shown to be able to predict utilizations well, and to be able to reduce the variances in utilization. The scheduling method has direct influence on the quality of the prediction made by the LightGBM method. Therefore, the model should be retrained over time, also to include new practitioners and locations.

Future research - Our results shows that the LightGBM method performs best at the prediction of utilization. A proper implementation of the Topsis, using the LightGBM method also reduces the variance in utilization. The simulation suggests that - with the weights set accordingly - waiting time can be reduced. Due to the conflicting criteria, a careful and proper implementation in practice is required. Future research should focus on whether comparable results can be found, and the performance of both the LightGBM model and the Topsis method in practice. Interesting questions are how other type of models perform on the researched data and whether more data as well as other predictors will improve the predictions. For instance, the reason for cancelled or rescheduled appointments and how these can be included in the model. Finally a very relevant and important question is how the models perform in reality.

Glossary

Session A practitioners working day at one specific date and location

Work schedule Period of time during which sessions are defined, recurs one day every other week

MARS Multivariate adaptive regression spline is a form of regression that automatically models nonlinear relationships and interactions between variables.

LightGBM Light Gradient Boosting Machines is a gradient boosting framework for machine learning with focus on training performance and scalability

XGBoost Extreme Gradient Boosting is a gradient boosting framework for machine learning with focus on model performance

Cross validation Cross validation is a method to assess the performance of a prediction model, by leaving out a set of data to see how the model performs on unseen data

Utilization With utilization, unless mentioned otherwise, is meant the fraction of a practitioners session time that was filled with productive time

Productive time Productive time is time that was scheduled for either patients visiting the practitioner (patientbound appointment), or the practitioner performing administrative tasks (Non-patientbound Appointment)

Non-patientbound Appointment A Non-patientbound Appointment is a scheduled event during which a practitioner performs administrative tasks such as regional meetings, or updating the administration

Patientbound Appointment A Patientbound Appointment is a scheduled event during which a patient visits a practitioner

Access time Access time is defined as the number of days between making an appointment, and the appointment taking place. Can be seen as waiting time

Hyperparameter A parameter that controls the learning process. The parameter cannot be estimated from data

Break A break is a short interruption of the session, during which a practitioner can have lunch, or some coffee. A break is defined for the workschedule, hence is also recurring every other week

Interruption An interruption is an interruption of all work schedules and can span multiple days. This is used for for example pregnancy leave of absence, following training, but also for when a practitioner is ill

Fragmentation The degree to which a practitioner's schedule contains unscheduled breaks

Topsis Technique for Order of Preference by Similarity to Ideal Solution, multi-criteria decision analysis method that ranks appointments on a geometric distance scale, where the best solution has the shortest distance to the positive ideal solution and the longest distance to a negative ideal solution

TPE Tree-structured parzen estimator approaches build models to estimate hyperparameter performance based on past measurements in a sequential manner, the method then chooses new hyperparameters to test based on this past measurements

API Application programming interface. Allows computer software to talk to other computer software, by means of a defined protocol, generally known as the API.

Production environment The production environment is also known as live, particularly for servers, as it is the environment that users directly interact with, and has its own standard in terms of robustness, security needs and speed.

Contents

1 Introduction

The purpose of this master thesis is to advise an organization specialized in providing feet care, on the prediction of utilization of foot care practitioner working days that take place within one month (30 days). The prediction is an important input factor for the scheduling process, which is about to be altered after an intensive study and intrinsically depends on the prediction. With better predictions of utilization, the scheduling process should be able to better distribute patient-appointments to practitioner working days, and, by that the waiting time for new patients can decrease. For this thesis, we regard the scheduling process to be fixed. We will not alter the scheduling process. We primarily focus on the prediction of utilization. This chapter discusses the organizational context, the problem statement, the research goal, the research approach and the scope of the research.

1.1 Organizational context & research motivation

The organization delivers a wide range of foot care services to patients in the Netherlands. The organization is active at over 300 locations, mainly situated in, but not limited to, the eastern part of the Netherlands. The organization positions itself as a high quality, one-stop shop for all types of care in the lower parts of the human body, ranging from the feet, and the knees, to the hip and lower back. What's more, the organization specializes in care for Diabetic patients as well as professional athletes and has specific resources for children, however the majority are regular patients. A regular patient is a patient that has no specific type of specialization. The organization employs over 150 practitioners. The organization has a country-region-location structure, in which one of the practitioners per region serves as a regional leader. This however does not pose any limitation to patients with respect to where care is given. Patients are scheduled to practitioners based on distance to the location, waiting time, the type of specialization and the practitioner. This is done in consultation with the patient.

Ever since the expansion of the company towards more patients and practitioners, and the addition of a larger number of locations, the organization felt the need to assess the efficiency and effectiveness of their planning and scheduling process. Following a preliminary study towards the planning and scheduling process, the company was advised to incorporate a new methodology with respect to the assignment of patients to practitioners and locations in such a way that both their needs are represented in the process. However, the organization has articulated that the quality of the new methodology highly depends on the quality of the input factors required for the methodology.

Where the preliminary study (Westerink, 2021) primarily focused on finding, composing, and evaluating an algorithm that is able to weigh patients' needs together with organizational needs, the implementation of the algorithm depends on an impression of the degree of utilization across the locations and practitioners. That is, patients generally have two desires. The first being the desire to be served quickly, which is primarily due to the burden of carrying a physiological problem. The second being the desire to be served within proximity since that is less costly in terms of travel cost and less time consuming in terms of travel time. These two factors often compete when it comes to scheduling. In practice, there might not be an appointment slot available any time soon that is also in close proximity, but there might be one further away at another location. This is exactly the

problem that the new scheduling methodology proposed by Westerink, 2021 addresses. By putting emphasis on distributing patients evenly across locations, at the cost of patients having to travel longer, the global effect is that new patients can be served faster by the same number of practitioners due to the reduced variance in utilization. Another issue the scheduling methodology addresses is the fragmentation of the practitioner working days, by reducing gaps. With gaps in the practitioner working day, it is meant that the working day of a practitioner contains gaps between appointments. This is undesired, since it decreases the level of flexibility of the scheduling system in general. For example when an appointment is scheduled, it may cut the working day in two halves such that a longer appointment can no longer be scheduled in either of the gaps.

The organization is looking to enhance the results of the new scheduling methodology proposed by Westerink, 2021. It became apparent that one of the key factors for distributing patients more equally over locations is having a good perception of the utilization of locations and practitioners at a given date. From now on, we define the combination of a given day, practitioner and location as a session. That is the time between two demarcated starting times and finishing times the practitioner spends on a location on a given date. The prediction of session utilization takes place whenever an appointment is made. This is done in real time, by either a service agent or a practitioner. This also means that the prediction is required in real time. Since the prediction of utilization is situated within the scheduling process, we will describe the scheduling process, but this study focuses on the prediction of session utilization at the time that the appointment is made. This chapter describes the layout and research design and research approach.

1.1.1 Scheduling process

The scheduling process is the investigated input process. That is, the schedule built by the scheduling process results in an achieved utilization. The process is mainly twofold. Either the appointment is scheduled by the practitioner, or the appointment is scheduled by a service agent. New patients call the organization when they have a physiological problem that fits within the care types the company provides. The organization also receives referrals for patients from general practitioners or insurance companies after which the organization initiates a call to the patient.

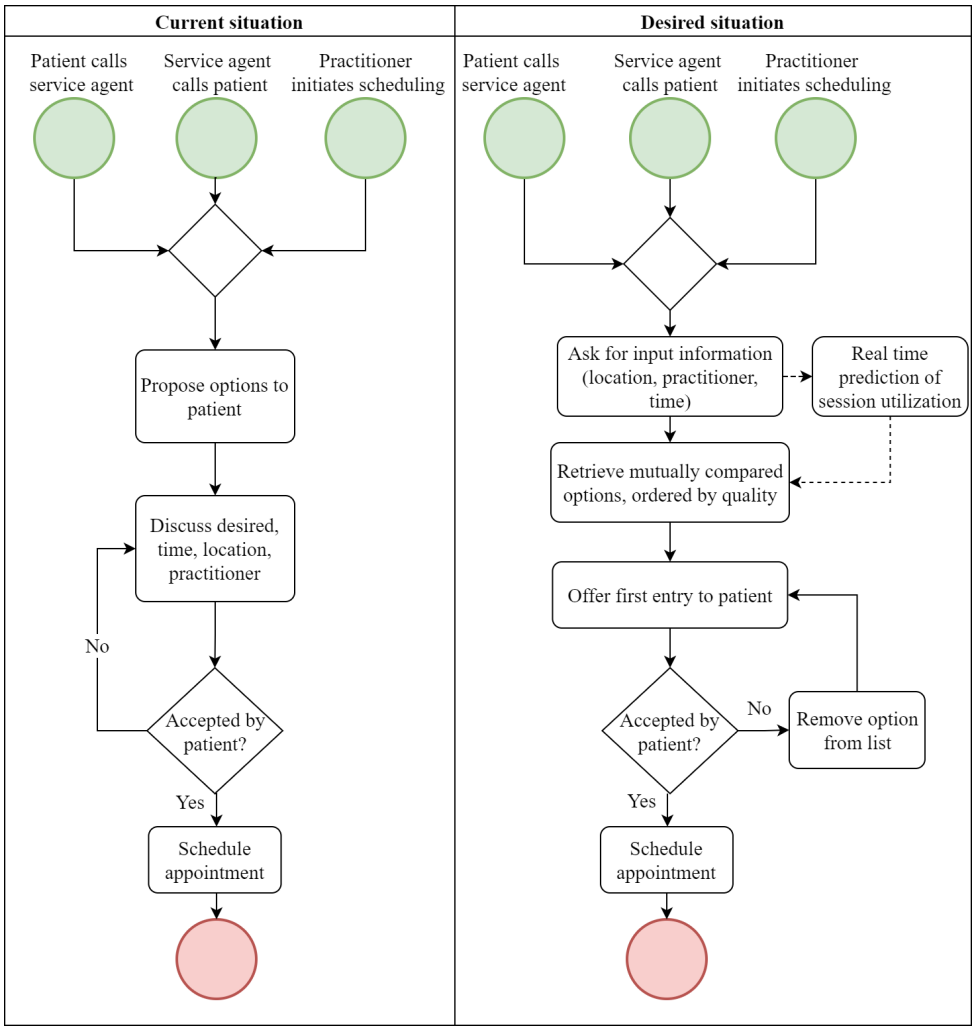

Figure 1.1: Scheduling process at the organization

Figure 1.1 depicts a simplified representation of the scheduling process as it is currently used, and how the organization desires the process to be. The agent and the patient then discuss the type of problem that the patient has. Using that information, the agent tries to determine what kind of care the patient needs. There are four major care types to consider. First of all, there is diabetes. Diabetic patients require specific care, and a careful examination by the practitioner. The next type is sports. This includes for instance patients that have pain during sport activities, such as cycling or running. This distinction is important as the type of care requires specialty equipment using which a practitioner is able to diagnose and treat the problem. This availability of the equipment is limited to one location. The organization also distinguishes between patients that are able to visit locations and patients that are not. In the latter case, patients are scheduled to a practitioner that visits patients at home. If none of the specialty cases are in order, then the patient is considered to be a regular patient and can be scheduled as well.

The next step in the scheduling process are appointments for recurring patients. These are patients that have a follow-up appointment after having received a type of care. For instance, to check the status of the problem after having received an orthosis or soles. These appointments are often scheduled during the preliminary appointment by the practitioner, or scheduled later. Generally, the patient then receives a phone call

from a service agent to schedule the appointment. In the desired situation, both the practitioner and the service agent use the new scheduling methodology to schedule the patient, but currently this is still a manual process.

Finally there are periodic patients. These are often diabetic patients that, due to the nature of the condition, need to have their feet checked in order to prevent more severe problems. Based on the severity of the condition, the organization decides how often the patient should be visited. Just as with the recurring patients, the patient is often scheduled at the end of the preceding appointment, but there are also general plans, for instance if many patients need to be visited on the same location.

In the desired situation, patients still discuss their condition with the service agent. Distance, but also a short access time are still valid concerns. What is new, is that when the agent retrieves the possible appointments, the list of appointments is ranked based on the input parameters of the patient the service agent enters to the system, but also the current state of the schedule is taken into account. The appointment with the highest rank is the appointment that scores the best on the interaction between travel distance, gaps in the session, the access time, the prediction of utilization of the session, and the type of appointment.

Currently, when scheduling appointments, the service agent finds the first appointment that fits the patient's needs. There is little to no degree to which the agent systematically distributes patients over schedules of different employees and locations, dates or times within dates. The agent has the ability to filter appointment options based on practitioner, distance, or the type of care the practitioner can offer and tends to opt for the closest option, disregarding the utilization or the current state of the schedule.

Practitioners only consider their own schedule. After the appointment, the practitioner schedules the next appointment if necessary. The practitioner may also call to reschedule should the practitioner not make the first appointment. The result is that the practitioner also offers the first possible appointment that fits the patients requirements. This is a problem, as this is a follow-up appointment. This does not contribute to the aim of the organization of lowering the access time of new patients, as it lacks coordination with the current state of the schedules. If the appointment chosen was the soonest appointment available in an area, or in general, then it increases the access time for new patients.

In the proposed scheduling process (section 1.1.2), the prediction of utilization is required when the appointment with the patient is made.

Westerink, 2021 describe that the scheduling problem can be formally defined as a $P|online-r_j|L_{max}$ problem (Graham et al., 1979), with m parallel servers in the form of practitioners and an online scheduling where jobs are released as soon as they are created. The objective is to minimize the maximum lateness, which Westerink, 2021 formally defined as $[max(accesstime)]^+$. However, it is noted that the main objective is obscured by conflicting objectives by for instance distance and utilization. For this problem, the processing times are assumed to be deterministic due to the standard processing times the organization gives to activities.

This paragraph illustrates that a problem resides in the current way of working with respect to the way both practitioners and service agents interact with the concept utilization in general.

1.1.2 Proposed scheduling system

Westerink, 2021 proposed a two-stage online scheduling system. The starting moment is the moment that either the patient wants to make an appointment, or the patient is called by a service agent to make an appointment, this is depicted by the desired situation in figure 1.1. The patient and the service agent have a short discussion regarding the complaints the patient experiences, after which the service agents tries to classify the problem into the appropriate category (Diabetics or Sport-specific complaints for instance, but note that in the general case, the service agent is by no means a medical professional). The service agent loads the scheduling tool, which is build around the Topsis multi criteria decision analysis (MCDA) method (section 3.1). The service agent asks the patient for some personal details such as the address, and patient preferences. Some patients request to be served as soon as possible, whereas other patients request to be helped as close to there address as possible. The service agent enters the details and preferences, and under the hood, the Topsis method returns a list of best appointments. Figure 1.3 shows the process visually.

This approach has two stages. In the first stage, the list of sessions which are regarded best is retrieved. The Topsis method uses the four criteria for this stage:

- Distance between patient and location.
- Deviation from target date.
- Utilization of the session corresponding to the appointment.
- The type of appointment that the patient requires and the type of the session (e.g. diabetics).

The sessions are ordered from best to worst, and the 30 best sessions enter the second stage. In this stage, the actual appointment is introduced. From this 30 sessions, all possible appointments are considered. These appointment enter the second Topsis method, but now the Topsis method also considers appointment-specific criteria. On top of the previously mentioned criteria, for which the values are carried forward to the second stage, two criteria are introduced:

- Fragmentation increase.
- Access time violation.

The appointments are again ordered from best to worst. The five best appointments are offered to the patient in order from best to worst. The patient picks the appointment that suits best, and the service agent schedules the appointment. The patient receives confirmation and the call is ended. This summarizes the scheduling process as proposed by Westerink, 2021.

We now discuss the criteria, and give the preferable direction. For the distance between the patient and the location, lower is better; patients generally do not want to travel too far, as this is costly in terms of money (e.g. car fuel) and (travel) time.

The (absolute) deviation of the target date is the number of days between the session date, and the target date. The target date is a factor considered only for recurrent and periodic patients. The target date is equal to the current date plus the desired access time, divided by the desired access time. The desired access time is determined by the practitioner.

Recall that the recurrent and periodic patient appointments are generally scheduled by the practitioner during the preceding appointment. Then, if the patient should be seen again in one year, the target date is today plus one year. A larger deviation from a desired target date is regarded worse. This helps ensuring that a patient is seen within the specified period of time, for instance once in six months.

We distinguish two types of the utilization-criteria. The first one is using the prediction method. This is the case when the appointment date is within 30 days. The second one is just the current level of utilization and is used when the target date is at least 31 days. The reason for this is that appointments scheduled within 30 days are regarded appointments for patients which want an appointment quickly. When the target date is at least 31 days, Westerink, 2021 assume that the important aspect to get an appointment close to the target date. Lowering the variance in utilization over sessions can be achieved by always picking the appointment with the lowest utilization, as scheduling an appointment to a session increases the utilization of the session. Therefore lower utilization is regarded as better.

Whether the type of the session is also the preferred type of the appointment is a qualifier. The value is 1 if the appointment type is, and 0 otherwise. So higher is better in this criterion.

The fragmentation increase has three possible values. The fragmentation increases by 1 if a new gap in the schedule is created. That is when an appointment is not scheduled at the start or end of a session, or directly before or after one appointment (adjacent to none of {appointment, the start of the session, the end of a session}). If the appointment is scheduled at the start, or just before the end of a session, or adjacent to at most 1 appointment, then the fragmentation increase is 0 (adjacent to at most one of {appointment, the start of the session, the end of a session}). If the appointment fills a gap between two appointments, or between the start of the session and another appointment, then the fragmentation increases by -1 (adjacent to exactly two of {appointment, the start of the session, the end of a session}). It should be obvious that a fragmentation increase of -1 is the most preferable option.

The access time violation is used for new patient appointments to see whether or not the access time is within the required access time of 21 days. This can be seen as a service level. The value is:

$$max(Access\ time - 21, 0)$$

This is the number of days after that the access time is larger than 21 days. For recurrent and periodic patient appointments, the value is always 0, as their appointments are mostly further ahead in time than one month, constituting to the assumption that for recurrent and periodic appointments, scheduling an appointment quickly is of less importance than scheduling an appointment around the specified access time.

1.2 Problem statement

The organization has executed an extensive study towards the scheduling process of patients. Since the algorithm that serves as the basis of the new scheduling method requires an impression of utilization, an exploratory study has been conducted. The exploratory

study first tried to predict the utilization of locations on an appointment date, but due to inaccurate results, the study altered the prediction method such that it was easier to predict. In other words, the organization made concessions on the outcome variable in order to have a more accurate prediction. In consultation with the executor of the study, it was mentioned that the main problem occurring from this is that the scheduling methodology and specifically the algorithm provides better results when it uses the prediction of utilization on a given location and date instead, provided that the quality of the prediction is good enough. This clearly indicates that the prediction of the utilization for locations and practitioners is a problem for the organization. The section problem definition is dedicated to identifying the problem globally. The problem is positioned in a broader context to see how it relates to other problems.

We use the managerial problem solving methodology, MPSM Heerkens et al., 2017 as the methodology for problem identification and solving. We choose the methodology for its generality and its applicability to a wide variety of problems in many situations of all sorts of expertise, and since we are familiar with the methodology. In the next section we draft an inventory of problems, we create a problem cluster, we define the core problem and express the problem in variables.

1.2.1 Problem definition

In consultation with the organization, and the executor of the preliminary study, the following problems are identified.

First of all, currently there is no prediction of utilization, or demand forecast in general. Patients are scheduled according to the available capacity. That also means that there currently is no measure of performance with respect to prediction. A few KPIs that serve as a measure for utilization have been considered. One of these is access time. Formally, access time is the time in days between the request for an appointment and the actual appointment. Within the organization the term is also used for the time between the current date and the third possible appointment time. The latter is defined as the expected access time. The aim of the organization is for 80% of the new patients to have an access time of at most 3 weeks. In 2019, this was 64.3%, with outliers to 10 weeks Westerink, 2021.

What's more, is that currently there is no structured use of expected utilization. The executor of the preliminary study has put much effort to be able to retrieve the achieved utilization at all. The database structure is such that data analysis is a challenging task. As a result, in the tested prediction methods, a small group of predictors is responsible for the predictions. This has two consequences. The quality of the prediction leaves to be desired, and therefore not the desired outcome variable is predicted, as said. Currently the outcome variable is the "Probability that the utility on a given date is larger than average utilization on a location" whereas the variable "Expected utilization in a given session" and arguably practitioner is a better input variable for the scheduling algorithm. Also note the aggregation from session to location in the currently assessed situation.

In the explored prediction method, the organization describes that there is evidence of "non-organic" interaction between the used predictors and utilization. According to the organization this is due to optimizations in the schedule. When a patient is scheduled by the service department, the patient might request the agent to note down that if an appointment slot comes up earlier, to give the patient a call to be scheduled earlier. As

a result, the organization orders the service agents to call patients to arrange quicker appointments. Due to these scheduled re-optimizations it is difficult to predict the utilization for a session when using the current state of the utilization in the session. A low utilization a few days in advance can result in very high utilization later due to re-optimization, whereas slightly higher utilization may not. In other words, this involves a nonlinear relationship. In consultation with, and after observation of the service department we learnt that re-optimization is a manual process with the time mostly being used to reschedule patients on locations that are heavily utilized. Another thing to note is that the scheduling system used by the service department has specific functionality for noting whether the patient wants to be scheduled earlier, with the preferred weekday as well. We learn that in practice, the service agents don't use this, but instead note this in a non-systematic way by just logging it. Service agents do so since it is considered to be faster and makes the task easier for the service agents.

Performance of the prediction system is also considered a condition for use of the system in practice. Predictions of utilization may only take a few seconds, as agents have to use the utilization estimate by means of the scheduling algorithm in real time.

Since service agents have no systematic way of scheduling and rely on their experience as a guideline for expected utilization at a location, the forecasting horizon implied by the agents is generally short and their forecast is underestimated. The latter follows from the inability to distribute patients across locations. The organization speaks out its desire to utilize a forecasting horizon that is at least a month, in order for the new scheduling methodology to better distribute patient appointments over locations and practitioners.

Another problem that the organization faces is difficulty with data processing and analysis. The database architecture was not designed having operational analyses in mind. Retrieving utilization is not simple, computationally demanding and utilization, or other KPIs are not stored in the database. This relates to the core of the problem to the extent that there may be important predictors we cannot use simply because we cannot identify or obtain the predictor. Also, with respect to performance, if the scheduling system cannot obtain the predictor in a timely manner, then the scheduling system is not usable by the service agents.

A few times per year, the marketing department sends out email batches to all patients that have allowance left with their insurance. Since the allowance generally resets at the end of the year, sending an email is considered a quality service to help patients remind them of the budget, for instance to fit new soles so that the patients always have recent soles. The idea is that in the days after the email batch, an increase in demand is observed, which leads to higher utilization. However there is no coordination between the marketing department and the scheduling department in terms of when to send the emails, and to whom. It is also not systematically registered when the emails are sent. Figure 1.2 depicts the relationship between the problems within the broader context.

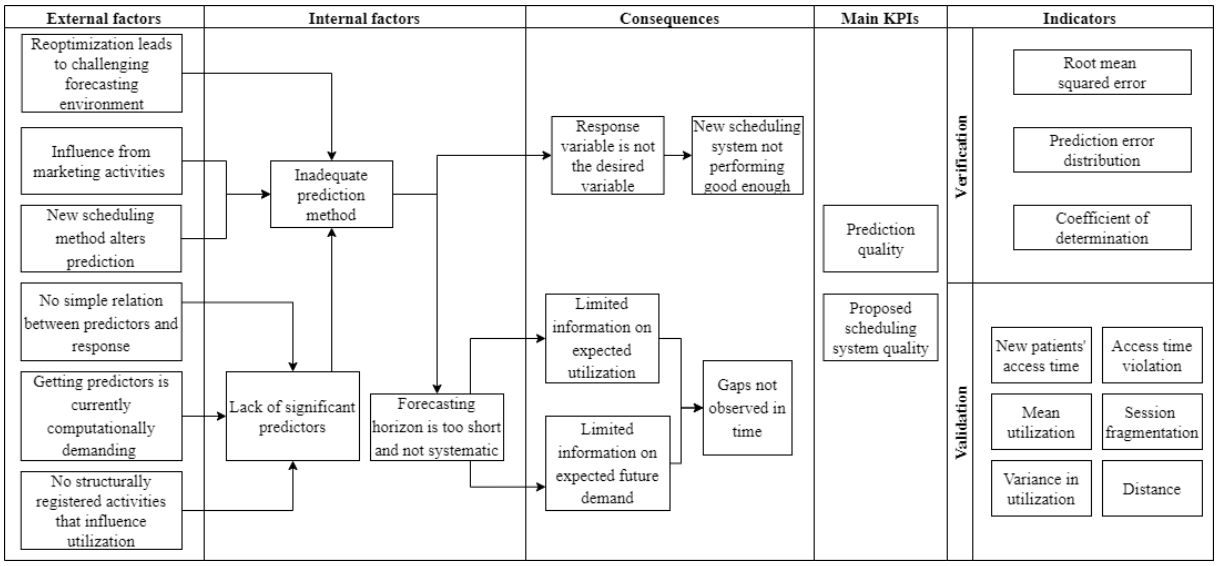

Figure 1.2: Problem cluster, depicts the relation between the occurring problems and the root cause

1.2.2 Core problem

Following the problem inventory we have several options to choose from as our core problem. We first discuss the alternatives that we consider to be external factors. Those are problems that we believe we cannot alter.

Marketing activities are essential for the company and out of scope of the assignment. We consider marketing activities to be external in the sense that we cannot alter this process and therefore the process. However, this does not limit us to include marketing activities within the prediction method.

Whilst we see that the new scheduling methodology will have an impact on the forecasts, we currently have no information on how the new methodology behaves, and we believe that the relationship between the prediction and the scheduling methodology should be the other way around. Scheduling should be done on the basis of the prediction and not alter the prediction based on the way the scheduling algorithm schedules appointments.

That there is no simple relationship between predictors and the response is also something we cannot alter. We could either draw up different predictors, or use a prediction model that is able to model the relationship. We learnt at the service department that re-optimization often only happens at heavily utilized locations, and from time to time depends on the utilization of the service department itself, as the main priority is answering inbound calls, in case of which there is no time for re-optimization.

Since influential activities are not structurally registered, we could either measure the activities, or let the activities out. We choose to leave them out as we otherwise would not be able to use the data preceding the measurements. This is at least the case for marketing activities and the re-optimization.

Not being able to obtain predictors due to that these are computationally demanding means we cannot use them in the model. We should then find other predictors, whereas using a computationally demanding predictor also does not guarantee a better result. That leaves the alternative core problems:

1. Forecasting horizon is too short and not systematic.
2. An inadequate prediction method is used.
3. Lack of significant predictors.

Following the MPSM (Heerkens et al., 2017), we should choose the root cause and the problem that has the most influence on the final result. Therefore we do not choose alternative 1.

That leads us to problems 2 and 3 as options for the core problem. These are closely related. Prediction methods will become better when using predictors that are better at explaining the fluctuation. However, if the predictors are inherently good, but the prediction model is unable to adequately model the relationship between the predictors and the response, then the predictors are of no use. Hence we choose the two as the core of the problem which we formulate as "the current prediction system quality is too low". Both the prediction model as the degree to which the predictors are adequate, influence the prediction quality. In other words, prediction quality is the variable. In the end, the problem leads to higher access times for new patients. Westerink, 2021 showed that the access time for new patients can be decreased to the average level of 14.8 days. This serves as the goal for the study.

Since predictive quality can be defined in multiple ways, we use the following indicators to assess the quality of the prediction. First of all, we verify that the prediction model predicts utilizations well. We use the root mean squared error as a measure of the difference between predicted and observed utilization. Secondly we use the coefficient of determination, as a measure of the amount of variance that can be explained by the model predictors. Finally, the degree to which the model is able to accurately predict different levels of utilization is important, as this is exactly what enables the scheduling system to determine the alternative appointment options. This is reflected in the indicator "prediction error distribution".

Since the prediction model serves as input for the scheduling method as described in section 1.1.2, we acknowledge that successfully developing a prediction method in terms of low errors does not directly show that the scheduling system works better. Therefore, we use a second variable regarding the proposed scheduling system as well. Following Westerink, 2021, the aim of the scheduling system is to minimize the maximum lateness per session. However, this is not acceptable if other criteria are neglected. Therefore a good prediction method influences the Westerink, 2021 proposed scheduling method such that these criteria are considered:

- New patients' access time - 20.7 days
- Mean & variance in session utilization - 0.917 & 0.00917
- Distance from patient to location - 3.45 km
- Access time violation for new patient - 37.1%
- Session fragmentation - 1.4

We analyze the performance on the mean and variance in session utilization as well as access time ourselves, as we know this directly influences the scheduling method, and for the other indicators we use the values from Westerink, 2021, since this is more related to

the scheduling method, which we do not alter. The values listed here are the values from 2019.

1.3 Research goal

This section is devoted to the research design and approach. The research goal is and research questions are formulated. The problem is also operationally defined.

1.3.1 Main research goal

The organization aspires to have a lower access time for new patients by developing an algorithm that is able to distribute appointments to lower the variance in utilization. The goal of this research is to develop a prediction model that is able to adequately predict the utilization of sessions.

The research goal is translated to the main research question:

How should the prediction model be composed in order to minimize the difference between the observed session utilization and the predicted session utilization?

The aim of this study is to find an answer to this question using the perspectives:

- An analysis of the currently proposed prediction method.
- A review of literature towards a methodology that contributes to the prediction model development
- Data analysis of the provided data in order to find predictors that are able to describe.
- A review of the existing literature to find models that are able to predict utilization using multiple predictors.
- The development and assessment of a predictive model that minimizes the difference between observed and predicted session utilization.

Following (Heerkens et al., 2017), we transform the core problem into an action problem. That is, we formulate the core problem such that it forms a discrepancy between a norm and a reality:

How can we develop a prediction model that reduces the difference between observed and predicted utilization, in order to influence the proposed scheduling method such that the access time for new patients is lower than 14.8 days?

1.3.2 Research questions

We draw up research questions that should contribute to answering the main research question. The research sub questions such that each chapter answers a research question. The approach to solving the questions is explained in section 1.4.

1. Research question 1: What is the current performance of the scheduling system and the proposed prediction method? Chapter 2
   (a) What is the current access time?

(b) What is the current variance in utilization?

(c) How does the proposed prediction method perform?

2. Research question 2: Which methods are available in literature regarding the prediction of utilization? Chapter 3

   (a) Which methodologies for implementing prediction models are present in literature?

   (b) Which methods for predicting utilization are present in literature and are suitable?

3. Research question 3: How can we develop a prediction model for session utilization that produces accurate results? Chapter 4

   (a) How does the proposed prediction model perform?

   (b) What are the most important predictors of the model in predicting utilization?

4. Research question 4: How would such a prediction model perform in relation to the by Westerink, 2021 proposed scheduling method? Chapter 5

   (a) How can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the variance in utilization?

   (b) To what degree can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the waiting time for new patient appointments?

1.4 Research approach

Partly described in the previous section, the study continues in the following structure. The next section defines the scope of the research. Chapter 2 continues with the current performance of the process as it currently is, using data-analysis. Chapter 3 proceeds with a literature review that is aimed at finding appropriate prediction models as well as how to find predictors in a structured manner. Chapter 4 describes how the model should be adapted and used, and what predictors are used. Chapter 5 assesses the performance of the model in a more practical setting. Chapter 6 contains a discussion of the results, describes the scientific contribution, research limitations, recommendations for the organization, recommendations for future research, and the generalizability of the study.

1.4.1 Research design

Chapter 2 starts with an analysis of the current situation. Chapter 3 continues with an literature review, in which we not only provide the theoretical foundation for prediction methods, but also elaborate on the Topsis multiple multi-criteria decision analysis (MCDA) method, which forms the basis of the proposed (online) scheduling support system. The theory on prediction models and the resulting models are tested and cross validated in chapter 4. Figure 1.3 depicts the relationship between the chapters 3, 4 and 5 and shows the chapters' relationship to the scheduling method. As seen in figure 1.3, the prediction of utilization is input to the by Westerink, 2021 proposed implementation of the Topsis scheduling method. We validate the scheduling method in combination

with the prediction method, which we developed in chapter 4, by means of simulation in chapter 5.

The first step is the analysis of the current processes. This is done using data analysis. First, the database is explored to know where the relevant data is. From the relevant data, a usable data set will be created. This serves as the starting point for the analysis. Using the data RQ1abc are answered.

We use literature to answer RQ2abc. RQ2a should help with the remainder of the study, as we are currently only familiar with more general methodologies. Our preliminary searches and perceptions reveal two options which we investigate:

1. Find literature that is related to the environment and process.
2. Find literature that is generally applicable to the kind of outcome variable we aim to model.

From RQ2abc we learn good applicable prediction methods. In chapter 4 we develop a prediction model and use cross validation to tune the prediction model. We also describe how we construct predictors and find the most important predictors of the model in predicting utilization.

We answer research questions RQ4ab using simulation in chapter 5. We do this to validate the model to ensure that the model influences the real world problem. Since the goal of the prediction model is to influence the scheduling system in such a way that eventually there is a lower access time for new patients and a better distribution of appointments over locations and practitioners, we use the indicators access time and variance in utilization as validation indicators to ensure the quality of the prediction with respect to the real world. In the preliminary study, a simulation model was built to assess the performance of the proposed scheduling method with respect to the indicators used there. We can implement the method we develop in that simulation model to assess the performance of the model with respect to the access time and the variance in utilization.

Since we are interested in the increase in predictive performance, and how this relates to the scheduling method, we compare both the method introduced in the preliminary study (logistic regression) with the model we aim to construct. A practical issue resides in that the preliminary study uses a logistic regression, so we cannot directly compare the result with our model. We identify three options:

1. Use the predictors out of the assessed prediction method in the proposed new method and compare the performance.
2. Use both methods in the existing simulation model and compare the outcome between the two.
3. Relate the probabilities resulting from the logistic regression model to the regression model.

Options 1 and 2 are a pragmatic option to compare the performance and are in line with the aim of the study in general, hence we choose to assess and compare the performance of both models in this way.

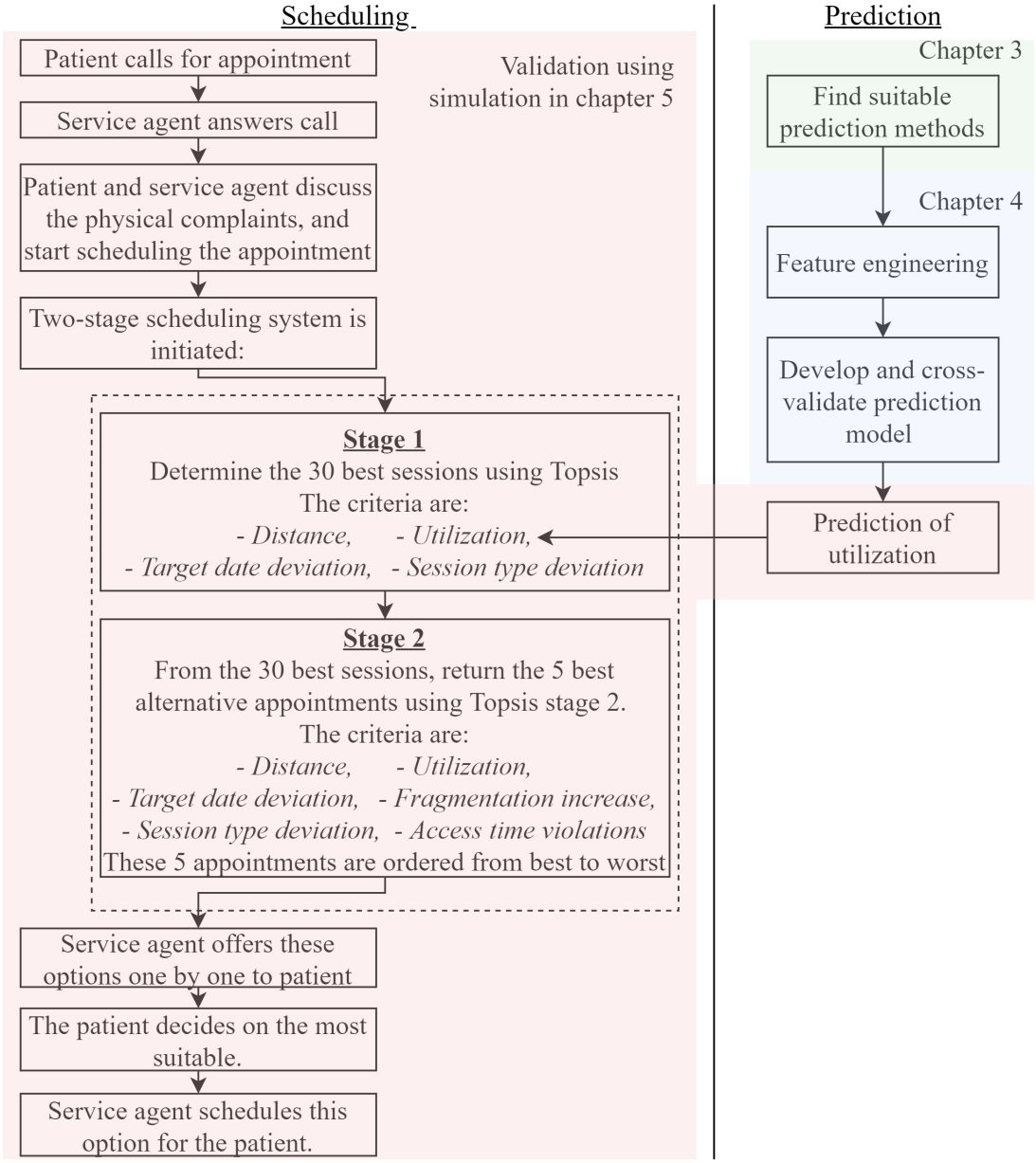

Figure 1.3: Research design with the activities and relationship between the chapters 3, 4 and 5

The final steps are to draw conclusions from the assessed models in practice, to describe how this can be used in practice in the form of an implementation plan, what we recommend the organization to do with the results, and why it is relevant to literature. This is part of chapter 6.

1.4.2 Scope

The scope of the research is limited to prediction modelling and finding predictors. Only data available in the database is used. Data will not be collected since the modelling is aimed at a larger scale application and an automated deployment where data availability is really important. This rules out the usage of marketing activities for instance. It does however include observation of processes from which predictors can be learned. With

respect to the data, data after 2019 is not considered, as the covid-19 pandemic disrupted the usual business processes and is therefore not considered to be representative. This is acknowledged by the organization and Westerink, 2021.

The predictions will be on session level as observed at the time of an appointment, since that is also the considered input variable for the scheduling algorithm.

The forecasting horizon will be one month (30 days) long, since almost all appointments that were not planned for future appointments (recurring, periodic) were made within 1 month ahead in time.

As scheduling is a large share of the preliminary study by Westerink, 2021, scheduling is mentioned a lot. Since the study was devoted to scheduling, we will not further develop scheduling in this study, but use the findings presented by Westerink, 2021. But we will investigate the effect of the prediction method on the scheduling system, and therefore allow subtle modifications of the scheduling method in the form of weight changes, which alter the relative importance of the scheduling criteria. 1.3 reflects this. The left part (scheduling) is not altered except for the change in weights. The right part (prediction) is the part that we develop and alter.

2 Problem Context

In this chapter we review the current performance of the organization with respect to planning & scheduling, and forecasting. The aim of the chapter is two answer research question 1: What is the current performance of the scheduling system and the assessed prediction method? We have subdivided the question into three separate subquestions, which together answer research question 1. The sub research questions relate to the indicators we have set to measure the variable of the core problem. The chapter starts with an introduction of some concepts that are incorporated by the organization in section 2.1. This should contribute to the understanding of the design choices. Section 2.2 is dedicated to the current access time, section 2.3 measures the utilization and its variance, section 2.4 assesses the earlier proposed prediction method. Finally, section 2.5 concludes the chapter.

2.1 Session

For patients to have an appointment scheduled, the session serves as a directive. A short recap of the general process is when a patient requests an appointment. Then the service agent offers the patient an appointment slot from the available time in the session. Recall that the session is defined as the time between a start datetime and an end datetime.

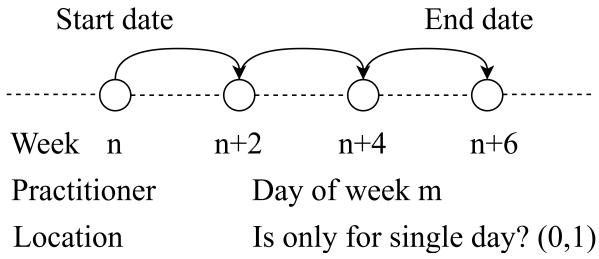

Figure 2.1: An illustration of how sessions relate to the work schedule

The session is constructed from a work schedule by its defining properties. Every distinguishable work schedule has a range of dates in which it is active, an even or odd week number on which it is active, a day of the week on which it is active, a location, a practitioner and a flag whether the work schedule is only active for a single date, or not (figure 2.1). The flag carries the value 1 when the session is marked as being an exceptional day, or 0 otherwise and then obviously the start date equals the end date. We refer to the virtual concept that is a single date on which a work schedule is active as the session.

The session has a time range during which a practitioner is available for patients. This serves as a soft constraint. From time to time, patients are treated outside of the session. This generally requires approval of the practitioner and is therefore often scheduled by the practitioner, or after approval of the practitioner. Only a small fraction of patients are served outside of the session. We do not consider this to be productive time since it is time that was not scheduled. Occasionally, there are appointments that are partly within the session and partly outside of the session. In such events we only consider the time overlapping with the session to be productive time. Employees can adhere to multiple sessions on a single day, but only one at the same time.

During sessions, breaks occur. The intended use of the break is for the practitioner to have some coffee or to have lunch. The break has a start time and an end time, but no date. The break is bound to the work schedule, so holds for all sessions that belong to the work schedule. Breaks are scheduled, and therefore we do not consider the break time to be part of the session. This is accounted for by subtracting the break length from the session length. However, breaks may completely or partly fall outside of the session, or appointments may overlap with the break. In the first case, the time that either falls outside of the session is not subtracted from the session. The time that overlaps the appointment is subtracted. In this case, the appointment time does not count towards productivity.

Then there are interruptions. The purpose of the interruption is to interrupt an employee's schedule for an extended period of time. For instance pregnancy leaves, holidays or attending training are interruptions. The interruption is bound to an employee instead of a work schedule, but stored together with the break. Unlike the break, interruptions have a start and end date to accommodate the extended period of time. Since the interruption is bound to the employee instead of the work schedule, it is able to span multiple sessions. The start time is only bound to the start date and the end time is bound to the end date. The result is that an interruption stretches all days in between the start date and end date. In hierarchy the interruption is generally higher than the break. Therefore the interruption can overlap breaks or appointments. If the interruption overlaps with a break, then the overlapping time is subtracted from the session length if the time also overlaps with the session. If the interruption overlaps the session, then the time is just subtracted from the session length. If an appointment is scheduled on top of the interruption, then the time is considered to be non productive time and therefore the time is not counted. The overlapping time is subtracted from the total productive time.

Finally there are non-patient appointments. These are appointments that are stored with the patient appointments, in which activities are performed that contribute to the company's services. Examples are quarterly management meetings or a non-patient appointment to bring the administration up to date. Remarkably, we also see that these are used for keeping track of tasks not contributing to the company's services such as asking approval for leave of absence, or notes regarding patient appointments, like when a pair of soles should be ready for delivery so we note that these are not always meaningful. We filter out those appointments that are intended to be used as reminders. In consultation with the organization we use the non-patient appointments as productive hours since they can be seen as scheduled appointments. We introduce the requirement that for a session to be regarded as a valid session it needs to contain at least one patient-appointment. The non-patient appointments show overlap with interruptions, breaks and the session as well. Hence, these are accounted for as well. Since the non-patient appointments are productive, the appointment time is not subtracted from the session length. When overlapping an interruption or break, the overlap time is subtracted from the total productive time for the amount of time that the appointment overlaps the session. This is done in this fashion in order to ensure that productive time during breaks or interruptions does not contribute to the utilization.

It has come to our attention that there are many exceptions to the kind of activities we discussed. Mapping the datetime ranges to the corresponding sessions in meaningful fashion is not straightforward in this context. Appendix A describes the extraction of useful sessions in depth.

2.1.1 Measuring utilization

From the constructed metadata we determine utilization by combining the individual components discussed in section 2.1 and appendix A. The utilization of the session is then determined by the components productive time and session length. This is defined as total session length without interruption time and break time, which are therefore subtracted from the total time. From this period, the total time that is used for appointments, either patient-bound or non patient-bound is defined as the productive time. The ratio of productive time over the session length without interruptions and breaks is defined as the utilization, more formally:

$$Session\,utilization = \frac{Productive\,time}{Session\,length}$$

Since the appointments, breaks and interruptions that lead to productive time and session length may or may not show overlap, the actual approach is described in appendix C. Using this procedure leads to utilizations with the distribution as in figure 2.2.

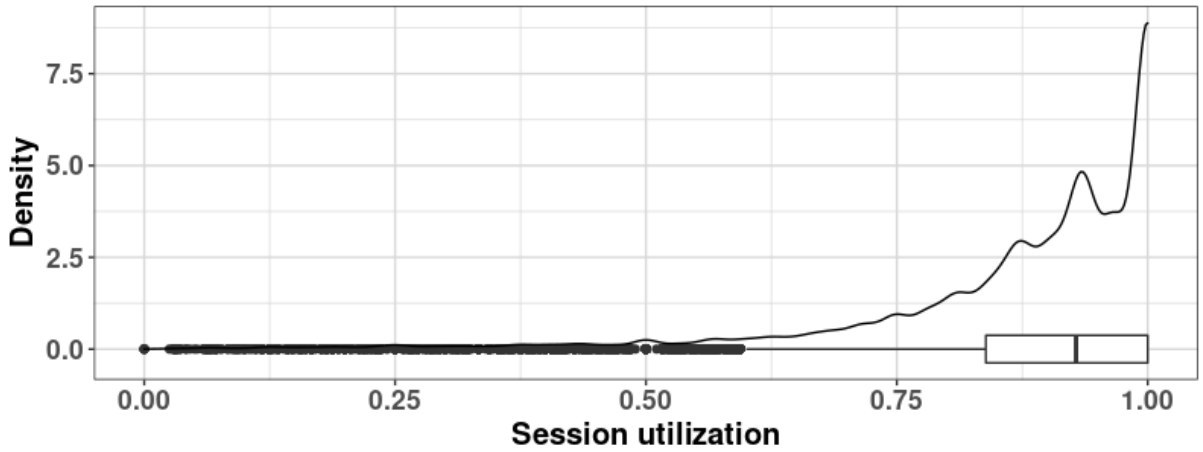

Figure 2.2: Density and distribution plot of utilization, (2013-10-02 until 2021-05-01, n=50486)

The figure shows that the procedure works well as all but 299 utilizations are in the unit interval. The excluded 299 sessions have a session length of 0, which means that the sessions would be excluded anyway. This happens when planners (i.e. the employees that manage the schedules) implicitly close a schedule by setting the start and end time to 00:00 hours. The local minimum between the median and 1.00 is due to that appointments have predefined lengths for the appointments and since sessions often are of the same length. Most appointments are 30 minutes long and the most occurring session length is 8 hours. If one appointment block has been left open, or if an appointment was cancelled, that decreases the utilization from 1 to 0.9375. We are well aware of that a utilization between 0.9375 and 1 is an options due to break lengths, non-patient appointments which have nonstandard lengths, or shorter appointments, but it in the context of the current paragraph it explains the gap.

2.2 Access time

Recall that we defined access time as the number of days between making an appointment and the actual appointment. The variable is an indicator for utilization in general. If the access time is low, then the number of free appointment slots must be relatively high, whereas if the access time is high, then there are many patients requesting an appointment. We analyse the access time in two manners. First of all, there is the appointment access time. This is the access time for individual appointments. This provides information about the process in general. Then we aggregate over appointments, and consider the average access time for sessions. We also compare the difference between the access time for new patients and for all patients.

2.2.1 Appointment access time

The appointment access time is a metric for individual appointments. The metric originates from the logging of the appointment and the time the actual appointment takes place. Sometimes appointments are scheduled in hindsight in order to bring the administration up to date. These appointments are not included. Sometimes, appointments are scheduled for multiple years ahead. Those are also not considered, as this is a very small number of appointments (1%) and are scheduled anyway.

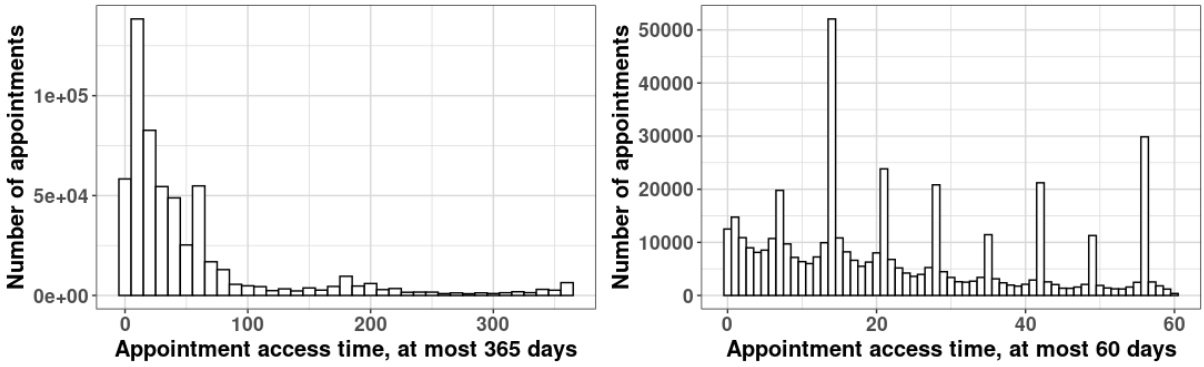

Figure 2.3: Histogram of appointment access times for all types of appointments, (2013-10-02 until 2021-05-01, n=(579952, 445531) bin-widths = (10, 1))

Figure 2.3 shows the distribution of appointment access time over the appointments. For all appointments, the largest part of appointments is scheduled at most 100 days ahead. If we zoom in to 1 day ahead, we see an interesting pattern occurring. There are surprising spikes that are an equal number of days apart. Disregarding the today or tomorrow case, we see that the spikes are seven days apart. A reason for this is that the largest part of these appointments are scheduled by the practitioner a recurring appointments. When practitioners schedule appointments, they generally ask the patient whether the patient is available exactly one week week, two weeks, or even more weeks from now. It appears that this is quite literal and due to that many practitioners often switch locations per day, then recur to the same location one or two weeks later. This is due to how the sessions are constructed from the work schedule (section 2.1). We see a clear distribution of appointments around the seven days ahead. If we disregard the peaks, we see a decreasing, somewhat exponential decrease of the number of appointments. We are under the impression that there are two processes influencing the access time. We elaborate on this.

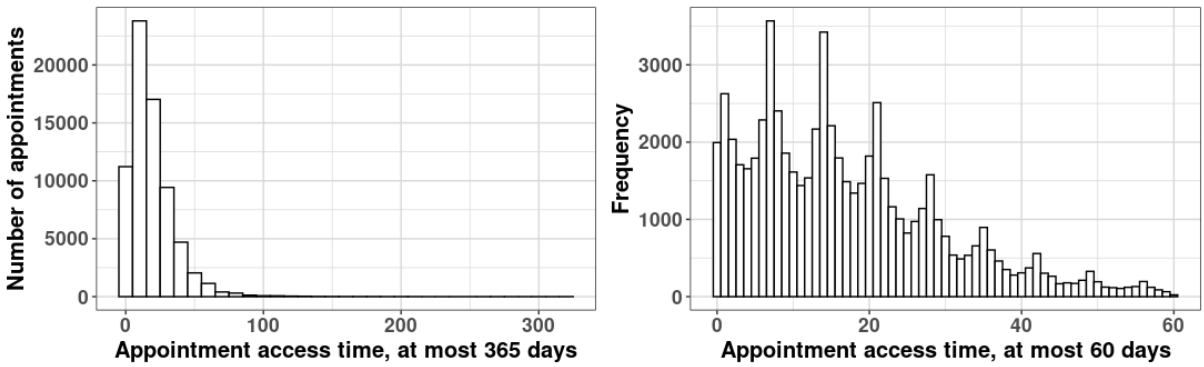

Figure 2.4: Histogram of appointment access time for new patient appointments, (2013-10-02 until 2021-05-01, n=(70498, 63672), bin-widths = (10, 1))

Figure 2.4 is similar to Figure 2.3, but now filtered to only include new patient appointments. Whether the appointment is a new patient appointment is an influential factor for the access time. New patients often simply require a quick appointment. If the patient calls, the first suitable appointment is selected. Practitioners generally schedule the follow up appointment. Thus, the distributions of access time for patients that are scheduled for follow up appointments should follow the seven, fourteen days ahead pattern to a higher degree. This is also reflected by figures 2.3 and 2.4 where we see that the new patients are more often scheduled as soon as possible. Yet, we still see the pattern. We obtain two insights. In consultation with the service agents we find that patients often call during a day off from work. Patients often have the same day off from work each week, which explains why the peaks are seven days ahead. The other insight is that when the patient calls, an appointment can always be made seven days ahead. This is not true for other number of days ahead. For example, if the patient calls on Friday, an appointment could be made for the succeeding Friday (7 days later), but not for the succeeding Sunday (2 days later).

Scheduling ahead options								
Day of the week / Days ahead	0	1	2	3	4	5	6	7
Monday	0	0	0	0	0	0	1	0
Tuesday	0	0	0	0	0	1	0	0
Wednesday	0	0	0	0	1	0	0	0
Thursday	0	0	0	1	0	0	0	0
Friday	0	0	1	0	0	0	0	0

Table 2.1: Options for scheduling ahead when calling on a day of the week. 0 is an option, 1 is not an option

In table 2.1, the value 1 mean that scheduling j days ahead when calling on day i is not possible, 0 otherwise. Scheduling on the same day is always possible provided there is a session with space available. The same is true for 7 days ahead. For the days in between it is not except for 1 day ahead. That is since on Saturdays no calls are taken. This is one of the reasons for observing the peaks every 7 days. The other is that the number of calls is generally much larger on Mondays than on Fridays (figure 2.5). This is reflected in that the pattern recurring at 2, 9, 16, ... days ahead has a smaller peak.

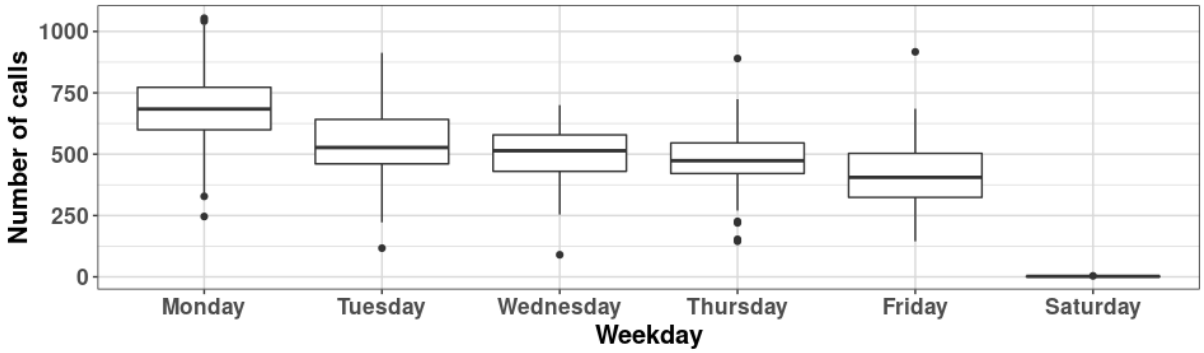

Figure 2.5: Distribution of incoming phone calls over weekdays , (2019-05-15 until 2021-09-10, n=308657)

One of the reasons for a larger number of calls on Mondays is due to that the topic of conversation is feet problems. Often, patients hike during weekends, do sports or other physical activities which either reveal or is the source of the physiological problem. Then the patients call on the Monday after which is reflected in figure 2.5.

Access times per year			
Year	μ	σ	% $\leq$ 21 days
2014	15.3	12.3	78.5%
2015	21.6	14.6	57.6%
2016	22.4	18.2	59.9%
2017	19.2	16.0	61.7%
2018	18.8	15.1	64.4%
2019	20.5	16.9	59.0%
2020	18.2	17.3	66.9%
2021	16.1	15.4	73.4%

Table 2.2: Mean access times and standard deviation in days per year (2013-10-02 until 2021-05-01, n=70832)

Finally, as an answer to RQ1a we find that the mean appointment access time over the last years was 18.8 days and the standard deviation was 16.5 days. Since 2021 is the running year and 2020 was the covid-19 affected year, 2019 is the most representative recent year and per 2.2 had a mean appointment access time of 20.5 days with a standard deviation of 16.9 and 59.0% of the new patients' appointments was scheduled within 21 days. This is slightly different than the 62.9% that Westerink, 2021 found in the same source. This is easily explainable as we focus on the appointment types that require prediction, where Westerink, 2021 also focus on different type of schedules. Also, we have a slightly different approach to transforming the data to usable values. While this might affect the eventual outcome, we believe the insights and patterns remain the same so we do not consider this an issue.

2.2.2 Session access time

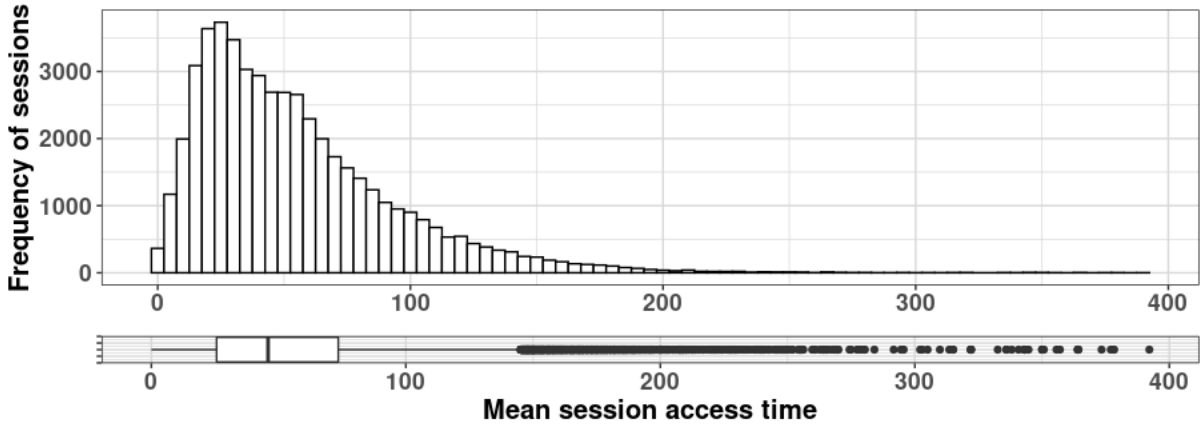

Figure 2.6: Histogram and corresponding boxplot of mean session access time (2013-10-02 until 2021-05-01, n=50486, bin-width = 5)

Session access time is an aggregation of average access times during sessions. It is a combination of multiple appointment access times therefore the seven day pattern is mitigated. We see that across sessions the access time is mostly within the range of 22 and 70 days, with a mean of 55 days. The access time is larger than the aimed 3 weeks of access time for new patients due to the number of recurring appointments which are often scheduled further ahead.

2.3 Utilization

We conduct data analysis to obtain a general impression of utilization at the organization and start with an aggregate on the highest level. We aim to find some general, higher level insights in utilization such as its distribution over time, but also more in-depth insights such as its distribution over locations or employees.

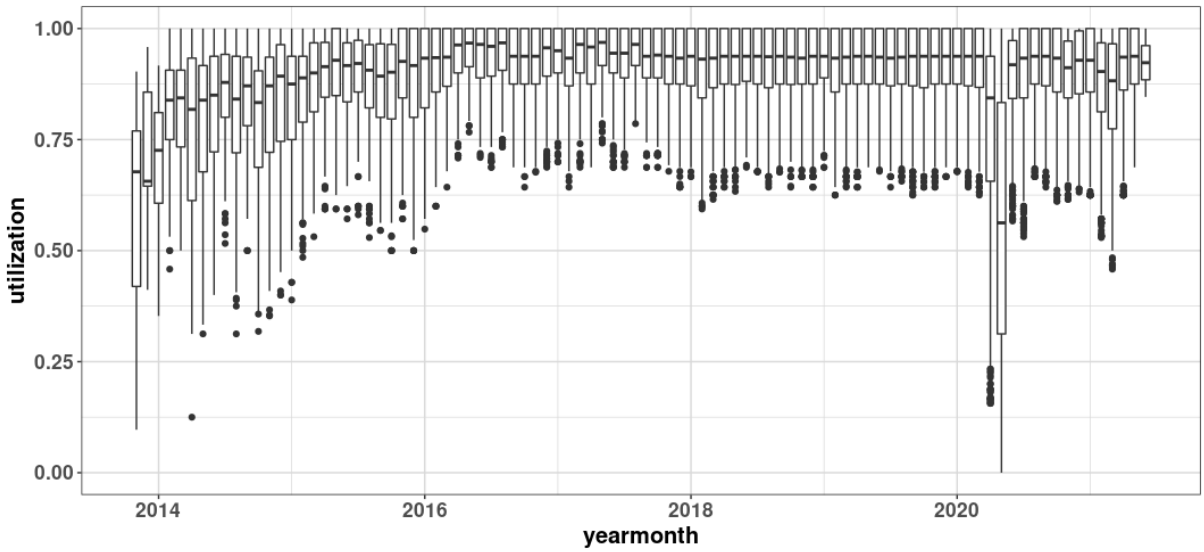

Figure 2.7: Utilization distribution per year month (2013-10-02 until 2021-05-01, n=47767, outlier removed per year month)

In figure 2.7 we see the distribution of session utilization per month of the year since October 2013. Outliers are removed to make the general pattern more easy to see. The first thing that stands out is the decrease in average utilization and the increase in variance just after the first months of 2020. This is due to covid-19 measures. The second thing that stands out is how the utilizations of the first months in the data set are increasing towards a more stable and higher average variance, until November 2015. There can be many reasons for this and the most practical solution is to limit the usage of data towards the interval starting in November 2015 and up to February 2020. In that interval the utilization is generally stable.

With respect to the trend for the period until the corona crisis we conclude that the utilization is not increasing nor is it certainly decreasing. Whilst the company has experienced major growth during the period, it is important to realize that the results are utilizations. In other words, if both demand and capacity follow the same increasing trend, then utilization is expected to follow this trend as well, under the assumption that patients tend to follow the same characteristics with respect to for instance showing up. We should be able to observe a difference in access time since access time is a form of waiting time. In general, waiting time increases with utilization.

Utilization itself is a fraction containing the available work time and productive time. Inherently, utilization is a real number on the unit interval. In this specific case only the numerator is stochastic. That is, the denominator is deterministic, as it is known beforehand on the prediction horizon we consider. The productive time is a summation of the appointment lengths. We decompose the total productive time in order to obtain understanding of the process.

2.3.1 Productive time

Productive time and the session length are the defining factors for utilization. Productive time can never exceed the session length is non negative. This results in a value on the unit interval. Differences between two corresponding session lengths and productive times lead to dis-utilization, as the difference can only be time that was not productive. The organization has articulated the growth of the number of sessions, practitioners and number of visited patients. We should be able to see this in the organization's productive hours, but also in the session lengths.

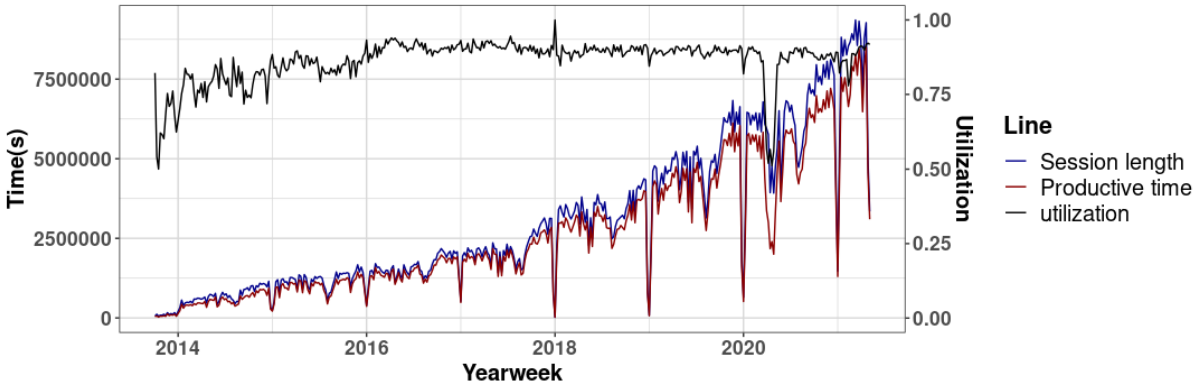

Figure 2.8: Sum productive time, session length and corresponding utilization per yearweek (2013-10-02 until 2021-05-01, n=50486)

In figure 2.8 we see the weekly sum of the appointment seconds per session for all sessions containing patients during the past years. There is a clear increase in the sum of productive time per week. We see a decrease in productive time during July and August which is due to summer vacations, and a decrease in December, which is due to the winter holidays. The closer the productive time and session lengths are, the higher the utilization. We see a clear drop in productive time as well as session lengths during the Christmas period. This is not reflected by the utilization. In other words, productive time tends to follow the session length but utilization is a process on itself.

As discussed, sources of dis-utilization are often unexpected disruptions in the schedule like illness. It should be clear that disruptions have a larger effect on days with a smaller number of sessions, as interruptions are relatively a larger part of the schedule. We know that Saturdays are almost always comprised of way fewer session hours. We also know that there are differences between other days of the week. We learn from the service agents that this is due to the larger number of callers on Mondays.

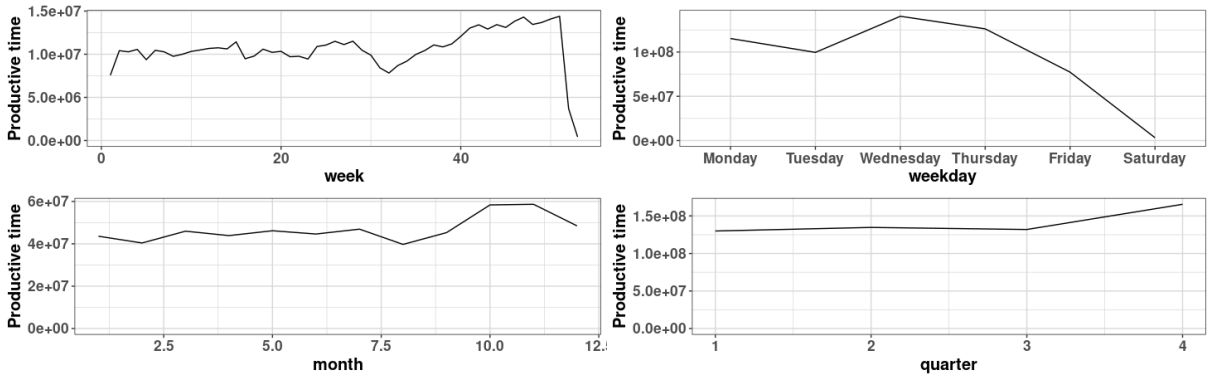

Figure 2.9: Practice time over time periods (2016-01-01 until 2019-12-31, n=26707)

Figure 2.9 shows the seconds of productive time per week, per weekday, per month and per quarter respectively. The week overview clearly shows the effects of the summer holiday period and the Christmas period. Also, The number of productive hours on Saturdays is small when compared to other days of the week. The month overview highlights how the number of productive hours fluctuates over the year, with a small drop around the summer holiday months. The advantage of using months over weeks is that the week number might or might not include a specific date depending on the year, whereas months always contain the same dates. The advantage of using weeks is that large fluctuations within a month are not observed. An example of this is The month December, which has a larger number of productive hours when compared to, for instance, August, but also has the Christmas week, which generally has a large drop in productive hours.

We have to realize that the productive hours tend to follow the number of session hours that are available on the day. As explained, this is seen in both (a larger) access time as well as the relationship observed in figure 2.8. This is important, as this means that are largely influenced by the session length.

2.3.2 Variability in utilization

Variability in utilization is the reason that prediction is necessary in general. Obviously, if there was no variability in utilization then utilization would be deterministic and a

constant was to be used. There can be so many sources of variability. We name some of them.

- Differences in patient demand.
- Differences in capacity.
- Regional differences.
- Location differences.
- Marketing activities leading to growth in patient demand.
- Periodic difference (Quarter, Month, Week, Weekday).

In the end, utilization is always the ratio productive time over session time. On the long term we expect productive time to influence the number of session hours, as high access time means that patients are waiting for an appointment. More practitioners helps with reducing waiting time. On the other hand, there is no more productive time than the session length. Therefore, if the access time is high, we expect higher utilizations as this indicates that schedules are filled to a large degree. We are able to compare session lengths with practice times by means of utilization.

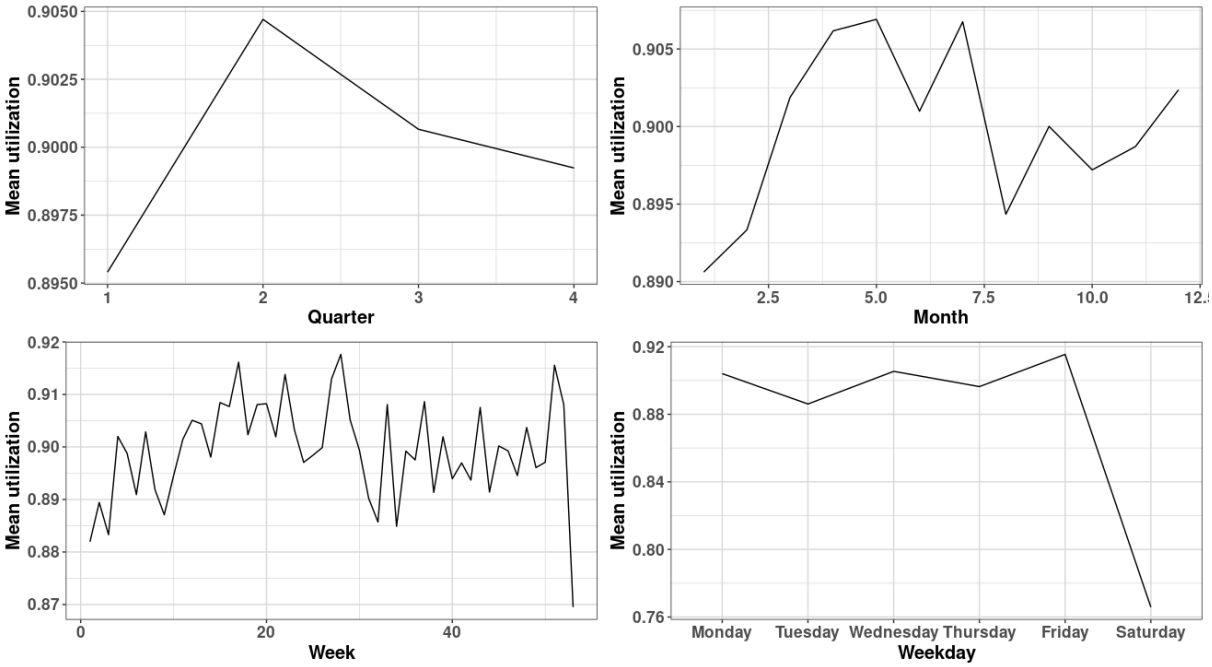

Figure 2.10: Mean utilization per period (2013-10-02 until 2019-12-31, n=30766)

In figure 2.10 we show periodic fluctuations in utilizations. If we compare this to figure 2.9, the figure backs the idea that lower utilizations are influenced by lower session lengths due to that the same length disruptions have a larger effect. We see that the smaller Saturdays have lower utilizations. We also see that the drop in productive times in August is reflected by lower utilizations in August. This is also true for the Christmas period. However, the degree of fluctuation is generally small except for day of the week.

We also see daily fluctuations which may have a large number of reasons. The most common reason is the day of the week. The low number of session hours on some Saturdays make that the day is very easily disrupted.

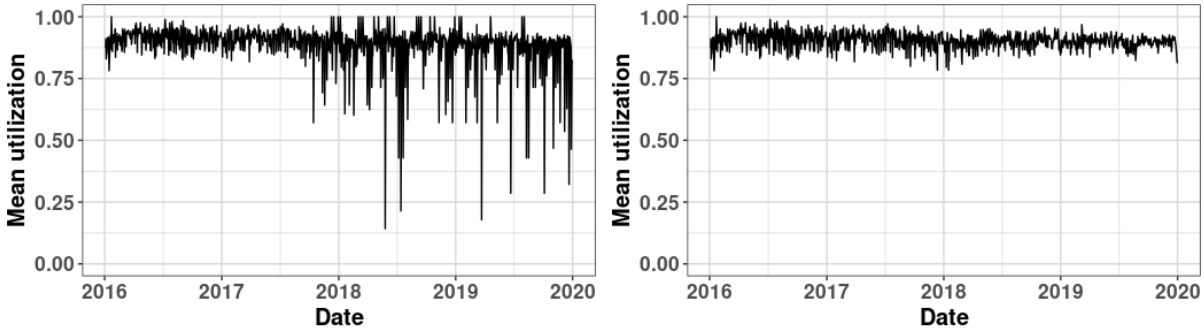

Figure 2.11: Mean utilization per day, with and without Saturdays (2013-10-02 until 2019-12-31, n=26717)

Figure 2.11 shows on the left hand figure the mean utilization per date, including Saturdays for the period we regard to be the most representative. The right hand figure shows the same, but excludes Saturdays. The effect of the Saturday is clearly visible.

In general we see that smaller session lengths in a period lead to a drop in the utilization. We find a correlation of $r(1670) = .38$, $p < .001$ between daily average utilization and total session length. We also find a strong correlation with daily average utilization and access time $r(1633) = .49$, $p < .001$, clearly indicating that access time tends to increase in busy periods, and can be used to predict utilization.

Regions and locations show fluctuations of their own. After grouping on location we obtain the per region results

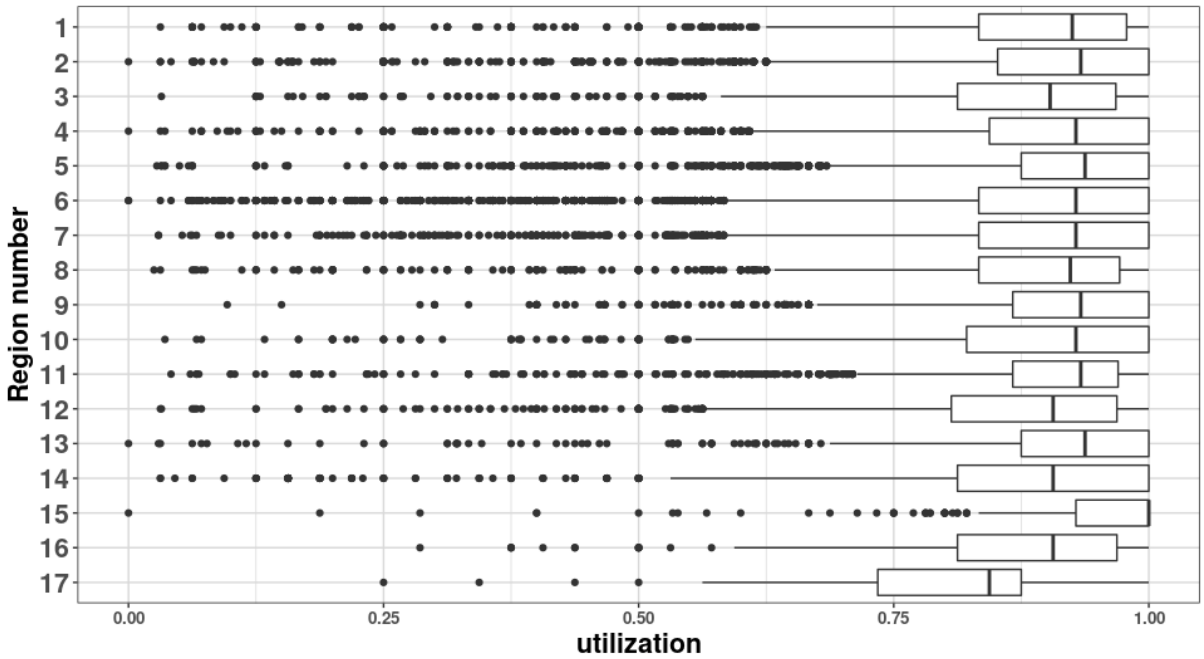

Figure 2.12: Distributions of utilization per region (2013-10-02 until 2019-12-31, n=50486)

In figure 2.12 especially region 17 is interesting. which clearly performs worse, however the region consist of only three locations. We learn from one of the organization's managers that the reason for the difference is that some regions and locations are strategic and therefore show under-performing results as they are to plan for the future or to tighten relationships with local partners.

To answer RQ1b we find the mean and standard deviation of the utilization over the last years.

Utilization per year			
Year	μ	σ^2	σ
2013	.670	.0338	.184
2014	.792	.0360	.190
2015	.850	.0288	.170
2016	.907	.0161	.127
2017	.907	.0158	.126
2018	.896	.0176	.133
2019	.897	.0173	.132
2020	.851	.0349	.187
2021	.877	.0194	.139

Table 2.3: Mean utilization and standard deviation per year (2013-10-02 until 2021-05-01, n=50486)

Table 2.3 shows the utilizations and variances of utilization per year. It shows the effects of the covid-19 as lower utilization and higher variance in 2020, and shows how the variance stabilizes towards similar variances after 2015 until 2020, just as in figure 2.7. The values suggest that 2021 will be comparable to the years before 2020, but we need to note that this is the running year. Again, the figures are similar to Westerink, 2021, but the utilization is slightly lower, and then obviously the variance is a bit higher due to that not all exactly the same schedules are considered. The all-time mean and standard deviation in utilization were .878 and .155.

2.4 Currently proposed prediction method

The currently proposed prediction method is a logistic regression model that predicts the probability that the utilization on a given date at one location is larger than average utilization of all sessions as observed in 2019 (0.922) (Westerink, 2021, section 1.2.1). The following predictors are included in the model.

1. *Region of the location that we want to schedule an appointment at.* Locations are distributed over regions that are geographically close. The number of locations per region differs. For instance, one of the regions has only one location whereas another region has 43.
2. *Year of appointment date.*, *Month of the appointment date.* and *Day of the week of the appointment date.* The year, month and day of the week of the appointment are included to model trends in the respective category.

3. *If mailing (marketing activity) in corresponding region occurs in month leading up to appointment date, then number of days between mailing date and appointment date, otherwise 0.* Mailing activities are included to model the increase in appointments due to marketing emails. Thereby, demand increases and utilizations rise. Mailing activities are not structurally registered, nor is it scheduled when the emails will be send. The effect of mailing activities should also be reflected in the access time. Due to the limited presence of mailing activity data, we leave out mailing activity data.

4. *Number of days between date of scheduling and date of appointment.* This is also called the access time, or waiting time of the appointment. The number differs over the assessed appointment when the date of the other appointment is different from the current date. The idea behind the predictor is that a long access time gives the session more time to fill with appointments, leading to higher utilizations, whereas for a short access the time is limited.

5. *Current utilization of appointment date (as observed on scheduling date).* The current utilization of the appointment date is been regarded important to indicate crowding. It is the utilization formed by the current scheduled appointments in the session. Similar to the current access time, if the current utilization is high, then it is expected that the location will have a high utilization in the end and if the current utilization is low it is expected that it takes more effort to fill the schedule with appointments and therefore the utilization is expected to be lower.

6. *Average current level of utilization during access time for all practitioners at the location.* This is regarded relevant as it shows how utilization changes in the days preceding to the appointment. This is the expected average utilization for all sessions at a location as observed at the date of scheduling the appointment.

7. *Total working time in the last four weeks for all practitioners at the selected location.* It models the time that needs to be filled in general with the idea behind the predictor being that a large number of hours is more difficult to fill and will therefore lead to lower utilizations. Our observation was that the session length positively correlates with utilization, suggesting the opposite. Using the predictor improved the model with 0.02%, but the predictor was excluded as it was 50% more computationally demanding.

The perspective of the predictors are from that of an incoming appointment scheduling activity. That is, it includes information available in general as well as information specific to the appointment scheduling moment, such as the current access time for the appointment.

Also, Westerink, 2021 acknowledges that it was not directly clear how the variables contribute towards utilization and therefore interaction effects were added, by multiplying the two variables as mentioned in Hardy, 1993. "Current utilization * access time" is used to model how utilization changes in the time preceding the appointment. The same interaction is also used for the average utilization by the interaction effect "Average current utilization during access time * access time". Finally interaction effects are used to model the working time in 4 weeks preceding the appointment, by multiplying the value with the utilization.

Since the prediction method currently does not predict the required continuous variable but classifies such that the result is the probability that the utilization is higher than the average, we cannot use the outcome of the prediction method as a benchmark for the model we aim to construct, we leave the comparison with our model to after we developed our model, and will then test that model using the predictors described in this section.

We use the logistic regression approach and fit such a model ourselves to the data. Following Westerink, 2021 we first leave out 30% (n=87748) of the observations to test the fitted model, leaving 70% to train the model. We tune the model by using 10-fold cross-validation on the 70% (n=206118) to obtain a measure of fit. The Brier score can be used as a measure. This is the equivalent of the mean squared error applied to the prediction of probabilities. Then the square root of the brier score is equivalent to the root mean squared error. The Brier score is .221 and the root Brier score is .470. Then we test the fitted model on the left-out 30%, which the model has not seen before.

	Actual	
Predicted	Lower	Higher
Lower	5872	5766
Higher	26431	49679

Table 2.4: contingency table using model as in Westerink, 2021

The resulting contingency table is shown in table 2.4. The corresponding classification test error rate is .370. We do the same for the models that include the interaction effects "access time * current session utilization" and "access time * current location utilization". The result is arguably of comparable size, with Brier scores of .217, .218 and .217 for the former, the latter and both of the interaction effects implemented respectively. The corresponding classification error rates are .362, .363 and .362 respectively. This allows us to use these results as a benchmark to compare the results of other models to, especially for the Brier score.

2.5 Conclusion

In the chapter we have extensively discussed the processes, the corresponding KPIs and we have provided a solid understanding of what influences the KPIs. The also answered research questions 1a, 1b and 1c.

We summarize the chapter with the research questions and the corresponding answers.

- 1a) What is the current access time for new patients? The access time during 2019, the most representative year, was 20.5 days and the all-time average was 18.8 days.
- 1b) What is the current mean and standard deviation in utilization? The utilization mean during 2019, the most representative year, was 0.897 and the standard deviation was .132. The all-time utilization mean was 0.878 and the standard deviation was .155.

- 1c) How does the assessed prediction method perform? Different models using multiple interaction effects have been performed. The model that is considered to be the best performing model among the logistic regression models reduces the classification error rate to .362 and cross-validated error to .217.

3 Literature Review

This chapter provides a theoretical background for the remainder of the study. Literature is reviewed to find models that are able to predict the expected utilization during a session. There are multiple approaches that could be taken with respect to literature. Alternative approaches are discussed in section 3.2. Section 3.3 goes more in depth to a machine learning workflow. The section discusses the most important considerations, challenges, problems and ways to overcome the problem. We have two goals in this chapter. First of all, we aim to find an methodology that helps resolving the prediction problem. That is a combination of a scientific perspective and a methodology with together with the challenges and problems that may arise. Secondly, from that approach, we aim to find models that are able to predict session utilization. We start the section with a description of the Topsis method, of which the usage was proposed by Westerink, 2021.

3.1 Topsis MCDA method

In the preliminary study, Westerink, 2021 proposed and evaluated a new scheduling system in which multiple criteria are analysed and weighted. The resulting system is capable of evaluating the possible appointment options on both criteria that are important to the organization's clients as well as criteria that are important to the organization itself. Westerink, 2021 discuss multiple multi-criteria decision analysis (MCDA) methods, and choose to use the Technique for Order of Preference by Similarity to Ideal Solution (Topsis) method. Topsis was originally developed by Hwang et al., 1981. The method evaluates every possible alternative (appointment options) on a set of criteria and determines how similar the method is to an arbitrary best alternative. We discuss the Topsis method more in depth here, since our prediction of utilization is one of the criteria which Topsis uses to rank. Following the notation by Behzadian et al., 2012, the method consists of the steps:

1. Create an $n \times m$ matrix with m possible appointments and n criteria. Each combination of appointment and criterion is referred to as x_{ij}.
2. Vector normalize the matrix using
$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{m} x_{kj}^2}},\ i = 1, 2, ..., m,\ j = 1, 2, ..., n$$
3. Determine the weighted normalized decision matrix v_{ij}. The method introduces weights here. The most important criterion has the highest weight. The least important criterion has the smallest weight.
$$v_{ij} = r_{ij} * w_j,\ i = 1, 2, ..., m,\ j = 1, 2, ..., n$$
where the weights can be normalized using
$$w_j = \frac{W_j}{\sum_{k=1}^{n} W_k},\ j = 1, 2, ..., n$$
such that
$$\sum_{i=1}^{n} w_i = 1$$
and W_j is the initial weight given to the indicator v_j, $j = 1, 2, ..., n$

4. Determine the best alternative A^*, and the worst alternative $A^{'}$:

$$A^* = \{v_1^*, v_2^*, ..., v_n^*\}$$

where

$$v_i^* = \{max(v_{ij})\ if\, j \in J;\ min(v_{ij})\ if\, j \in J^{'}\}$$

and

$$A^{'} = \{v_1^{'}, v_2^{'}, ..., v_n^{'}\}$$

where

$$v_i^{'} = \{min(v_{ij})\ if\, j \in J;\ max(v_{ij})\ if\, j \in J^{'}\}$$

That is, v_i^* is the vector of the best scores of all alternative options i. v_i^* is the maximum value of all alternative options i for the criteria for which higher is better ($j \in J$), and v_i^* is the minimum of all alternative options for the criteria where lower is better ($j \in J^{'}$). Conversely, $v_i^{'}$ is the vector of worst scores of all alternative options, where the minimum of all alternative options is taken when $j \in J$ and the maximum is taken when criterion $j \in J^{'}$.

5. Calculate the euclidean distance $S_i^{'}$ between the target alternative i and the worst condition $A^{'}$

$$S_i^{'} = \sqrt{\sum_{j=1}^{n}(v^{'} - v_{ij})^2},\ i = 1, 2, ..., m$$

and the distance S_i^* between alternative i and the best condition A^*

$$S_i^* = \sqrt{\sum_{j=1}^{n}(v^* - v_{ij})^2},\ i = 1, 2, ..., m$$

where d_{iw} and d_{ib} are euclidean norm distances from the target alternative i to the worst and best conditions, respectively.

6. Calculate the similarity to the best condition:

$$C_i^* = \frac{S_i^{'}}{S_i^* + S_i^{'}},\ 0 < C_i^* < 1$$

7. Rank the alternatives according to $C_i^*,\ i = 1, 2, ..., m$.

3.2 Alternative approaches

In literature, there are multiple ways to capture the problem. One way is to assume that a series of previously observed utilizations show patterns over time. The area of research that tries to capture the pattern observed in the series is time series analysis. We have also seen, not only in Westerink, 2021 but also in our own analysis that we can use observed factors that correlate with utilization. The scientific area of interest using observed events to learn predicting accurate outcomes, is machine learning. These are two general areas that have proposed models that are able to predict our outcome variable. A third approach is to find literature that is related to the scheduling problem and tries to predict utilization as an outcome variable.

There is an abundance of scientific literature devoted to both the areas time series analysis as machine learning, and models derived from the areas are used extensively in practice. During this literature review, we are particularly interested in predictive models as it is the aim of the study in general. We first introduce the relevant topics in literature.

3.2.1 The related literature approach

We learn from a systematic literature review (Appendix D) that there is not much scientific literature devoted to predicting utilizations in an appointment-based scheduling system. The studies either aim to predict another type of utilization, for instance the utilization of energy usage, or are used to predict utilization of walk-in processes instead of scheduling processes, for instance with the prediction of utilization of machines in a flow-shop system.

What's more is that the studies also not focus on the application of geographically dispersed or employee-specific predictions, but only regard one process in general. It also justifies this study as it is as far as we know the only study aiming to utilize both an approach for the prediction of utilization and the use of models that can handle geographical and employee specific differences.

We only find one comparable study that also has some valuable insights. Schiele et al., 2021 aim to predict hospital bed utilization in an operating room environment, using neural networks. The study is comparable as the beds are scheduled with patients, similar to how sessions are scheduled with patients in our setting. However in the study, the bed can only be occupied by one patient per day. This means that the patient-appointment is directly responsible for the bed utilization. In our setting, session utilization is a combination of the patient-appointments scheduled to the session. In other words, we cannot directly predict utilization from the patient appointments, but we need to dis-aggregate the session utilization such that the dis-aggregation can be predicted from the patient appointment.

Since we aim to use the prediction model in a decision support scheduling system, and we want the model to be able to predict for both geographical and employee-specific differences, and we observe that there is not much comparable literature, we regard the approach using related literature as a bad fit for this study. We believe that a more general approach is a better fit. With general we mean that we make use of models that are able to provide predictions in general, such as machine learning and time series. Another advantage of using a general approach is that we do not limit ourselves to the models evaluated in the studies found in literature.

3.2.2 The time series approach

In general, the idea behind time series analysis is to find common properties in data for the output variable that is to be predicted (R. Hyndman et al., 2021). Based on De Gooijer et al., 2006 and R. Hyndman et al., 2021, we can categorize time series models into autoregressive (AR) models, moving average (MA) models and exponential smoothing (ES). Combinations of these categories exist. Time series are often decomposed into patterns, often level, trend, seasonality, cyclic and a remainder (R. Hyndman et al., 2021).

Exponential smoothing and ARIMA models are the two most widely used methods in the field of time series forecasting. The approaches complement each other. The ARIMA methods aim to describe the autocorrelation. Exponential smoothing methods aim to describe the trend and seasonality in the data R. Hyndman et al., 2021.

ES used to be seen as an *"ad hoc techniques for extrapolating various types of univariate time series"* (De Gooijer et al., 2006). In simpler words, exponential smoothing produces a weighted combination of past observations. R. J. Hyndman et al., 2002 present, and Taylor, 2003 extends a helpful categorization of ES methods. The methods are categorized based on one of five types of trend (none, additive, damped additive, multiplicative and damped multiplicative) and one of three types of seasonality (none, additive and multiplicative). Among these 15 types of ES methods, the model with no trend and no seasonality is known as simple exponential smoothing. The model with additive trend and no seasonality is known as Holt's linear method (Holt, 2004), and the models with additive trend and either additive or multiplicative seasonality is known as Holt-Winters' method (Winters, 1960).

AR models are autoregressive since the variable of interest is predicted using a linear combination of the past observations of the variable. Autoregressive models are normally restricted to stationary time series. Time series are regarded stationary when it's statistical properties (e.g. mean, variance, autocorrelation) do not depend on the moment at which the series is observed. One technique to construct a stationary series is called Differencing. Differencing is computing the difference between successive observations. The series obtained after differencing is often stationary and the series that needs to be differenced in order to be stationary is called an integrated series. Moving average (MA) models - not to be confused with moving average smoothing - uses past forecast errors instead of past observation of the variable as predictors for the variable of interest (R. Hyndman et al., 2021). The AR and MA models can be combined into the autoregressive moving average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) model. Integrated means that the series needs to be differenced in order to have a stationary series. Box et al., 1970 develop the Box-Jenkins approach. This is a three-stage iterative method in which time series models such as the ARIMA model can be identified, estimated and verified. Box et al., 1976 present an updated version in which also among others intervention and outlier detection is described. De Gooijer et al., 2006 present an overview with examples of real applications of ARIMA models. For instance, a one month ahead department store sales forecast study compared to simple exponential smoothing methods by Geurts et al., 1986. Also, there are extensions to the ARIMA models. For instance the VARIMA model, which is a multivariate generalization. The ARIMAX model aims to include exogenous predictors to ARIMA models simply by adding the predictor as a covariate to the right hand side of the regression. The SARIMA model uses seasonal parameters. A combination of the ARIMAX and SARIMA (SARIMAX) model exists as well.

Most time series models require a steady, stationary, non-interrupted time series. That is, no missing data may exist in the series. As we are trying to predict the session utilization, our data shows large degrees of missing data per session as that is how the system initially was designed. Every "single work schedule" that interrupts the ongoing work schedule for only one specific date is such an event. Also, session interruptions often result in missing data points. The same is true for sessions without patient appointments which are excluded. Then there are periods in which a session is not active, and finally,

a session is only at least structurally defined for every two weeks, leading to a recurring session every two weeks. This results in a large number of individual time series with a small number of observation. There are models that are able to handle missing data or methods that fill the gaps in data such as imputation. However, as it is the aim of models to find patterns in the series and in section 2.3.2 we do not see a clear pattern over time, it is our perception that time series analyses will not contribute to a good prediction of utilization for the session and we are still able to implement features such as the day of the week in other type of models.

3.2.3 The machine learning approach

The final approach is the machine learning approach. Although inherently our process shares many characteristics with time series, the specific 14-day session structure and the geographical and employee-based differences we try to predict make it such that we regard the machine learning approach as the most applicable approach. Section 3.3 describes a more in-depth explanation of the research topic.

3.3 Machine Learning

The machine learning area studies algorithms that consume data in order to provide predictions. Machine learning has many facets, which contain general characteristics. There is much literature covering the facets. We use James et al., 2013 and Géron, 2017 as a guideline.

- The type of supervision
- The type of response variable
- The approach towards learning

The type of supervision comprises the question of what the aim of learning from data is. James et al., 2013 and Géron, 2017 name three general types of learning. The first type of learning is supervised learning. That is, the system uses data with corresponding response variable (label) as input data on which it trains. Based on the training set, the system tries to identify the relationship between the input parameters and the response variable by means of setting the system parameters such that a performance measure is optimized. The simplest example is arguably linear regression. The second type is unsupervised learning. In unsupervised learning, systems try to determine general characteristics in the total data set, without having a label of what is the right output. Approaches of unsupervised learning include clustering. Then there is reinforcement learning. Reinforcement learning is about identifying a strategy of taken steps that generate a best possible reward. Reinforcement learning is not relevant for this study.

The type of response variable depends on the type of learning that is used. For supervised learning, common response variables are either continuous or categorical. Systems that return a continuous response are named regression models. Systems that return a categorical response are classification models. In unsupervised learning, the response is generally a group or a division of the input data, as there is no response variable. The result is then a group of observations that are thought to be related by assessing the input variables. In reinforcement learning the result is the decision in the current state of the system.

The way of learning is about how the system trains itself. The general distinction is batch learning or online learning. In batch learning the system learns at once, with all available data, such as with linear regression. In online learning, the system learns incrementally on individual instances or smaller batches, where each learning step is done quickly. The clear advantage is that systems are able to update over time (Géron, 2017).

We learn from James et al., 2013 and Géron, 2017 that using models to solve our prediction problem draws up some general problems that can prevent such models to perform well. The problems may seem evident, but are nonetheless important to the final performance and are therefore discussed. The problems are:

- Insufficient quantity of training data
- Non-representative training data
- Poor quality data
- Irrelevant features
- Overfitting to training data
- Underfitting to training data

Insufficient data is the problem that we require a large enough data set to make models learn well. Following Géron, 2017, the degree of the input data should be in the thousands examples for simple problems and in the millions for complex problems such as image or speech recognition.

Non-representative data makes it difficult for models to generalize on new occurrences, as it makes the system think it fits well, whereas it only does on the training data. It biases a model.

Similar to non-representative is poor-quality data. Examples are outliers, noise and errors. Solutions to poor-quality data is to discard or impute outliers. Another approach in the event of missing features can be to train a model with and a model without the feature.

One of the most important steps in a machine learning methodology is feature engineering. Irrelevant features do not contribute to a good model fit. The steps in feature engineering are feature selection, feature extraction and creating new features. Feature selection is choosing those features that contribute to good model fit. feature extraction is combining features that together contribute to a better model fit. Creating new features is more or less gathering new data.

Overfitting to training data occurs when the model performs so well on the training data that the model does not generalize well over new occurrences any longer. A pattern might be followed that does not truly exist and it often happens with complex, flexible models. Possible solutions are to simplify the model by using fewer parameters or by gathering more training data so the system.

The alternative is underfitting. It is choosing the model such that it is not able to model the underlying process well. Solutions to underfitting is to use different types of models that are able to understand the underlying process better. To use more predictors to the algorithm that or to reduce constraints on the model so that it can be more flexible.

In James et al., 2013 the Bias-variance trade off plays an important role. The Bias-variance trade off is the trade-off between a good model fit and a good generalization performance. Variance in the trade off is the amount by which the model result would change if it was trained on a different training set. Bias is the error that occurs by evaluating the model on real world data. Often the degree of flexibility is used. Variance tends to increase with flexibility as a highly flexible model trained on different data will follow the patterns occurring in the different data set. On the other hand, less flexible models assume that the underlying process is less complex. That is not necessarily true. The models therefore generally result in more bias. The amount by which both change by altering the flexibility is not of the same order. Per James et al., 2013, bias often reduces faster than variance increases. It is the aim to find a model in which both the bias and variance are low.

3.3.1 Model validation

The existence of the bias-variance tradeoff implies that there is a necessity to find an approach which yields valid models. During the development of the model, which is discussed in sections 3.3.2 to 3.3.9 there are multiple approaches to assess the validity of the model. These approaches are discussed now, since the approaches are often used not only for the evaluation of the final model, but also for intermediate steps of model development. The intermediate steps are then discussed in the named sections 3.3.9 to 3.3.9. Often cross-validation is used. Cross-validation makes use of the validation set approach. The validation set approach is leaving out a sample of the data to validate the trained model on. This is the test set. The model is trained on the other part of the data, which is the training set. Cross-validation does this k times for k samples of data. This yields the question of what the size of the sample (k) must be, or in other words how many samples there must be. One method is leave-one-out-cross-validation (LOOCV), which, for every instance, trains the model on all but this instance. The model is then tested on the instance. Another approach is k-fold-cross-validation. In k-fold-cross-validation, k samples are constructed and the model is trained on the other $k-1$ samples. Values of k that have been shown not to suffer from excessive bias and very high variance are in the order of $k = 5$ or $k = 10$ (James et al., 2013). To measure the performance of the cross-validated model, the mean of the mean squared error between each of the k sets can be used for regression problems.

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i$$

3.3.2 Machine Learning methodology

A good way to cope with the mentioned challenges is by following a methodology. Géron, 2017 offer a methodology for performing a machine learning study. The methodology covers the following steps

1. Look at the big picture
    - Frame the problem
    - Select performance measure

- Check the assumptions

2. Get the data
    - Create the workspace
    - Collect the data
    - Glance at the data structure
    - Create a test set
3. Discover and visualize the data to gain insights
    - Visualize data
    - Look for correlations
    - Experiment with combinations
4. Prepare the data for Machine Learning algorithms
    - Data cleaning
    - Handling specifically typed attributes
    - Transform and scale data
5. Select a model and train it
    - Train and evaluate on training set using cross validation
6. Fine-tune the model
    - Grid search, randomized search, ensemble methods
    - Analyze the best models and their errors
    - Evaluate model performance on test set
7. Launch, monitor and maintain the system
    - Create automated workflow
    - Automatically train model and fine-tune parameters periodically

There are more methodologies, for instance the "Cross industry standard process for data mining,(CRISP-DM)" by Chapman et al., 2000. CRISP-DM breaks down into the six phases: Business understanding, Data understanding, Data Preparation, Modeling, Evaluation and Deployment, but as CRISP-DM and many other methodologies offer a different flavor of the same basic approaches anyway, we do not discuss these. In the methodology discussed by Géron, 2017, we can take a shortcut to step 4 since step 1 to 3 are to a large extent already covered by chapter 1 and 2. Subjects from the methodology that need to be discussed are feature transformation and scaling, the handling of specifically typed attributes and the methods to fine-tune the model. Subjects that are not listed in the methodology but are important topics and considerations in machine learning in general are model selection and hyperparameter tuning. Not only this methodology, but most other methodologies have a large focus on the models that are used. The methodologies do not discuss the relationship to reality or the problem that a model must help solve. Therefore we refer to the research structure described in chapter 1 since it discusses the

relationship to reality as well in terms of access time and variability in utilization as well as how we aim to measure the actual performance.

3.3.3 Performance measures

Performance measures should be chosen to validate the prediction models and to structurally chose among proposed models. The most important performance measure for model selection is prediction accuracy (De Gooijer et al., 2006). There are a large number of accuracy measures. However due to the decision to take a general machine learning approach, we decide to restrict ourselves to general, well-known, accuracy measures. For regression problems, typical options are based on the squared error and the absolute error. Both are measures of distance between an observation and a prediction. Options among these types of error are the mean squared error (mse), root mean squared error (rmse) and mean absolute error (mae). The rmse is simply the square root of the mse. Compared to the mae, the mse and rmse penalize large errors more. The suitability of prediction accuracy measures is based on the type of data and the business environment. More specifically, the type of loss, which the performance measure should reflect, leads this choice. Since the model proposed by Westerink, 2021 did not produce accurate predictions specifically for lower session utilizations, we decide to penalize larger errors more. Hence, we choose the rmse as the prediction accuracy measure, which is more interpretable than the mse as it is on the same scale as the utilizations. The following paragraph gives an example.

Since utilization is defined on the unit interval, the error between the observed and predicted utilization is defined on the interval [-1, 1]. Therefore the squared error is defined on the interval [0, 1] hence the mean squared error is also defined on the interval [0, 1]. As a result, the rmse is always larger than or equal to the mse. Due to the smaller number on a different scale, the mse might imply a too optimistic interpretation of error.

3.3.4 Specifically typed attributes

As machine learning models generally have a solid mathematical foundation, the models often work with numbers. For instance, linear regression is often based on least squares optimization. This draws up the challenge of how to model non-numeric data. Especially for, for instance locations this may pose a challenge. Every location has an employee ID, but two employee IDs that are close to each other should not be considered similar due to that their IDs are close. One option is to use one-hot encoding. One-hot encoding is using the value 1 if employee X has ID Y, 0 otherwise. The drawback is obviously the large number of predictors this leads to. The alternative, similar to using the IDs, is called label-encoding.

3.3.5 Transformation

Although mentioned as being an important step by Géron, 2017, transformation is not really discussed by both Géron, 2017 and James et al., 2013. R. Hyndman et al., 2021 discuss different types of transformation. The focus is time series methods, but the discussed transformations are more general. The value of transformations lie with simplifications of the observed patterns to make the patterns more consistent across the data. This is done by alleviating sources of variation. Possible options are:

- Calendar adjustments
- Population adjustments
- Inflation adjustments
- Mathematical transformations

Calendar transformations are to adjust the observations into comparable measures. We see that with our data. Some months contain more days, and therefore contain more appointments. Hence dividing by the number of days of the month scales the variable to comparable units. The number of appointments in a session is another example. One hour sessions can only obtain 4 quarter-length appointments whereas two hour sessions can already hold 8. A possible transformation is to divide by the session length.

Population transformations are comparable to calendar transformations, and are used to adjust for the difference introduced by using a total. An adjustment is to divide the total by the number of persons that contributed to the total to obtain a per-capita number.

Inflation adjustments are used to model monetary value over time, but are not relevant for this study.

Mathematical transformations consist of a larger group of functions that transform a predictor in such a way that the result is easier to use. Logarithms are often used as they are interpretable, make data more uniformly distributed and linearizes exponential growth. Power transformations are also used to make data more normally distributed. The Box-Cox transformation features a combination of both. The transformation depends on a parameter λ, where 1 is equivalent to not transforming data, 0 is equivalent to using a natural logarithm scale and anything else is equivalent to using a power transformation:

$$y_i^{(\lambda)} = \begin{cases} \frac{(y_i+\lambda_2)^{\lambda_1}-1}{\lambda_1}, & \text{if } \lambda_1 \neq 0 \\ ln(y_i+\lambda_2), & \text{if } \lambda_1 = 0 \end{cases}$$

As described in Box et al., 1964. Note that in literature there is also a single parameter variant that was proposed in the same paper. The two parameter variant is able to accommodate for negative values for y. The parameter λ should be chosen such that the seasonal pattern is approximately equal across the series.

3.3.6 Scaling

Géron, 2017 discuss how machine learning algorithms don't perform well on predictors with different scales and offer two standard approaches to deal with the problem to scale the predictors. These are normalization and standardization. Normalization is shifting the predictor vectors such that they are scaled between 0 and 1. This is generally done by

$$X_{scaled} := \frac{X - min(X)}{max(X) - min(X)}$$

Standardization is scaling such that the predictor values always have mean 0. the result is divided by the standard deviation in order for the resulting distribution to have the unit variance.

$$X_{scaled} := \frac{X - \bar{X}}{\sigma_X}$$

The range of values is not specified. The advantage is that it is less affected by outliers.

By means of an example James et al., 2013 also highlight the importance of scaling. In an example using ridge regression, the conclusion is "It is best to apply ridge regression after standardizing the predictors" due to that the predictors not only depend on the hyperparameter (3.3.8), but also on the scale of the other predictors. Their definition of standardization is equivalent to the definition discussed in this section.

3.3.7 Model selection

Model selection is the process of selecting a model. There can be many approaches. In James et al., 2013 a few approaches are mentioned, and three categories are considered that increase model interpretability and predictive quality. Model interpretability is difficult to define mathematically (Molnar, 2019). General definitions are "Interpretability is the degree to which a human can understand the cause of a decision." Miller, 2017 or "Interpretability is the degree to which a human can consistently predict the model's result" Kim et al., 2016

Model interpretability is about the inclusion of predictors that add value and the exclusion of predictors that do not add value. Using irrelevant predictors in the model leads to an unnecessary degree of complexity. Three categories are discussed. Subset selection, shrinkage and dimension reduction. The methods all have their own ways of controlling the model variance.

Subset selection is the method of identifying the predictors that explain the outcome the best. There are multiple methods, for instance best subset selection is the process of fitting all $\binom{p}{k}$ combinations of p by increment k in each iteration, to obtain the best set of predictors among all combinations of p. This way, a set of at most p predictors is obtained. Forward selection is similar, but instead of fitting all models, the method remembers the set of best predictors in each iteration and adds the predictor that contributes the most to the solution. As a measure of optimality there are multiple approaches. This can be the cross-validated error for each model, or the measures such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC) or the adjusted R^2. James et al., 2013 state that the adjusted R^2 is not as well motivated in literature as the others.

Shrinkage is the process of training a model on all predictors after which the selection of predictors is shrunken until a satisfactory set of predictors is obtained. The way this is done, is by constraining/regularizing coefficient estimates in such a way that the coefficients are shrunken to zero. Often used shrinkage methods are the ridge regression and the lasso. The methods are quite similar, but the lasso has the advantage of being able to set the predictor coefficients to 0, which is equivalent to deselecting the predictor. The lasso is also able to select a subset of predictors.

Dimension reduction is the process of reducing the number of predictors by modelling one predictor using the others such that meaningful properties are retained, but the predictor is omitted. Methods that are able to do this include principal component analysis and partial least squares.

3.3.8 Fine-tuning methods

After finding well performing models, there are a few options to improve the model result by altering the model hyperparameters. One option is hyperparameter optimization. Hyperparameters are parameters that influence the learning rate, and can be used to control overfitting. Not all models require hyperparameters. Some models require many hyperparameters. Examples are the shrinkage parameter in the Lasso. Selecting the parameter can be done in multiple ways. One way is creating a grid of parameter values and selecting the best parameter among these values using cross validation.The parameter is often denoted by λ and also goes by the name of regularization coefficient (Bishop, 2006).

The approach behind hyperparameter optimization is to find a good combination of regularization parameters that prevent the model from overfitting, but also allow the model to fit well. One option is grid search. Grid search is creating a list of all combinations of the hyperparameters. This is also known as exhaustive search and can be very time consuming. A second option is to randomly search over the hyperparameters for it to search a good combination of parameters. This approach is called a randomized search and is generally less time consuming, but it does not guarantee finding good hyperparameters. This is the problem that a good algorithm addressess: the tradeoff between execution time and finding good parameters. Bergstra et al., 2011 discuss automatic sequential hyperparameter optimization algorithm and find that these outperforms the randomized search algorithm. Among the discussed methods, the Tree-structured Parzen Estimator Approach (TPE) is one that is seen in practice as well, in the Optuna framework (Akiba et al., 2019) for instance.

Another option for fine tuning models is to combine models that show promising results. These methods are called ensemble methods. There are a few types of ensemble methods. Common types are bagging, stacking and boosting. Bagging - bootstrap aggregation - is the approach of creating multiple training sets from an original set, by means of bootstrapping. Then train models on the training sets, and finally average the results. Stacking refers to the idea of first training different models, and then train a combining model that is able to combine the results of the first trained group of models. The idea behind boosting is similar, but instead of training a model on bootstrapped training set, new models are fit on the result of the previous model. That way, the model grows over iterations.

3.3.9 Curse of dimensionality

As most traditional statistical techniques for regression are primarily dedicated to situations in which the number of predictors is significantly lower than the number of instances in the training set, issues arise in situations where the number of predictors is large. The definition high-dimensional setting is introduced. This is the case where the number of features is larger than the number of instances, but the issues occurring also arise when the number of predictors is close to the number of observations.

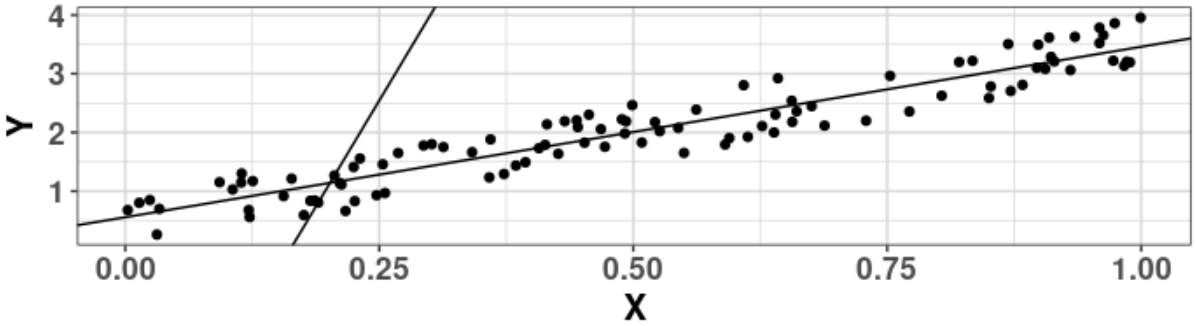

Figure 3.1: Curse of dimensionality in practice

To demonstrate the issue, figure 3.1 shows two linear regression models. The data are originating from first two vectors X_1 and X_2 both consisting of 100 random $U[0, 1]$ values. Then we construct a vector Y using the linear function $Y = 3X_1 + X_2$. If we fit a model to all the 100 points then the model fits well, but if we fit the model using only 2 of the values, whether or not the model fits well highly depends on the two observed instances. The model is too flexible and overfits the data.

James et al., 2013 state three important points that contribute to a successful implementation of a machine learning model. These are :

- Regularization or shrinkage plays a key role in high-dimensional data.
- Appropriate tuning parameter selection is crucial for good predictive performance.
- The test error tends to increase as the dimensionality of the problem increases unless the additional features are truly associated with the response.

In general, adding more predictors increases the risk of overfitting. This highlights the importance of approaches such as cross-validation. The first point highlights the importance of model selection, and the second point highlights that such models, often incorporating a constraining parameter, require a structured approach towards setting this parameter. This directly relates to hyperparameter tuning and the grid-search approaches discussed in the section fine-tuning the model.

Bishop, 2006 mentions that the choice of using a large number of individual training sets is a situation in which the risk of overfitting can prevail. This can be resolved using one large training set. This is related to the problem occurring at the organization in the sense that we have to choose between making a prediction for every work schedule, or look to incorporate the location and employee as predictors for the variance between the two. While this choice primarily arises from the situation of choosing time series analysis to model the process, this is still a relevant consideration.

3.4 Machine learning models

In the previous section we have discussed machine learning methodologies. We presented the challenges, the most important considerations and some examples of methods and models that help to overcome the challenges. In this section we discuss the models that are present in machine learning. There are many models in different fields, with different

purposes. We only discuss models that are in the field of supervised learning and are able to result a prediction of utilization.

The related field of machine learning can be subdivided into the more traditional models and newer models. The traditional models include the linear regressions, generalized additive models and tree-based methods. Newer methods are generally based on a form of deep learning. Ensemble methods such as boosting models can hand be seen as a category of models on its own, but the iterations of a boosting model consist of'weak learners', more traditional models, such as trees.

3.4.1 Linear regression

Linear regression methods originate back to 1805 at which the least squares method was proposed (Stigler, 1986). Linear regression often serves as a basic model. linear regression models are fitted by the least squares approach. The residual sum of squares is minimized. The approach is not very flexible, but the approach is simple and easily applicable. A similar structure is found in polynomial regression. Instead of only adding the predictors multiplied by the coefficient, in polynomial regression the function can be a polynomial. This adds flexibility to the model. The basic linear regression model tries to minimize the quantity

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

3.4.2 Lasso and ridge regression

The lasso and ridge regression are extensions of linear regression. In ridge regression, first presented by Hoerl et al., 2000, instead of minimizing the residual sum of squares, an l_2 shrinkage penalty ($\sum_{j=1}^{n} \beta_j^2$) is added, which is small when the coefficients of the model are close to zero. The penalty is multiplied by a tuning parameter λ. This is the hyperparameter, or regularization parameter as discussed in section 3.3.8. When the parameter is zero, the penalty has no effect. When the parameter approaches infinity, the influence of the penalty increases. The penalty is applied to all coefficients, except the intercept. The advantage of ridge regression over least squares is discussed. The tuning parameter ensures that ridge regression is able to deal with the bias-variance trade-off 3.3. Ridge regression has the form

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{n} \beta_j^2$$

The lasso, presented in Tibshirani, 1996 is similar to ridge regression. The difference resides in the form of the penalty ($\sum_{j=1}^{n} |\beta_j|$), known as the l_1 penalty multiplied with λ. One of the disadvantages of ridge regression is that the model coefficients will never truly be zero as $\lambda \xrightarrow{\infty}$. The effect of this is that ridge regression will always require all parameters. The lasso offers an alternative as it sets the coefficients to zero. The predictors corresponding to zero coefficients can then be omitted. The lasso has the form

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{n}|\beta_j|$$

Many subtle alternatives of ridge regression and the lasso exist. For instance the elastic net by Zou et al., 2005, which combines the lasso and ridge regression penalties using an additional parameter that controls the amount by which the penalty is in the form of l_1 or l_2: $\lambda\sum_{j=1}^{n}(\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$. The elastic net has the form

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{n}(\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$$

Meinshausen, 2007 propose the relaxed lasso. The relaxed lasso makes use of two parameters and passes two phases. In the first phase the variables are selected using the general lasso. In the second phase the shrinkage parameter is used on the subset of predictors that were selected in the first phase.

3.4.3 Splines

A spline is a combination of polynomial functions that are fitted to different regions of the training data, under the constraint that the resulting function is continuous. A region is separated by a point which is called a knot. The number of knots defines the degree of flexibility of the model and can be determined using cross-validation.

Regression splines offer an easier way to produce flexible fits than polynomial regression due to the lower order polynomial the spline requires. Flexibility is increased by increasing the number of knots while keeping the order of the polynomials the same, whereas in polynomial regression flexibility is increased by increasing the order of the polynomial. It is due to this that regression splines often produce better fits than polynomial regression.

Smoothing splines (Reinsch, 1967) share the general idea, but add a penalty and smoothing parameter λ (hyperparameter) similar to what ridge regression does to linear regression, which controls the roughness of the spline. Without the tuning parameter the smoothing spline is far too flexible. The smoothing parameter can be chosen using cross validation. Especially LOOCV can be computed very efficiently. The smoothing spline has the form

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda\int g''(t)^2 dt$$

in which lambda is the smoothing parameter, g a function that minimizes both the residual sum of squares ($(y_i - g(x_i))^2$) as well as ensures that the function is smooth. $\lambda\int g''(t)^2 dt$ is the penalty to variability in g.

A spline method proposed by Friedman, 1991 is multivariate adaptive regression splines (MARS). The MARS method fits linear regression models to each predictor and observation. The method introduces knots where two different linear relationships achieve the smallest error until a large degree of knots have been identified. The resulting model

will be prone to overfitting. Therefore a number of knots can be removed using cross validation. Due to the large number of knots the training process is generally slow, but the advantage is that Boehmke et al., 2019 the resulting function is fast and can be deployed easily. Another advantage is that the method requires little predictor scaling and transformations and the method is able to work with both numerical as categorical predictors.

3.4.4 Generalized additive models

A generalized additive model (GAM) is another nonlinear extension of the linear regression model. In the GAM, each combination of coefficient and predictor is replaced by a linear function to allow for non-linear relationships (Hastie et al., 1984). The result is then the summation of all of the individual functions. The advantages are the usage of nonlinear functions allowing the model to make more accurate predictions. Since the model is additive, the effects of each individual predictor is still visible, increasing the interpretability of the model. The drawback is that interaction effects cannot be directly used since the model is restricted to be additive, however the interaction effects can be added manually. The GAM has the form

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + ... + f_p(x_{ip}) + \epsilon_i$$

observe that each of the $\beta_j x_{ij}$ component has been replaced by a smooth nonlinear function $f_j(x_{ij})$. This way, each of these function components can serve as a building block to fit the model. Splines can for instance be used as one of these blocks in the GAM. Polynomial regression, local regression or a combination can also be used.

3.4.5 Trees and random forest

A decision tree-based method is a graphical approach towards prediction. The methods division the predictor space into several regions. The division into section can be visualized by a tree. Therefore the term decision tree is often used. Tree methods itself are generally very interpretable, but not that good in terms of predictive quality. The pros and cons are discussed by (James et al., 2013).

- Trees are easy to interpret, arguable even more easy than linear regression.
- Some argue that decision trees are similar to human decision making. This statement is similar to the definition of interpretability of Kim et al., 2016 as discussed in 3.3.7.
- Trees can be plotted, which make the tree concept even more easy to interpret.
- Trees are able to use categorical predictors without having to transform the predictor.
- As discussed, the downside of trees is that trees generally do not have the same prediction quality as other models such as regression may have.

The downside can be alleviated by the aggregation of a large number of trees, like the random forest method (Ho, 1995), or the in section 3.3.8 discussed boosting and bagging

methods. In this case, bagging (Breiman, 2001) is a special case of the random forest. Where in bagging the trees are trained on the bootstrapped training sets and in every iteration all predictors are used, in random forests this is not the case. In every iteration a random sample of the predictors is taken as predictors for the tree construction. This way the trees are not correlated.

3.4.6 Support vector machines

Support vector machines (SVM) were first developed for classification, we discuss them as the SVM has been extended towards the use for regression (SVR), proposed by Drucker et al., 1997. The principle of support vectors is the maximum-margin hyperplane. For classification this is the hyperplane for which the distance to the nearest data point is maximal. A problem with the least squares method such as in linear regression, is that it is responsive to outliers. Therefore the SVM uses a classifier called the ϵ-insensitive error function Bishop, 2006. In SVR, the difference is to create give the regression a margin ϵ in which as many occurrences as possible should lie Boehmke et al., 2019. One of the main advantages of SVR is that the model parameters are obtained using a convex optimization problem, making the local solution a global solution Bishop, 2006. There are multiple formulations for the optimization problem, for instance the formulation described by Smola et al., 1998:

$$minimize \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$subject\ to \begin{cases} y_i - \langle w, x_i\rangle - b & \leq \epsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i & \leq \epsilon + \xi_i^* \\ \xi_i,\ \xi_i^* & \geq 0 \end{cases}$$

C controls the tradeoff between model flexibility and One of the assumptions of the first approaches towards SVMs and SVRs was that there always exists a separating hyperplane, i.e. the problem is feasible. That assumption is resolved in this formulation by the slack variables ξ_i and ξ_i^*. The optimization problem can be solved using Lagrange optimization and the dual of the problem. The major downside of SVRs is that they are not easily trained, especially if the number of training instances is large. This is due to that parameters need to be estimated for each instance in the training set.

3.4.7 Gradient boosting machines

From existing literature we conclude that boosting is not only an approach to finetune other models, but can be seen as a class of models on itself. The boosting models still make use of other kind of models - they are generally tree-based - called ‘weak learners‘. The power of these models comes from the ensemble effect. This is pointed out by the authors of Freund et al., 1995, in which the Adaboost model is presented. This is one of the first applications of boosting:

"Perhaps the most surprising of these applications is the derivation of a new application for "boosting", i.e., converting a "weak" PAC learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy."

Specifically the gradient boosting machine (GBM) is a group of models that appears to be applicable. Gradient boosting machines are a combination of sequential trees where each tree tries to improve on the result of the one before. The gradient refers to the gradient of the loss function. That is the gradient of the difference between the observed value and the predicted value of a newly constructed tree. Boehmke et al., 2019 point out that gradient boosting algorithms are hard to beat with other algorithms, when they are tuned correctly. There are a few variants of gradient boosting machines. The basic gradient boosting process is:

1. For B trees
   (a) Fit a tree with d splits
   (b) Update the objective function with a shrunken version of the new tree.
   (c) Update the residuals for each tree.
2. Return the model

Friedman, 2002 present stochastic gradient boosting. Stochastic gradient boosting offers a subtle alternative to gradient boosting, by fitting the model, typically a random forest, to a subset of the training set. Each of the subsets is randomly drawn from the training set without replacement. This is where the stochastic element comes from. The result is reduced tree correlation and therefore better prediction accuracy (Boehmke et al., 2019)

Chen et al., 2016 present extreme gradient boosting (XGBoost). XGBoost not only offers a more regularized variant to the GBM, but XGBoost also generalizes other aspects. The model allows for the use of other learners instead of random forests, such as linear models. XGBoost also implements Dropouts meet Multiple Additive Regression Trees (DART) (Rashmi et al., 2015). This is an approach to reduce overfitting and makes model training faster. Trees that are fitted in later iterations tend to only contribute to the prediction accuracy for only a very small number of instances. This is called over-specialization and can be seen as overfitting. The basic idea of DART is to offer a structured approach to not continue the boosting iterations but instead drop the node. The result is a lower degree of overfitting and faster fitting of the whole model. One particular downside of XGBoost is that can be time consuming when both the number of features and the number of instances are large. That is due to that the model needs to estimate the importance of the predictor on all of the splitting points, for instance in a tree model.

Ke et al., 2017 aim to solve the problem. The LightGBM model can be seen as a much faster alternative to XGBoost, while approaching the same level of predictive performance. Two techniques are used. Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS reduces the number of data instances by only using the number of instances that have larger gradients, meaning that only the instances that are regarded more important to the predictive performance are used. EFB combines mutually exclusive predictors to reduce the number of predictors. A greedy algorithm is used for the decision of which predictors should be combined together since Ke et al., 2017 show that the problem of partitioning features can be reduced to the graph coloring problem, which is NP hard.

3.4.8 Deep learning

Deep learning is a very active area of research in machine learning. The foundation of deep learning are neural networks James et al., 2021. The neural network starts with a vector of input variables and builds a nonlinear function that predicts a response. This is also the case for boosting, GAM models or random forests or regression trees, but the difference is in the structure. There are many variants of neural networks, for both classification as regression. Examples are convolutional neural networks (CNNs), or recurrent neural networks (RNNs) more often used for time series and other types of sequential processes.

Neural networks consists of several layers. The vector of predictors is named the input layer. Each element of the input layer is used as input for each element of what is named the hidden layer. In the hidden layer, the input for a layer is processed through an activation function, that transforms the input variable. After processing in the hidden layer, the transformed input variables are fed to an output layer where the results are combined into one output variable.

The activation function is specified beforehand. Options include the sigmoid function that is also used in logistic regression, the rectified linear unit (ReLU) or the normalized exponential function known as the softmax function.

A specific type of neural networks are recurrent neural networks. First presented by Rumelhart et al., 1986, RNNs are networks in which the specific occurrence of an instance matters. Examples include time series and recorded speech. In RNNs we can model the input layer and the output layer as a sequence of occurrences. Not only does each activation function produce a result to the output function, it also produces a value that influences the next activation function. The next activation function then combines the result of the previous activation function as well as the input of the predictor at that time, quite similar to how time series models incorporate exponential smoothing and auto-regressive models in general.

In RNNs, retaining elements from further back in time. This is known as the vanishing gradient problem. Then the input units retained from further back in time either blow up or tend to zero. An approach is to store input units from further back in time unchanged (Calin, 2020). The long short term memory model is capable of doing this. First proposed by Hochreiter et al., 1997, the LSTM is an extension of the RNN where the model can decide on what to forward to its forget state and what to store in its long-term memory. This way seasonal effects and early signals that change the outcome are retained but non-affecting elements are forgotten. The downside to the LSTM model is that it can take quite a while to train as it can require a large number of layers.

Deep learning models have shown impressive results in literature, and choosing models from that perspective make it a tempting choice. The downside of neural networks is that they take a lot of time to train, and also takes a lot of time to set the parameters correctly, while there is no guarantee of better performance. From that perspective, much simpler models such as linear regression but also alternatives such as the lasso are faster to tune and train and can perform better or similar. Another upside is that it is much easier to implement simpler models, let alone maintain. James et al., 2021 point out that it makes sense to always compare more complex models with more simple models so that

a performance/complexity tradeoff can be made. Deep learning models should be chosen when training sets are large and interpretability is not important.

3.5 Discussion of the models

From the deep learning perspective, recurrent neural networks are the most applicable as the process is a sequence. One issue is that neural networks generally require a large number of training instances. This serves as an argument to rule out neural networks as an applicable choice. Other arguments include that training and fine-tuning are complex, and the models are not interpretable.

Model alternatives							
Model	**Training difficulty**	**regularization**	**Variable selection**	**Interpretability**	**Flexibility**	**Complexity**	**Automatic feature interaction**
Linear regression	Low	No	No	High	Low	Low	No
Regularized regression	Low	Yes	Some	High	Low	Low	No
Regression Spline	Med	No	No	Med	Med	Low	No
Smoothing Spline	Med	Yes	No	Med	Med	Low	No
Mars	Med	Yes	Yes	Med	High	Med	Yes
GLM	Med	Yes	No	Med	Med	High	No
XGBoost	Med	Yes	Yes	Med	High	Med	Yes
LightGBM	Med	Yes	Yes	Med	High	Med	Yes
SVR	High	Yes	No	Low	Med	High	No
Deep learning	High	Yes	Yes	Low	High	High	Yes

Table 3.1: Alternative model options listed with their characteristics

In table 3.1 we list the models that are present in literature and their characterizations. The levels for the characteristics are based on the discussion of the models in this chapter, but also on Kuhn et al., 2013 who present a similar table. We explicitly do not include a predictive performance category, even though we have seen statements that some models can perform well. The predictive performance is dependent on the situation and the model tuning. Methods can perform equally well on one problem and there can be a difference on the other. It is our perception that Deep learning models are too complex, difficult to tune and not really applicable to our situation due to the high complexity and the number of training instances we have. We have also learned from Westerink, 2021

that a simple linear model was not a good fit to the problem due to nonlinear relations between predictors. Eventually the model should be applied in a more broad, automated scheduling system in which we do not expect the agents using the system to be able to mirror the results. As a result, interpretability is of less importance.

Boosting models are an appealing choice due to that there is an abundance of evidence that these models perform well. Another advantage of the boosting models is that they have been developed for usage in production environments (the operational environment where the model is actually used by the end user, e.g. the service agent), therefore require relatively little extra work to implement. They are from that perspective easy to train when compared to for instance a neural network. The difficulty is in the parameter tuning. Finally, the models automatically interact the available predictors, which is a convenient extra. The MARS model also has the option to interact the predictors. Mars model is very well covered in literature, but the approach somewhat lesser present in production environments. The model shares other characteristics with boosting models such as that the training difficulty is in the parameter tuning and the model is more flexible. Therefore we fit a MARS model as well, also to serve as a benchmark for the boosting models. An answer to the research question 2b is then that the suitable models are the LightGBM model, the XGBoost model and the MARS model.

3.6 Conclusion

In this chapter we used literature to find the answer to the research question 2a and 2b

We summarize the chapter with the research questions and the corresponding answers.

- 2a) Which methodologies for implementing prediction models are present in literature? There are many methodologies that all share the same elements and also list the aspects that are key to the successful implementation of a prediction model. These include dealing with the bias-variance tradeoff using cross-validation in order for models to generalize well, handling specifically typed attributes, transformation and scaling to enhance model predictive accuracy, using model selection and regularization approaches to increase model interpretability and to handle the curse of dimensionality and the use of cross-validated hyperparameter tuning and making use of ensemble methods for models to fit well.
- 2b) Which methods for predicting utilization are present in literature and are suitable? There are many models, with many characteristics. The multivariate adaptive regression splines, Light gradient boosting and Extreme gradient boosting models give us three different type of models to test. The three models can handle predictor interaction, are flexible and the difficulty is in the parameter tuning. The Light gradient boosting and Extreme gradient boosting models have been developed for usage in production environments and require relatively little extra work to implement. The Mars model is not that much adapted to production environments, but is included as it can be used as a benchmark for the two boosting models.

4 Proposed Model

In this chapter we train and test the models we selected in section 3.6. We discuss the predictors we have constructed and regard relevant. We aim to find the best performing predictors and model. These are to be used and validated in chapter 5. This chapter consists of several sections. Section 4.1 explains how session utilization is predicted. In section 4.4 we implement the models and discuss the relevant implementations. We also compare the model performance with the predictors used by Westerink, 2021 as an assessment of the predictors.

4.1 Prediction of utilization

We aim to predict the session utilization at the time of scheduling the appointment. That is the moment that either the patient or the service agent call each other. We do this by using supervised machine learning. As explained in section 3.3, in the supervised machine learning approach, we train a model by estimating model parameters using a set of predictors on one side, and the observed result also known as labels, on the other, such that a prediction error metric is minimized. However, the predictors are at the level of the appointment, and the session utilization (the label) is at the level of the session. Recall that the session is the combination of a date, practitioner and location. Since multiple appointments together form the utilization of one session, there is a dependency among the predictors. This is a resource of data leakage.

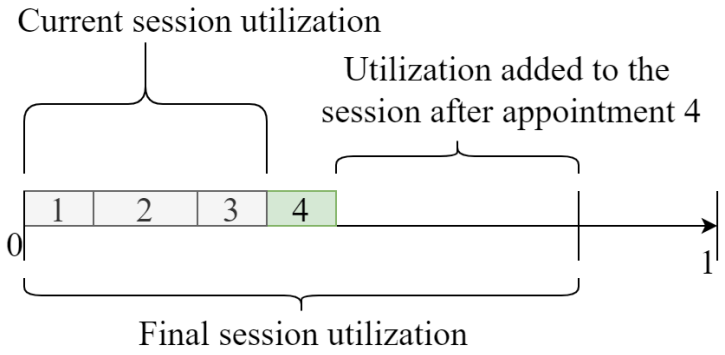

Figure 4.1: Example of utilization added after appointment 4

We resolve that issue by instead having the model predict the expected utilization that is added to the session after scheduling this appointment. Figure 4.1 depicts an example. In figure 4.1, if we would add the appointment for the currently calling patient to this session, that would be appointment 4. Since there are three earlier appointments, if the session would take place there and then, the utilization of the session would be formed by the three appointments. This is the state of the session. We know appointment 4 was added to the session, and the session eventually had a utilization of level X, formed by more appointments. So after appointment 4, the utilization that was added to the session was that of the final session utilization less the current utilization and the utilization contribution of appointment 4. We know the last two terms. We aim to predict the first term. We compute a prediction of session utilization by adding the current utilization, at the time of the appointment, the contribution of the appointment (appointment 4 in the example) and the expected added utilization that we predict using the model.

Since the added utilization is only defined on the unit interval as well, we can still use the performance measures as described in 3.3.3.

4.2 Predictor selection

In this section we study the predictors that can be relevant for the models we aim to use. Not only do we discuss predictors that we deem significant, but we also discuss how the predictors can be used in the scheduling system.

Since the model should eventually be implemented and used in a broader scheduling framework, we have to remember that we cannot use predictors that are not available at the time of prediction. We take the perspective from the incoming patient appointment. We elaborate on this.

First of all, we can subdivide the predictors into three groups. The first group consists of static predictors. These predictors are static in the sense that the predictor itself do not depend on the other appointments or sessions. The predictors are easy to retrieve, and are generally simple measures. The second group consist of the stateful predictors. These predictors depend on the state of the session schedule at the time of the appointment, and thereby on other appointments. Examples are the current number of appointments in the session and the current utilization of the session. The third category consist of aggregations. Aggregations are combinations of multiple appointments to a specific aggregate such the session or location. Aggregations may consist of both static or stateful predictors. For example the average number of patients at a session, or the average travel time of patients in a region. Other types of predictors that we do not specifically categorize are lagged predictors and external predictors. A lagged predictor becomes available at a later moment in time and an external predictor does not originate from within the organization, but is collected externally.

Finally, recall that when a patient calls, a list of appointments is retrieved from the Topsis algorithm discussed in Westerink, 2021. The list is ordered by how good the appointment is, in descending order. These appointments require a prediction of utilization. The appointments differ in date, time, location and practitioner. The sessions, defined by the combination of date, location and practitioner also posses different attributes.

4.2.1 Static predictors

For the static predictors, we start with session information. A session may be longer than another, meaning that more time needs to be filled. Other sessions may currently have a large number of short appointments or a large number of interruptions. The time until the appointment can also differ. A longer period of time until the appointment means that the session also has a longer period of time left during which it can be filled with appointments. This can therefore result in a larger increase in utilization as opposed to when the time until the session is shorter. Since the session takes place on a specified date, we construct predictors based on time as well. These include the year, quarter, month, week, day of the week, year-month and year-week. Where the month and week are numbers in the region of 1 to 12 and 1 to 53, the year-month and year-week also includes the year. For instance, March 2019 has the value $2019 + 3/12$. The intended use of these variables is to account for increases and decreases in utilization over time as observed in section 2.3.2. Also part of the session are the employee, the location and

the region. The organization knows that some locations are more crowded than others, and thereby we know that this differs over the regions as well. We have also observed this in section 2.3.2. Then there is the variable indicating whether or not the session is for an odd week, and the predictor indicating whether or not the session is only for a single day or not. The idea behind the latter is that a single day session is from time to time scheduled in order to accommodate for other than usual activities, for instance visiting a nursing home to see many patients. This can be responsible for a part of the variance in utilization on single days. Finally, the starting times and finishing times of the schedules are included. Schedules that start early or finish late might have fewer patients as in general patients might not want to visit the location later in the evening for instance.

In consultation with the organization we learn that interruptions and breaks are always scheduled before the start of our prediction horizon. That means that the predictor is available at the time of prediction. This is also the case for non-patientbound appointments. Therefore, we can directly use predictors based on these. These are the number of, the total interruption, break and non-patientbound-appointment time, and the ratio of these respective to the session length.

Since we have seen that holiday periods influence the practice time, the total session time and to a smaller extent, utilization. We add the known holiday periods to the data sets. In the Netherlands, holiday periods are distributed based on three geographically defined regions north, middle and south. However, this is for some of the holiday periods, but not for all. The period around Christmas and the holiday around May are in the same period for all regions. The holiday periods Autumn and Spring, both one week long, are distributed over two periods that are given to north, middle and south, alternating per year. That means that two regions have the same week, and one region has its own. In the summer period, which generally lasts for 6 weeks, every holiday period starts and ends one week after the other. Whether a region starts, ends or is in the middle also alternates each year. We expect the highest effect on utilization during overlapping holiday periods. We can construct a predictor that accounts for this by modelling a vacation-factor for one region with a one, when two regions have a holiday we can use a two, and when three holiday periods are active we use a three.

In the Netherlands, there are days that are not public holidays, but are nonetheless important to the Dutch. Many of the Dutch take into account these days in their calendars.

- 4th of May: The dead are remembered, specifically those who died during the second world war. This is regarded an important event for all of the Dutch. Another reason why this date is special is due to that it is one week after Kings day, which is a public holiday and hence, by the pattern observed in section 2.2.1 is expected to show a lower utilization.
- 5th of May: This is liberation day in the Netherlands. The Dutch celebrate the liberation after being occupied by Germany in the second world war. This is a public holiday, but not necessarily a day of. This often depends on the collective agreement.
- 5th of December: In the evening of the 5th of December, many Dutch families celebrate the feast of Sinterklaas, a public holiday for kids.

- 24th of December: This is Christmas eve. Although this is not a public holiday, many choose to take the day off.
- 31st of December: Similar to Christmas eve, the last day of the year is not a public holiday, yet many choose to take the day off.

The organization holds a list of days that are public holidays for all employees. By the pattern observed in section 2.2.1, appointments that are scheduled 7 and 14 days later are demarcated with a 1 to indicate that their utilization might be influenced by the special day the week before. Also part of the list is a conference organized by the organization and heavily visited by the practitioners. This is included as well.

Another group of static predictors we have at our disposal are patient-level characteristics. These predictors can act as sources of dis-utilization, due to that a source of dis-utilization is a no-show appointment, and the no-show can be more prevalent for patients characteristic such as the age or gender of a patient.

We also have access to the patient postal codes. Patients that live further away from the location can experience heavy traffic or it can be more difficult to fit the appointment within their schedule, which can lead to dis-utilization. We can use the postal code to construct a degree of travel time towards a location. We find the coordinates corresponding to the postal code. The coordinates of the locations are already available. We convert the coordinates to northings and eastings. northings and eastings are basically a transformation from coordinates to kilometers. We can use the northings and eastings in a distance function. A parameterized and generalized distance function is the Minkowski metric:

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

In which $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_n$ are the northings and eastings of both the location and the patient.

The parametrization allows the function to return distances that are to a degree a Manhattan distance ($p = 1$), a Euclidean distance ($p = 2$) or another type of distance. This allows the resulting distances to approach the actual travel distance, which is never a straight line like the Euclidean distance and underestimates the actual distance, but also not a right-angle such as the Manhattan distance, which overestimates the actual distance. X. Wang et al., 2013 show that $p = 1.31$ provides the best approximation of travel distance in semi-urban areas. Finally, travel time is the distance divided by speed. We assume that an average speed of 40km/h is a reasonable choice when we find that the average travel distance is 4.88 km (n=536180).

Since the largest share of patients is, or is approaching old age, another source of dis-utilization can reside in illness. We have already seen the effect of covid-19 measures on the utilization. Another source of illness is the flu. The Dutch National Institute for Public Health and the Environment (RIVM) reports the influenza virus detections on a weekly (Wednesday) basis to the World Health Organization (Rijksinstituut voor Volksgezondheid en Milieu, 2021). The influenza detections can be retrieved from the Global Influenza Surveillance and Response System (GISRS) (World Health Organization,

2021). We add the last reported number of influenza values that was available at the scheduling date as a predictor.

4.2.2 Stateful predictors

The stateful predictors depend on other appointments in the session, or other sessions in general. This has its implications on the way the predictors are retrieved. One challenge that arises, is how to handle in case there is no other session or appointment. Options are to exclude the observation, to impute the observation or to not use the predictor.

The first stateful predictor is the current utilization at the time of scheduling. Since we do consider the appointments without patients as productive time and given that these appointments are scheduled at least one month in advance, the time these appointments sum up to, is regarded as the initial utilization. Hence, the first appointment has the initial utilization, plus it's own contribution towards utilization as predictor for the utilization at the time of scheduling.

Secondly, the current access time is used. This is the average access time of the appointments scheduled in the session so far. The initial access time consists of the access time of the first appointment in the session. Another predictor based on the current access time, is the current variance in access time. This is the variance introduces by the difference in access times. The idea behind using the variance in access time, is that during a more busy period, more people call, and an appointment that is available as soon as possible will be offered. This means that the appointment access times are more close together, forming a lower variance in appointment access time.

Two other stateful predictor that we construct are the session utilizations at the time of scheduling, for sessions taking place 7 and 14 days earlier. Many sessions are recurring in the sense that they are often at the same location, by the same employee, during the same hours, but now one week, or two weeks earlier. This should consider the pattern as described in 2.2.1 and is possibly one of the more important predictors. However, the predictor is not defined for all observations. That is since the first occurring session does not have a session taking place before it. We identify four options. 1. We can impute the utilization, for instance with the average utilization in the region at the same day. 2. We can leave the observations out. This way we do not consider pretty much all single work schedules, and we might be making the problem simpler than it is. 3. we can construct and fit a model for which the predictor is defined and to create a model for which it is not. 4. We do not consider the predictor, leaving us with the same observations as before.

Another issue is that if we construct the utilizations and access time predictors for the session, the location, the region and nationwide. We take the perspective of 7 and 14 days before the session date. That brings up the problem that the date of the preceding session might occur later than the appointment request date. Therefore, to have a representative state, we use the predictors as determined for the closest appointment to the request date. This way we are able to retrieve the utilization and access time predictors for the session preceding the appointment date given the request date.

4.2.3 Aggregation

We also use aggregated predictors. These predictors are an aggregate over the location, region and the nation. The idea behind aggregated predictors is to be able to find characteristics of utilization and dis-utilization, that are not shown in the dis-aggregation.

We construct aggregates of patients characteristics. The first predictor is the average age per session, location, region and nation. An example of the influence of an aggregate predictor can be observed with older patients. Older patients often require relatives to bring them to the appointment. Therefore the average age of the patient can be a predictor for dis-utilization, as the dependency on others increases the probability that one of the two cannot come to the appointment. The larger the average age in a session, the more likely this is.

A second aggregation is the travel time for patients. This is similar to patient age. The idea behind travel time is that a larger distance to the appointment location implies that the patient is more likely to heavy traffic and or personal schedule issues that lead to the patient not being able to adhere to the scheduled appointment. We take the average to normalize over the number of patients seen. We aggregate the number over the location, region and nation.

Other aggregates that we use are the location, region an nation utilization on the session date. the location, region and nation average access time on the session date, the location region and nation number of interruptions and number of interruption hours and the total session time on an aggregation level. One of the advantages of using these predictors is that they allow for a comparison with the aggregate level. As an example, a large current utilization in a region may imply that one of locations with a sub par current utilization is likely to observe more appointment requests.

Appendix E contains a list of all predictors and a corresponding short description.

4.3 Data preparation

This section discusses the data preparation steps discussed in section 3.3. Transformation, scaling, handling specifically typed attributes and the cross-validation are discussed.

4.3.1 Scaling

Tree based methods as well as the MARS model are invariant to the locations and scales of the predictor in terms of predictive performance (Friedman, 1991). In other words, scaling is not required.

4.3.2 Handling specifically typed attributes

Predictors that are categorical need to be treated with special caution. Categorical predictors can be subdivided into three groups. The first group is ordinal data. These are variables of which the categories have order, like temperature (e.g. warm, neutral, cold). Ordinal variables can be label encoded. Our data does not show ordinal predictors. The second group is nominal data. These are variables with categories that are not following an order. The nominal predictors are

- Employee id

- Location id
- Region id

We can use one-hot encoding for these predictors, but as the predictors consists of many categories, this will increase the number of predictors with about 400 variables. According to the LightGBM documentation (Microsoft Corporation, 2021), it is best to use a contiguous range of integers started from zero for the LightGBM model. Hence, we label-encode the predictors the LightGBM categories. The XGBoost documentation mentions a similar feature, but states that the feature is still experimental and should only be used for development purposes (XGBoost developers, 2022). Therefore, for XGBoost and MARS we use one-hot encoding.

The third group is dichotomous data. Dichotomous variables consist of two categories, generally yes or no. The dichotomous predictors are:

- is single work schedule
- is odd week
- is Christmas eve
- is last day of the year
- is dead remembrance day
- is liberation day
- was liberation day 7 days ago
- was Kingsday 14 days ago
- was public holiday 7, 14 days ago

These predictors can be included with one-hot encoding relatively easy, as every of these will only add one predictor to the data.

4.3.3 Cross validation

To evaluate the model performance, we first split the full data set in two parts. One part of the data is used for training, the other part is used for testing. A 80% train / 20% test split is a well-known rule of thumb as long as there are enough observations to train and test on. Since we have 161908 observations, this leads to a (129526, 32382) split, which should be enough for both and also leaves enough observations per cross validation set. The idea is that the test data has never seen the model before testing, so that the measured performance was not due to that the model tries to fit to the test data. There are multiple approaches for splitting, and the approach matters.

The first approach is to randomly split the full data set on the appointments. This approach is the easiest, but has it's downside. Since often multiple appointments together form the one session, there is no guarantee that data in the test set also have appointments from the same session in the training set.

The second approach is to split chronologically. We could train on the sessions until 2019, and test on the sessions taking place in 2019. This way, we ensure that the appointments that were used for training are not used for testing. However, the downside is that we

do not allow the model to obtain valuable information that could have been present in the 2019 sessions. An example are the work schedules that only take place in 2019 for employees or locations that started in 2019.

A third approach is to randomly split on the sessions that have corresponding appointments in the data. This way we guarantee that the sessions corresponding to appointments that were used for training are not used for testing, but we still allow the model to learn from data in 2019. The downside is that we nevertheless do not have a guarantee of that every location and employee in the test data is also present in the train data.

Since the approach described in 4.1 resolves the issue mentioned in the first approach, we choose to use the first approach, since it is the easiest.

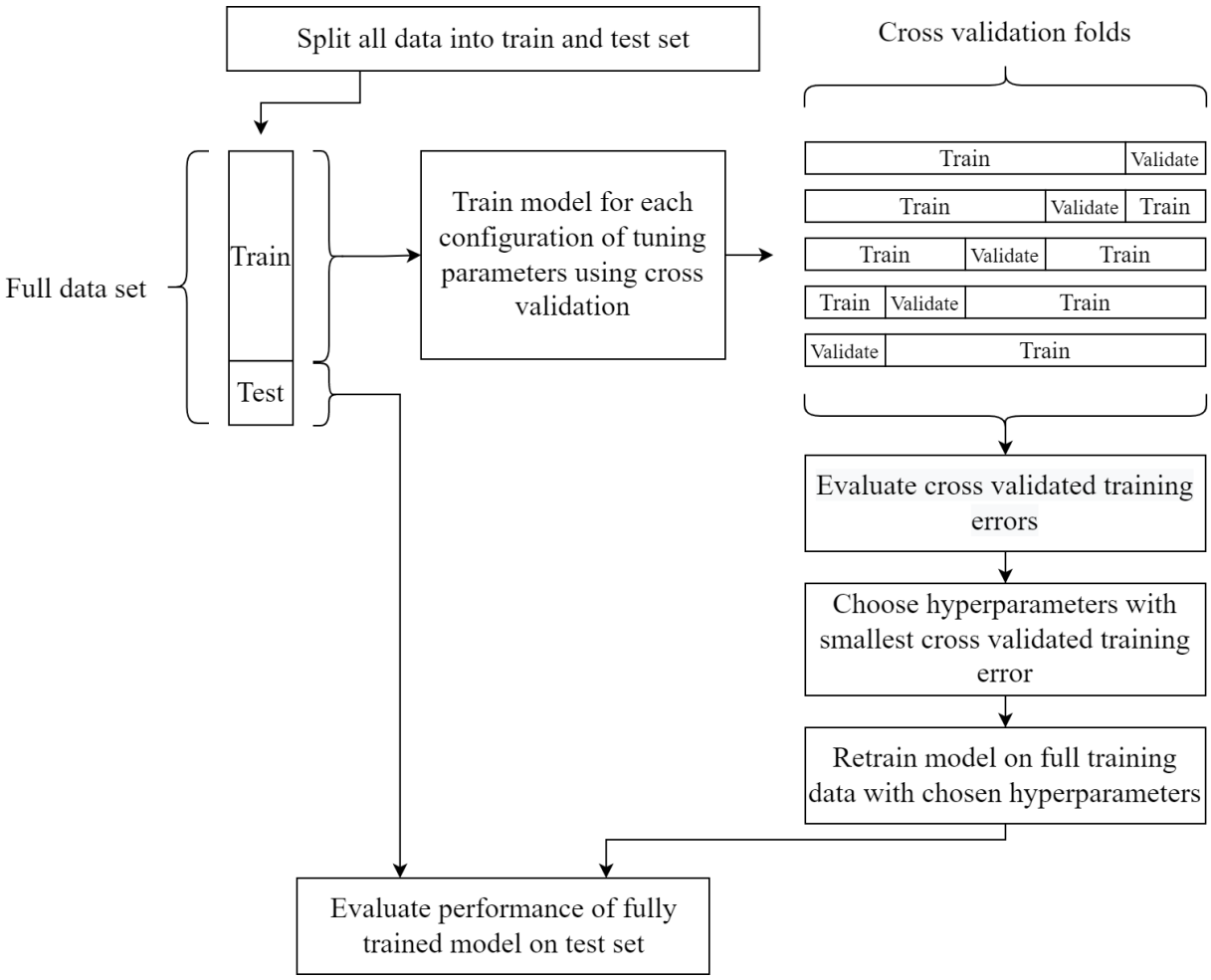

Figure 4.2: Training, validating and testing approach

Figure 4.2 shows the training, validation and testing approach visually. After splitting on the test and the training set, we fit the model on the training data. We use 10 fold cross validation for this as a tradeoff between having enough training instances per fold. Yet have enough data to validate the performance and prevent overfitting. We perform the cross validation process many times for each model. During each iteration, we test a different set of hyperparameters. We use the mean of the cross validated root mean squared error as a measure of performance of the model trained by the configuration of parameters. We use the Tree-structured Parzen Estimator (TPE) to suggest parameter

options over the iterations of the model building as a tradeoff between evaluating a large number of parameters, which helps identifying a promising configuration of parameters, and on the other hand to reduce the necessary training time. As a guideline, the parameters are given a range of values in which they may lie. In section 4.4, for each of the models, the parameters as well as the corresponding ranges are specified.

After the cross validation approach, the trained model is evaluated on the test set.

4.4 Model performance

In this section we discuss the model tuning and performance. We use the three models discussed in section 3.5 and discuss the most important parameters that the models use. We start with a discussion of the MARS model. Then we discuss the implementation of the XGBoost model. We end with an evaluation of the LightGBM model. All numerical experiments are performed on a machine with 128GB of RAM and 2 Intel® Xeon® E5-2680 v2 processors, providing 40 threads at 2.8 GHz.

4.4.1 Multivariate adaptive regression splines

Recall that the MARS model fits multiple regression models to different sections of the domain of a predictor, and ties these together with hinge functions on the knots. An increase in the number of knots increases the flexibility of the model. The model keeps training until many knots are found. This is known as the forward pass and can be time consuming. As the resulting model is prone to overfitting, the model removes knots that are not significantly beneficial. This is done by pruning and known as the backward pass and is generally faster than the forward pass.

To control the model training speed, the forward stepping threshold parameter can be altered, as well as the pruning method. The exhaustive method is time consuming, and the backward or cross validation methods are alternatives.

The degree of interaction specifies how many predictors the model will interact with each other at most. In Boehmke et al., 2019 it is mentioned that there is hardly any gain in using interactions larger than the third degree.

The minimum number of observations between knots and minimum number of observations between the first and last observation are parameters that control the training speed as fewer knots will be fitted. These parameters can be determined based on the number of predictors and the number of observation at which these predictors are nonzero. The parameters are called minspan alpha and endspan alpha respectively.

The maximum number of model terms before pruning and the maximum number of terms in the pruned model controls how many predictor terms will eventually be in the model before and after pruning. These parameters can also be used to train faster.

The penalty is similar to the the regularization parameter in for instance regularized regression. By the penalty parameter, models with more knots, generally seen as more complex models, are penalized more than models with fewer knots, which helps preventing overfitting.

As preliminary runs of the MARS model resulted in unacceptably long training times, the method described by Friedman, 1993 adds valuable parameters for increasing the speed

Parameter	Value / Value range	Best value
Nr of predictors in final model	[1, 100]	22
Interaction degree	[1, 3]	9
Penalty	[-1, 4]	9.819
Fast k	[1, 5]	3
Fast h	[1, 5]	5
Minspan alpha	[0, 1]	1
Endspan alpha	[0, 1]	0.771
Maximum nr of terms before pruning	[3, 200]	56
Forward stepping threshold	0.001	0.001

Table 4.1: MARS training parameter scheme with best found parameters.

of training, at the cost of predictive quality. These parameters are the rate at which optimization over input variables is performed and the number of parent terms (hinge functions) considered at each step. We also include these parameters. These parameters are called fast h and fast k respectively.

The best value for each of the mentioned parameters can vary for different underlying prediction problem. Some studies suggest a range of large values for regularization parameters, other studies suggest a lower range of values. After initial runs using a broad range of parameters mentioned in Mars optimization studies (García Nieto et al., 2017, Kartal Koc et al., 2015, Boehmke et al., 2019, Dey et al., 2016), we find a grid of parameter ranges that is likely to include the optimal value for the parameter. The grid of parameters is listed in table 4.1.

We use the TPE method to find suitable values for the parameters mentioned in table 4.1. We train 500 different configurations of parameters. Figure 4.3 depicts the optimization process on the left, with the red line depicting the configuration with the best (lowest)cross validated rmse value. On the right is the relative hyperparameter importance

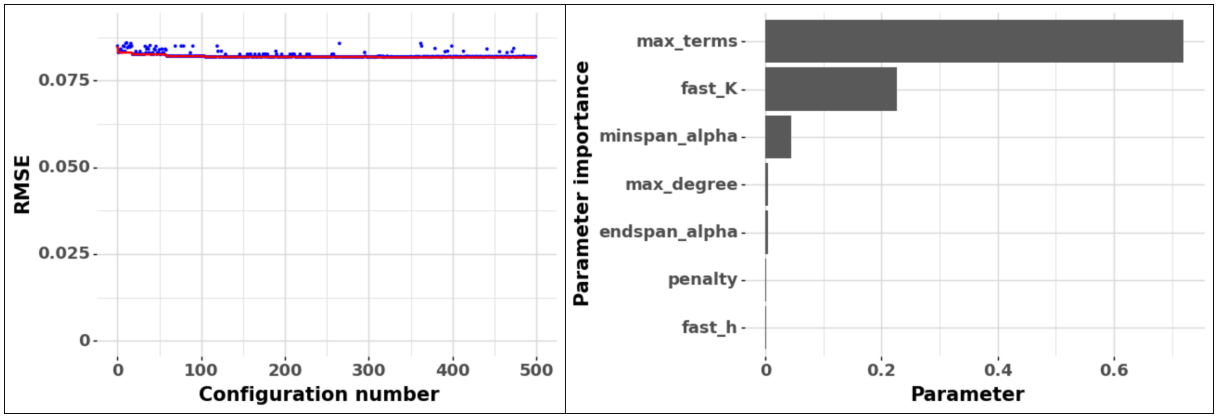

Figure 4.3: (Left) Mars optimization history. (Right) relative hyperparameter importance

Evaluating all 500 iterations takes 7 hours. The configuration with the lowest rmse value has a mean cross validated rmse value of 0.082. The corresponding parameter configuration can be read in table 4.1. An overview of the results of the tuning process can be

found in appendix F. We then retrain the model again using these parameters on the full training set. Recall that the model has never seen these observations. We predict values using the test set on the trained model and compare the results with the observed values.

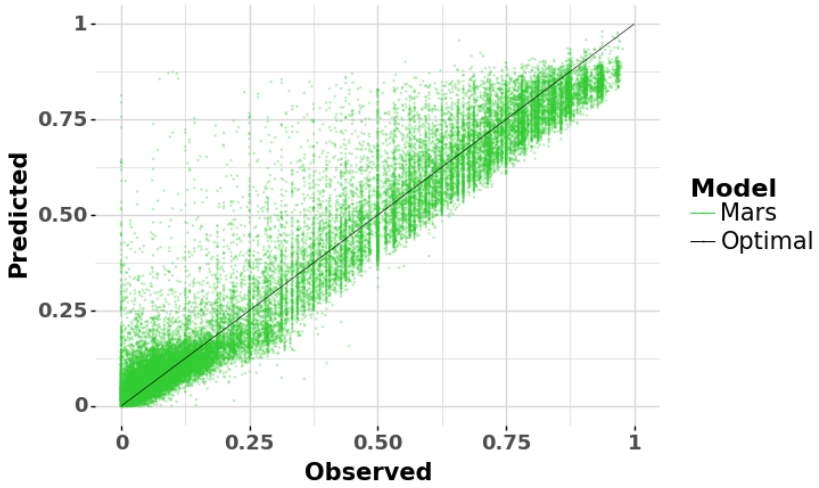

Figure 4.4: Scatter plot with the observed and predicted values, (n=32382)

Figure 4.4 shows the observed and the predicted values of the added utilization. The optimum is on the diagonal. This is the point where the prediction equals the observation.Note that the prediction method can only alter the points in vertical direction. It can be seen that the predictions resemble the line to some extend, but the errors are quite large. The model seems to especially overestimate the added utilization when the added utilization was actually low. We report a test root mean squared prediction error of 0.081 and an R^2 value of 0.929.

4.4.2 Extreme Gradient Boosting

Recall that boosted trees are simply a combination of decision trees. Each XGBoost model consists of a number of steps. In each iteration step, a decision tree is fitted and based on the fit, a split is made. The decision on which split to make is based on an optimization function, which is basically the combination of loss and a regularization term, as with many other machine learning models. The optimization function for the t-th tree is

$$f^{(t)} \approx \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$

The first summation is the loss function and the second summation is the shrinkage term. As the XGBoost model is aiming to find the largest gain in performance in each iteration, the performance of the model (corresponding to the first summation, but without going further into this) is bound to the parameter γ in the sense that the gain in the iteration should at least be of size gamma to make a further partition on the corresponding leaf node. The larger the size of gamma, the more conservative the model is.

In our specific case the loss function is defined by the mean squared error. The regularization term, which helps reducing overfitting, is controlled by a parameter of the model.

This parameter, defined on the unit interval, is called the learning rate. Every step, the parameter shrinks the, in that iteration, obtained split values. In general, the lower the learning rate, the better the model deals with overfitting. One downside to implementing a lower learning rate, is that the number of steps required to reach the same result needs to be higher, which makes training the model slower. The parameters α and λ are the l_1 and l_2 regularization terms respectively.

Setting a maximum on the depth-level of the tree is another parameter that aims to reduce overfitting since it reduces the complexity of the model. Another parameter is the minimum child weight. This is the sum of the number of instances in a leaf node needed to allow for a partition in that node. If the sum is lower than the minimum child weight, then there is no partitioning in that leaf. Thus, a larger minimum child weight leads to a more conservative model.

The subsample ratio controls the amount of training instances that is randomly sampled before growing trees in each iterations. The proportion of all features randomly selected in each tree, colsample bytree, is another model parameter, where subsampling takes place for each constructed tree. Two similar parameters on a lower level are colsample bylevel and colsample bynode. These parameters control the ratio of features to randomly select for each level and for each node of the tree respectively.

After initial runs using a broad range of parameters mentioned in XGBoost optimization studies (Xia et al., 2017, Probst et al., 2018, Thomas et al., 2018, Y. Wang et al., 2019, Zhou et al., 2021, Anghel et al., 2018, see appendix G for the listed parameters and corresponding values), we find a grid with parameter ranges that is likely to include the optimal value for the parameter. The grid of parameters is listed in table 4.2.

Parameter	**Value / Value range**	**Best value**
Learning rate	0.01	0.01
Number of iterations	1000	1000
Max depth	[1, 20]	18
Max delta step	$[10^{-3}, 2]$	0.78
Min child weight	[1, 300]	10
Subsample	[0.4, 1]	0.93
Colsample bytree	[0.5, 1]	0.84
Colsample bylevel	[0.5, 1]	0.75
Colsample bynode	[0.5, 1]	0.74
Gamma	$[10^{-8}, 1000]$	8.13×10^{-9}
Lambda (l_2)	$[10^{-8}, 1000]$	219.28
Alpha (l_1)	$[10^{-8}, 1000]$	1.66×10^{-6}

Table 4.2: XGBoost training parameter scheme with best found parameters.

We again use the TPE method to find suitable values for the parameters mentioned in table 4.2, and then retrain the model again using these parameters on the full data set.We train 500 different configurations of parameters. Figure 4.5 depicts the optimization history on the left, with the red line depicting the configuration with the best (lowest) cross validated rmse value. On the right is the relative hyperparameter importance.

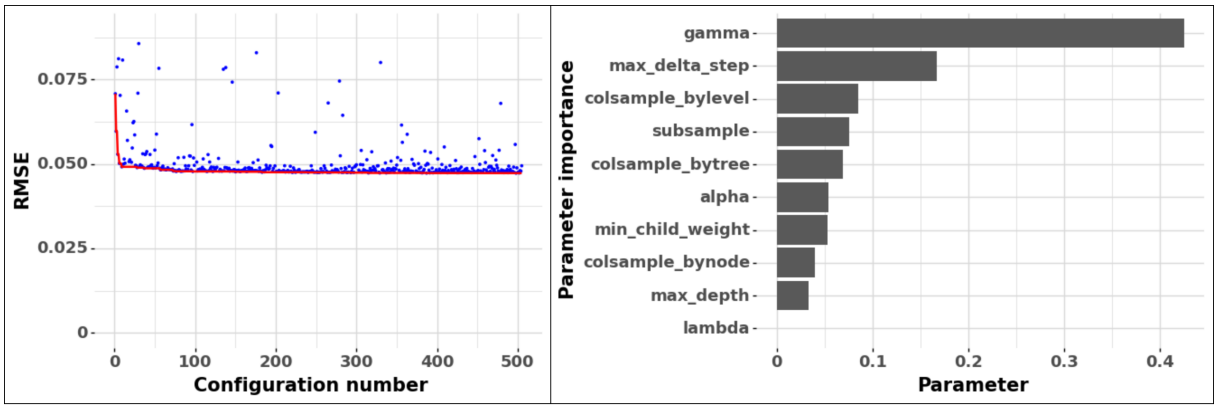

Figure 4.5: (Left) XGBoost optimization history (Right) relative hyperparameter importance

Tuning the parameters took 6 days and 7 hours. The configuration with the lowest rmse value has a mean cross validated rmse value of 0.047. The corresponding parameter configuration can be read in table 7 and the full tuning process can be found in Appendix G.

We retrain a model using the parameter configuration corresponding to the lowest cross validated rmse on the full training set. Then we introduce the test set. Recall that the model has never seen the records in the test set. We let the model predict using the predictors in the test set. We compare the predictions with the actually observed values. Figure 4.6 depicts the result.

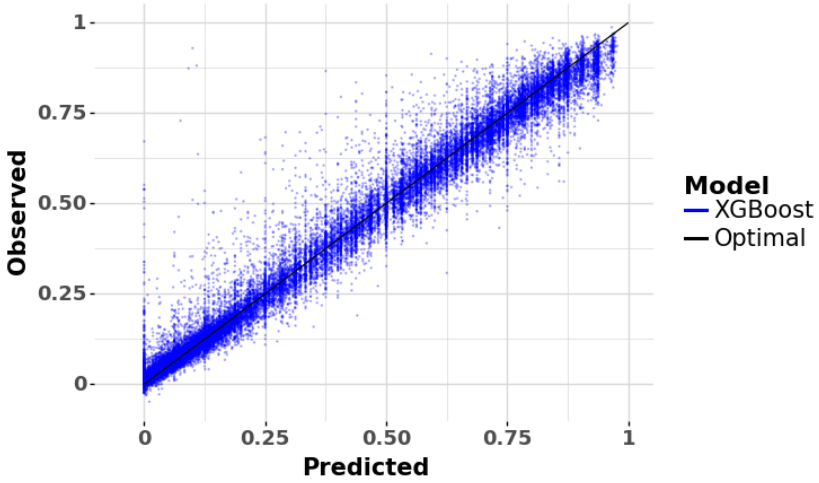

Figure 4.6: Scatter plot with the observed and predicted values, (n=32382)

Again, in figure 4.6, the optimum is on the line where the predicted value equals the observed value. Figure 4.6 resembles that line to a better extend than the Mars model.The results do not show the positive skew observed in the Mars predictions. We report a test root mean squared prediction error of 0.048 and an R^2 value of 0.975, which shows evidence that the model is producing better results than the Mars model.

4.4.3 Light Gradient Boosting

Recall that the aim of LightGBM models is generally to find only slightly worse solutions than the XGBoost model, but far more quickly. The model grows leaf-wise instead of level-wise. That means that trees are split on the nodes with the largest change instead of nodes that are close to the tree. The downside of leaf-wise growth is that it can cause overfitting with a small data set. The LightGBM model can be constrained with a maximum depth parameter that controls the maximum tree depth. This doesn't prevent the model to grow leaf-wise. This is controlled by the maximum 'number of leaves' parameter. This parameter controls the maximum number of leaves in one tree.

In LightGBM, the continuous predictors are bucketed into discrete bins, comparable with a histogram. This has the advantage that calculations based on the bin instead of the continuous variable is much faster. The number of bins can be specified as a parameter to LightGBM. By that parameter, the training speed can be controlled.

The model requires categorical predictors not to be one-hot encoded, as he predictors can be unbalanced in the sense that one of the categorical instances - say a specific employee - is underrepresented in the data set. Therefore the LightGBM models split categorical predictors into two subsets such that they are split optimally using the method presented by Fisher, 1958.

The most important parameters that are present in XGBoost are also present in Light-GBM. The learning rate as well as the number of iterations are present. The minimum bagging fraction and the feature fraction are the same as the subsample ratio and the colsample bytree parameter of XGBoost respectively. The minimal sum of the hessian in one leaf is the same as the minimum child weight parameter in XGBoost.

After initial runs using a broad range of parameters mentioned in LightGBM optimization studies (Anghel et al., 2018, Dev et al., 2019, Zhang et al., 2019, Massaoudi et al., 2021, J. Wang et al., 2018, Tang et al., 2020, see appendix H for the listed parameters and corresponding values), we find a grid with parameter ranges that is likely to include the optimal value for the parameter. The grid of parameters is listed in table 4.3 . Comparing XGBoost to LightGBM can be done in two ways. We can either train with the same amount of resources, and leave the parameters free. This gives a degree of how efficient the models train. A second option is to use the same parameters. This gives a degree of how good the predictive quality of the model is. We are more interested in predictive quality, therefore we aim to use roughly the same parameters for the LightGBM model, especially for the learning rate and the number of iterations.

Parameter	Value / Value range	Best value
Learning rate	0.01	0.01
Number of iterations	1000	1000
Number of leaves	[20, 600]	292
Max depth	[2, 20]	9
Min data in leaf	[2, 500]	10
Min child weight	$[10^{-3}, 1]$	0.268
Subsample	[0.5, 1]	0.870
Subsample frequency	[0, 15]	1
Feature fraction	[0.5, 1]	0.976
Feature fraction bynode	[0.5, 1]	0.982
Lambda (l_1)	$[10^{-8}, 10]$	2.333 x 10^{-7}
Lambda (l_2)	$[10^{-8}, 10]$	1.610

Table 4.3: LightGBM training parameter scheme with best found parameters.

Also this time we use the TPE method to find suitable values for the parameters mentioned in table 4.3. Then we retrain the model again using these parameters on the full data set. We train 500 different configurations of parameters. Figure 4.7 depicts the optimization history on the left, with the red line depicting the configuration with the best (lowest) cross validated rmse value. On the right is the relative hyperparameter importance.

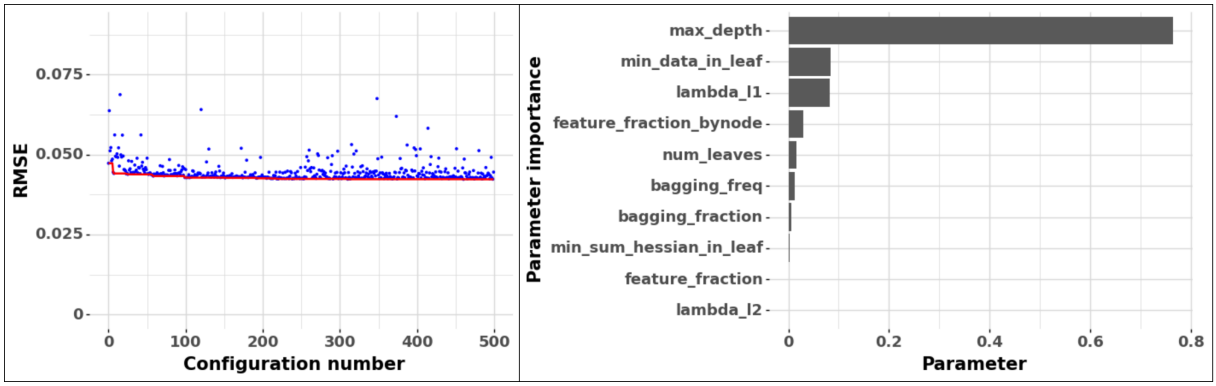

Figure 4.7: (Left) LightGBM optimization history. (Right) relative hyperparameter importance

Tuning the parameters took 12 hours. The LightGBM model is considerably faster, in our case 12 times. The configuration with the lowest rmse value has a mean cross validated rmse value of 0.042. The corresponding parameter configuration can be read in table 4.3 and the full tuning process can be found in Appendix H.

We retrain a model using the parameter configuration corresponding to the lowest cross validated rmse on the full training set. Then we introduce the test set. Recall that the model has never seen the records in the test set. We let the model predict using the predictors in the test set. We compare the predictions with the actually observed values. Figure 4.8 depicts the result.

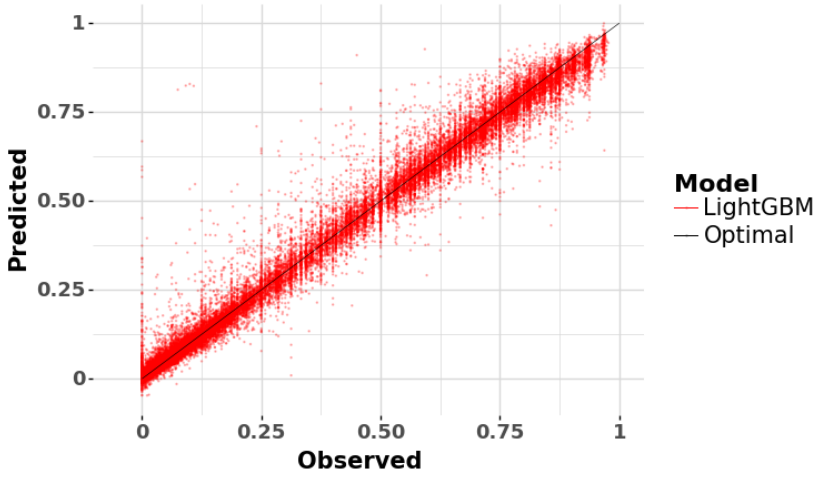

Figure 4.8: Scatter plot with the observed and predicted values, (n=32382)

Also in figure 4.8, the optimum is on the line where the predicted value equals the observed value. Figure 4.8 resembles that line to some extend, arguably to a better extend than the XGBoost model. The errors seem to be less dispersed. The results also do not show the positive skew observed in the Mars predictions. We report a test root mean squared prediction error of 0.041 and an R^2 value of 0.982. These are the best values for the three tested models. In the next section we discuss the differences.

4.4.4 Comparing the proposed models

In this section, we combine the results of fitting the three models, and discuss how the models perform. We do this by comparing the model errors.

Figure 4.9 depicts the densities of the prediction errors following from the three models.The densities are based on 1000 histogram bins. The LightGBM model has the largest value for errors around 0, which is in line with the model having the lowest test-rmse.The XGBoost model performs similarly, with a slightly smaller value around zero. The Mars model not only performs worse, the errors show a bit of a bulge around the error value 0.125 which is hard to explain.

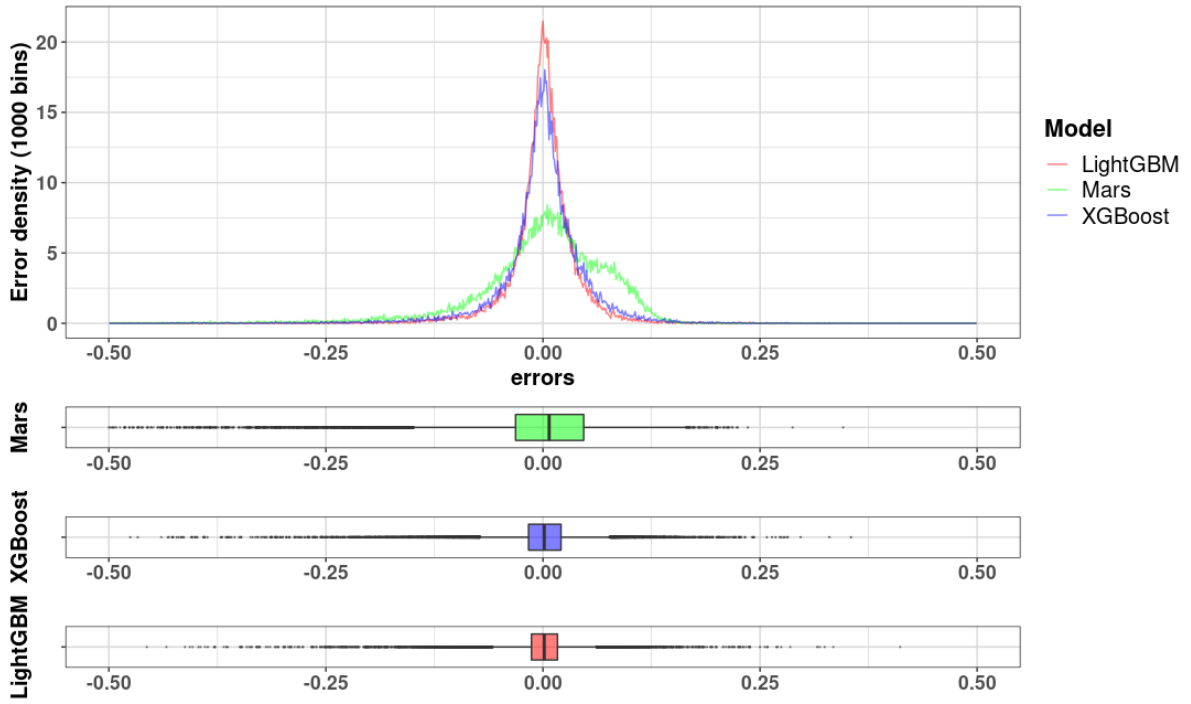

Figure 4.9: Error density for the three proposed models, with corresponding boxplots

According to the lower test-rmse for the LightGBM model compared to the two other models and also seen in figure 4.9, we hypothesize that the LightGBM model produces a lower rmse, and that the LightGBM model produces a lower variance in prediction error than the other two models.

Model	**Abs errors**		**Hypotheses**	**Test statistic**	**P value**
	Mean	SD		t(32381)	
LightGBM	0.025	0.033			
Mars	0.054	0.058	$H_0 : \mu_{LightGBM} = \mu_{Mars}$		
			$H_1 : \mu_{LightGBM} < \mu_{Mars}$	-116.86	$p < .001$
XGBoost	0.031	0.04	$H_0 : \mu_{LightGBM} = \mu_{XGBoost}$		
			$H_1 : \mu_{LightGBM} < \mu_{XGBoost}$	-40.74	$p < .001$

Table 4.4: Statistical tests on the absolute error means

Since errors are scaled around zero, we cannot directly test the hypothesis for the mean.Therefore we test the mean hypothesis on the absolute error. The tests are shown in table 4.4. In comparison with the LightGBM model, we reject the null hypothesis of equal absolute prediction error means between the LightGBM and the Mars model as well as between the LightGBM and the XGBoost model. The absolute errors are show in figure 4.10.

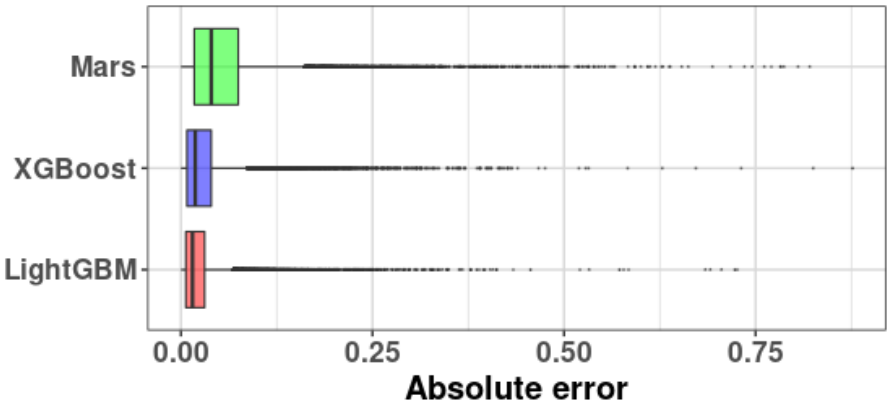

Figure 4.10: Boxplots of the absolute prediction errors for each of the models

We continue with a test on the variance. For a test on the variance between the methods, we can directly use the errors (as seen in figure 4.9). There is no evidence that the errors are normally distributed, hence we use the Brown–Forsythe test. The Brown–Forsythe test is recommended due to its robustness against non-normal data (Derrick et al., 2018). In comparison with the LightGBM model we reject the null hypothesis of equal variances between the LightGBM model and the Mars model as well as between the LightGBM model and the XGBoost model. The test results are shown in table 4.5

Model	**Errors**		**Hypotheses**	**Test statistic**	**P value**
	Mean	SD		F(1, 64761)	
LightGBM	0.0002	0.0417			
Mars	0.0009	0.0793	$H_0 : \sigma^2_{LightGBM} = \sigma^2_{Mars}$		
			$H_1 : \sigma^2_{LightGBM} < \sigma^2_{Mars}$	6035.70	$p < .001$
XGBoost	0.0004	0.0505	$H_0 : \sigma^2_{LightGBM} = \sigma^2_{XGBoost}$		
			$H_1 : \sigma^2_{LightGBM} < \sigma^2_{XGBoost}$	462.39	$p < .001$

Table 4.5: Statistical tests on the prediction error variances

Therefore we conclude that the LightGBM model produces lower prediction errors, and prediction errors with a lower variance. The LightGBM model is the best choice among the three models and is recommended as option for the organization.

A final issue to address, is that of predictive performance for different levels of observed utilization. It was reported in Westerink, 2021 as well as in consultations with the organization that it was particularly difficult to predict good values of utilization when the observed utilization was low and was one of the reasons for requiring more flexible models. Given that the three proposed models sport flexibility as well as regularization options, the errors resulting from the predictions should not have much difficulty predicting different levels of added utilization. Figure 4.11 shows that this is indeed the case for the XGBoost and LightGBM model, but the Mars model has more difficulties with prediction for different levels of utilization, which was also reflected in figure 4.4 and the error density plot of figure 4.9.

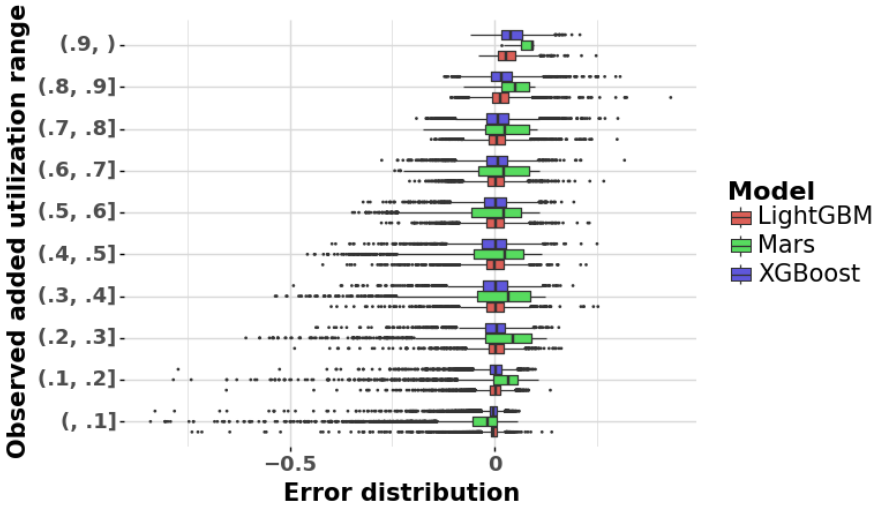

Figure 4.11: Error distribution for each of the models, grouped by intervals of size 0.1

Figure 4.11 shows the prediction error distributions for different levels of observed added utilization. What is especially interesting to see, is that the LightGBM model seems to be able to predict added utilizations that are quite large better than the XGBoost models.For instance when the added utilization was more than .9. Note that an added utilization of .9 means that the current utilization level is at most .9. That is also one of the reasons for that the outlier range becomes smaller when the observed added utilization range becomes larger. The second reason is that the number of observations in the groups with smaller added utilizations is larger than the group with large added utilizations, which is also reflected in the mean session utilization of .88.

4.4.5 Comparing the relevance of the predictors

There are multiple ways to identify the important predictors. This is due to that different models use predictors in different ways. The Mars models offer three ways to measure the importance of predictors (Milborrow, 2021). The first option is to counts the number of model subsets that include the predictors. Predictors that are included in more sub models are considered more influential.

The second option is to use the residual sum of squares (RSS).In each backward pass,the reduction in RSS is determined for the subsets of predictors corresponding to the previous subset. Then, for every importance variable corresponding to the predictor, the RSS reduction is added to the variable for every subset that includes the predictor. The resulting set of importance variables is scaled on the unit interval. Larger importance variables are regarded more influential.

The third option is using generalized cross validation (GCV). The GCV measures are also used to evaluate the performance in the backward pass, which make it a natural choice as a measure for feature importance. The GCV was first introduced by Craven et al., 1978 and is an approximation of the RSS, with the extension that it penalizes the flexibility of the model. Using the GCV for predictor importance is similar to using RSS. Instead of using the RSS reduction, the reduction in GCV is used, summed, and scaled to the unit interval.

The Mars model values the current utilization level of the session if the appointment is to be included as by far the most important predictor. Secondly, the total time of appointments without patients is regarded an important predictor, although its influence on the RSS and GCV is relatively limited. Only seven predictors are used in any of the subsets. The model is quite conservative in using a larger number of predictors, which is possibly due to the penalty term (9.82) which is used in the best parameter configuration of the Mars model and primarily penalizes the usage of more predictors. It might also be due to the threshold. If we run an experiment with both a lower and a higher threshold, we find that the model respectively decreases and increases the number of used predictors in the subsets. Setting a lower threshold comes at the cost of requiring considerably more time to fit the model, whilst the predictive quality does not increase by much. If we change the threshold from 0.001 to 0.0001, the test-rmse decreases from 0.081 to 0.079 but the training time increases from 10 minutes to 2 hours. Figure 4.12 shows the most important predictors.

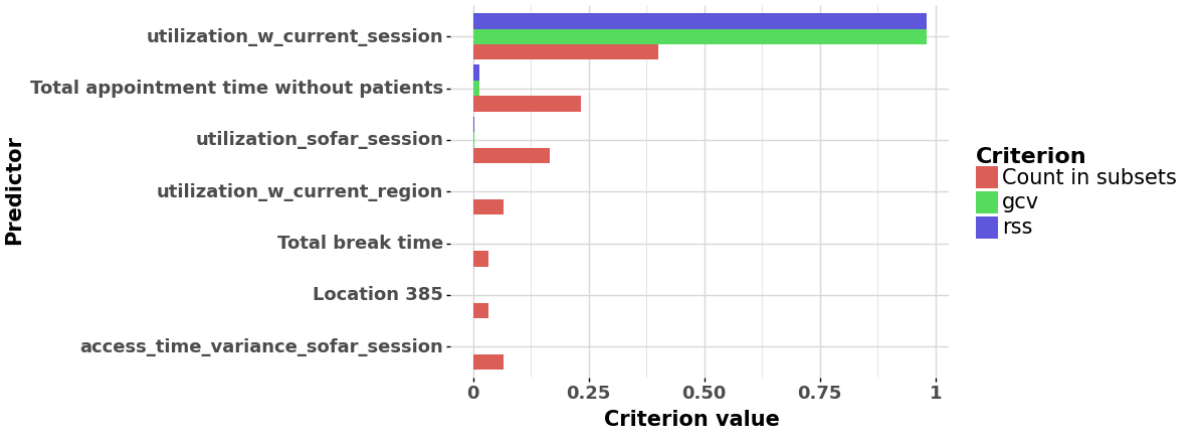

Figure 4.12: Subset count, RSS and GCV for predictors used in Mars model.

In figure 4.12 we clearly see that the current utilization has far more influence than any of the other predictors. The other predictors do contribute, but not nearly as much. We now check to see if the usage of these predictors is consistent with the other models.

For decision trees, there are different options to measure the importance of predictors.One option is to count the number of times that a predictor was split on. Another option is to sum the gain that the predictor contributed when a split was made on the predictor. This is also the criterion on which splits are made in the LightGBM and the XGBoost model algorithm, which makes it a natural choice for comparing the relevance of the predictor within the trained model. The XGBoost model offers a third option named cover. This represents the sum of the number of observations that pass a node corresponding to the predictor. So the predictor cover count increases each time that an observation passes a node that was split on that predictor. We show the results for the XGBoost model in figure 4.13 with for each on the importance criteria the ten best performing predictors.

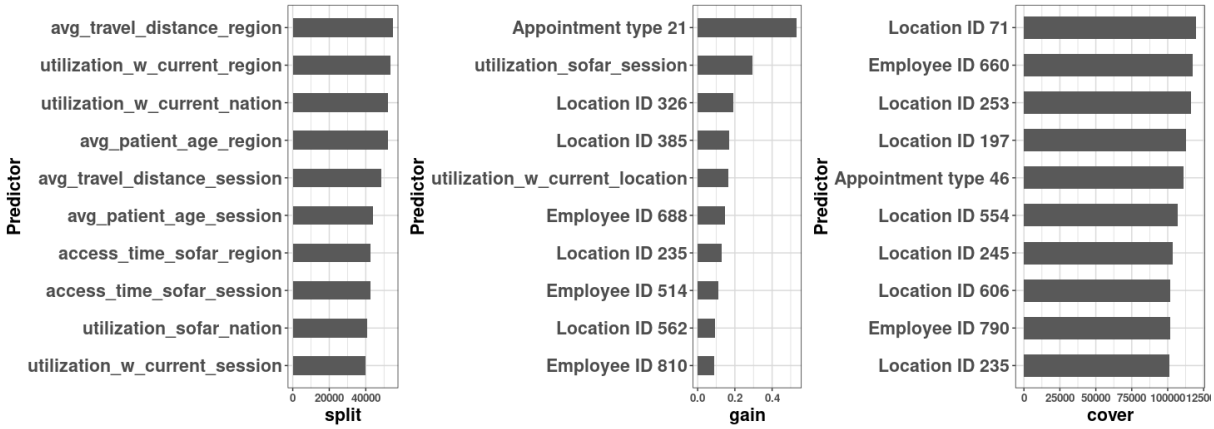

Figure 4.13: (left) Split count of predictors. (Middle) Sum of gain of predictors. (Right) Cover per predictor

Compared to the Mars model, we see a different result. Figure 4.13 shows that the most important predictor in terms of gain is appointment type 21. This is not surprising as we learn that the appointment type is specifically for laser measurements, which require a skilled employee. The model also gains from specific locations. The current level of session utilization is the other important predictor, which is also the case for the Mars model. The XGBoost model also shows to gain from knowledge on which employee is scheduled to. This is also reflected in the cover. Observations are at some point split on whether or not the location has ID 71. One possible reason for this is that the location is only staffed once in two weeks. In terms of on how many nodes a predictor is found in the trained decision tree, we see that distance and age are occurring. Unlike the current utilization predictors, which show larger degrees of gain, age and distance predictors have very limited gain.

Finally, using the LightGBM model we show a comparison of the ten most influential predictors in figure 4.14.

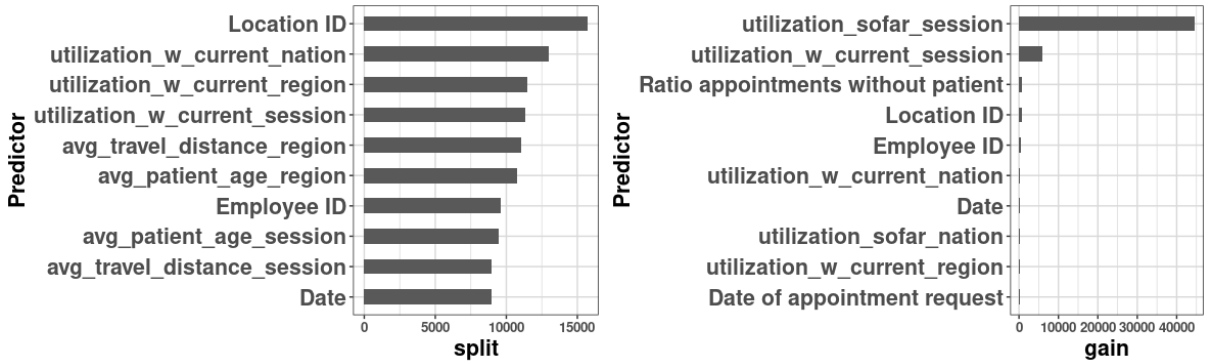

Figure 4.14: (left) Split count of predictors. (Right) Sum of gain of predictors

The right figure in figure 4.14 shows that the current utilization levels of the session (utilization so far session and utilization w current session) are by far the most important predictors. The utilization levels are an important factor for determining the level of utilization that will be added to the session. One possible reason for this can be that there are many higher utilizations levels, and the higher the level of utilization is, the

fewer utilization there is left to be added to the session. Hence, a larger level of utilization leads to a smaller level of added utilization automatically. The left figure in figure 4.14 shows that the trained LightGBM mostly splits on the location of the practitioner. This is perhaps related to the variance in utilization between the locations. A split on location in tree node A is then related to a split on current utilization in tree node B for instance. The ratio of total session time filled with appointments without a patient is another factor that the LightGBM model regards relevant. This predictor has a reducing effect on the level of utilization that can be added. This is possibly the reason for the predictor being relevant. In comparison with the Mars and XGBoost model we see that the LightGBM model gains are much less evenly distributed across the predictors. The predictors themselves are broadly the same.

There are multiple arguments to choose a limited number of predictors from the total group of predictors. First of all, it increases the model's interpretability, as discussed in section 3.3.7. For instance, interactions between predictors are difficult to understand, especially with many predictors and many interactions present. A more practical argument is that retrieving predictors is not always an easy task. It makes implementation of the model in a production environment more difficult. Another practical argument is that it makes training a model less time-expensive. In our model, we have seen that a small set of predictors explain a large part of the variance. The level of utilization of the session is by far the most important predictor. We did already retrieve the other predictors, and trained models on this set of predictors. Secondly, in production, interpretability is of lesser importance than predictive quality, as the output is going to be used as input for another algorithm, which mainly relies on a good prediction of utilization. Interpretability should not limit us in using the available predictors.

We see across the three type of models that consistently, the current level of utilization, the ratio of appointment time scheduled for non-patient activities, the location at which the appointment is at, and the practitioner bound to the appointment are the most important predictors. A practical issue with predictors such as the location or the employee is that new employees and new locations do not have any historical data. In decision trees, the model is then still usable. For example, the employee split on a node on the question whether or not the employee is in the subset of employee IDs X, which is never true for a new employee. Therefore the decision tree will return a general prediction of utilization, based on the other predictors. In this case, an updating procedure is an obvious choice to include the employee or the location at a later moment, when enough data is available. A second option is to consider not using a location or a practitioner as a predictor. Therefore we train the LightGBM model, and compare the results in the 4 cases where both predictors are available, where practitioners are not used as a predictor, where locations are not used as a predictor and finally where both are not used as a predictor. The results are shown in table 4.6

Model	Rmse	Percentage rmse change
Using both location & practitioner	0.041	-
Not using practitioner	0.043	4.88%
Not using location	0.044	7.32%
Not using both location & practitioner	0.052	26.8%

Table 4.6: Effect of not using practitioners and locations as a predictor

We see that not using both the location and the practitioner as a predictor increases the rmse with 26.8%. This serves as an argument to use the two as a predictor. If we remove practitioner as a predictor, we find and increase of 4.88% in rmse. Location is still used. If we remove location and still use practitioners as a predictor, we find an increase of 7.32%. This is also reflected in the predictor importance plots, where location is generally regarded to be a better predictor than the practitioner. However, using both locations and practitioners as a predictor is generally better. If an updating procedure is regarded difficult to implement, the hierarchy should therefore be to first implement location as a predictor, and then practitioner.

The answer to the research question 3b, what are the most important predictors of the model in predicting utilization, is therefore to definitely include the current level of utilization, a degree of time spent on non patient-bound appointments, the location of the appointment, and the employee of the appointment. Going forward we will still be using the other predictors, as our goal is to achieve the highest predictive performance, not so much interpretation. The other predictors have shown - be it to a lesser extend - to contribute to the explanation of variance in the added utilization.

4.5 Conclusion

In this chapter we developed three prediction models, and a set of predictors, to find the answer to the research question 3a and 3b. We summarize the chapter with the research questions and the corresponding answers.

- 3a) How does the proposed prediction model perform? In chapter 3 we proposed three prediction methods. In this chapter we developed these methods, first by finding the best tuning parameters for each of the models using cross validation. After cross-validating 500 tuning parameter configurations for each of the models, we train a model on the full training set using the best tuning parameters. We then make predictions using a testing set on these models,and compare the predictions with the actually achieved values. We find that the LightGBM model, with parameters shown in table 4.3 produces significantly lower absolute errors, and a lower variance in prediction errors than the Mars and XGBoost model, with a root mean squared error of 0.041 (R^2 = 0.982) on the test set, compared to Mars, 0.079 (R^2 = 0.929) and XGBoost 0.048 (R^2 = 0.975). Therefore we conclude that the LightGBM model, with mentioned tuning parameters, should be used for the prediction of added utilization, which can then be transformed to a prediction of utilization.
- 3b) What are the most important predictors of the model in predicting utilization?? We have seen that a small group of predictors explains the largest share of the

variance in the added utilization. These predictors are the current level of utilization, the degree of session time spent on non patient bound appointments, the location of the appointment and the employee of the appointment. This is consistent across the three methods. The predictors of the model should therefore at least include these. Going forward, we still use the other predictors as our goal is to achieve the highest predictive performance, not so much interpretation. The other predictors have shown - be it to a lesser extend - to contribute to the explanation of variance in the added utilization.

5 Model Validation

In this chapter, we evaluate the performance of the model trained and cross validated in chapter 4. The purpose is to evaluate the performance of the prediction method in relation to the scheduling environment. First we discuss how the scheduling method interacts with the prediction method, then we show an example of the scheduling method, and the difference between the by Westerink, 2021 proposed Logistic regression method and the in chapter 4 developed LightGBM method. Then we present a simulation model based on the simulation constructed by Westerink, 2021. The advantage of simulation is that it allows us to experiment in a more realistic environment than the train-test-validation approach does. Simulation also allows us to test the LightGBM model in interaction with the Topsis method. Finally, we discuss the experiments and afterwards we present the results.

5.1 Scheduling in practice

In this section, we show how the LightGBM prediction method relates to the scheduling method, described in section 1.1.2 and depicted in figure 1.3. We start by a description of the Topsis method in relation to our scheduling environment and then show how this works in relation to the LightGBM model.

5.1.1 Topsis method

Taking the more technical description of the Topsis method in section 3.1 as a guideline, we describe how Topsis is proposed by Westerink, 2021 to be implemented in the scheduling environment:

1. Create an $n \times m$ matrix with m possible appointments and the n
2. Normalize the matrix using vector normalization.
3. Determine the weighted normalized decision matrix. The method here introduces weights. The most important criterion has the highest weight. The least important criterion has the smallest weight. The sum of the weights equals one. The weights are described later in this section.
4. Determine the positive ideal and negative ideal solution vectors. These are, for each criterion, the best value (for instance, the smallest weighted distance, or the smallest weighted utilization), and the worst value (for instance the largest weighted distance or largest weighted utilization).
5. Determine for each alternative option the relative closeness to the positive ideal solution and the negative ideal solution.
6. Rank the possible appointments in the order of both closest to the positive ideal solution and furthest from the negative ideal solution.

Westerink, 2021 report the weights for each of the criteria. We show the weights in table 5.1. The weights reflect the importance of a criterion with respect to the other criteria. For instance, if distance would carry the weight 1 for every occurrence of the criterion, implying that all the other criteria should be 0, then the Topsis method would always

return the list ranked with shortest distance first on the most important position and the furthest distance of on the least important position.

Required access time ≤ 30 days		**Required access time > 30 days**	
Stage 1			
Criterion	**Weight**	**Criterion**	**Weight**
Distance	0.83	Distance	0.83
Predicted utilization	0.10	Utilization	0.10
Days deviation from target	0.05	Days deviation from target	0.05
Appointment same type as session	0.02	Appointment same type as session	0.02
Stage 2			
Criterion	**Weight**	**Criterion**	**Weight**
Distance	0.30	Distance	0.33
Predicted utilization	0.33	Utilization	0.37
Days deviation from target	0.08	Days deviation from target	0.10
Fragmentation index delta	0.09	Fragmentation index delta	0.10
Appointment same type as session	0.10	Appointment same type as session	0.10
Access time violation days	0.10		

Table 5.1: Topsis weights as reported in Westerink, 2021

The important thing to note is that the weights can have both a limiting as well as an increasing effect on the results, depending how the weights are set. For the positive ideal solution, the method looks for the smallest distance, the smallest predicted or current utilization, the smallest value for the days deviation between a targeted date and the actual scheduled date, the largest value for whether or not the appointment is of the same type, the smallest fragmentation index and the smallest access time violation. The fragmentation index delta represents the change in fragmentation for the assessed appointment slot.

5.1.2 Scheduling and prediction in practice

We show a practical situation in which the advantage of using this method in table 5.2. The table depicts an example in which the Topsis method returns a list of five appointments for the real time state of the system at 20-06-2022. The patient is fictional. The duration is 30 minutes.

Employee	Location	Date	Time start	Time end	Distance	Fragmentation	Predicted Utilization
1	1	05-07-2022	15:00	15:30	9.4 km	-1	0.88
2	2	06-07-2022	08:30	09:00	20.7 km	-1	0.89
1	3	05-07-2022	10:15	10:45	16.5 km	0	0.99
1	3	05-07-2022	10:00	10:30	16.5 km	0	0.99
3	2	05-07-2022	11:20	11:50	20.7 km	0	0.76

Table 5.2: Actual example of using the Topsis method to schedule a patient on 20-06-2022 (scheduling date), with duration of 30 minutes

In 5.2, the first entry is the best appointment option for this patient. The appointment fills a gap in the session between 15:00 and 15:30, and is within the desired access time of 21 days. The location is also the closest to the patient from the five best offered options.

Option 2 also fills a gap, but is worse due to the larger distance and the higher predicted utilization. Observe how appointment options 3 and 4 are for the same practitioner as option 1. The two offered options are worse than option 2 due to the higher expected utilization and the gap of 15 minutes that would be left open by choosing either of the two options. The options are still regarded better than option 5 due to that they are closer to the patient. Distance is clearly regarded more important than the expected utilization, which is of course reflected by the weights, but the question should be asked whether a 4.2 km alternative distance between options 3 and 4 on one side, and option 5 on the other side, is more important than aiming to reduce variance. To summarize, the table shows the advantageous effect of using the Topsis method, but also shows that improvement might be achieved by putting more emphasis on other criteria.

We now elaborate on why a good prediction is important to scheduling. Figure 5.1 shows an example for one practitioner and location from 01-10-2021 to 01-11-2022. The red line shows the actual level of utilization. This is achieved in the first, solid red, part of the figure, and ongoing in the second, dashed red, part of the figure. The predictions are for a new appointment on that same session date.

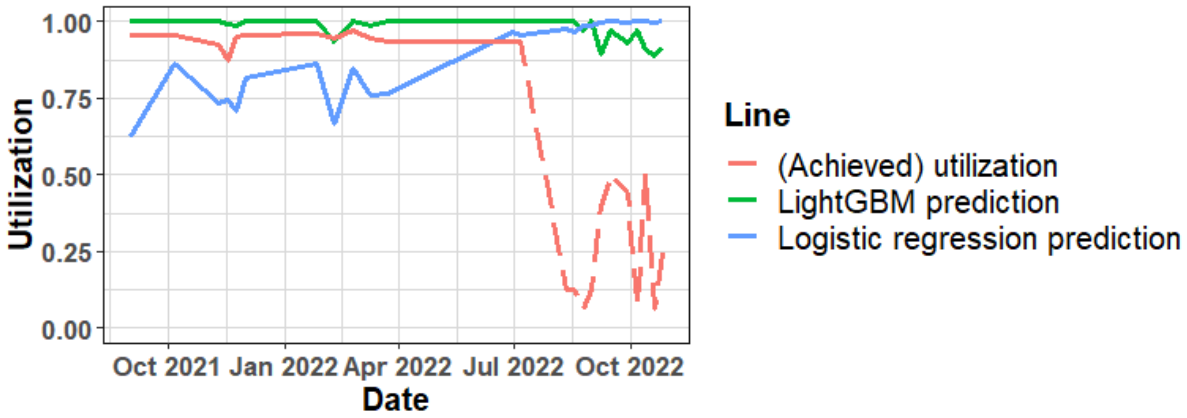

Figure 5.1: Actual prediction example for one employee and location compared to observed (solid red) and current (dashed red) level of utilization

The prediction models add a prediction on top of the latest observed appointment. The LightGBM prediction makes sense. Adding an appointment to the current level of utilization increases the level of utilization, therefore a higher level of utilization is expected. However the logistic regression model, predicting the probability that utilization is higher than average (0.922) seems to predict exactly the opposite. For lower levels of utilization, the probability is predicted to be relatively large, and for higher current levels of utilization a, for instance when the current level of utilization already is larger than average, the predicted level of utilization is lower. For future sessions, we see that the expected level of utilization predicted by the LightGBM model decreases and gets a bit unstable due to the lower current level of utilization and higher access time, but the predictions are still relatively high due to that the previous observations of the same location and practitioner show that the location is generally relatively occupied. Therefore the prediction is closer to one.

In figure 5.1, the logistic regression model follows that in general, but is predicting 1 for sessions that are not heavily occupied. The probability that the utilization is going

to be larger than average is regarded relatively high. The problem with this is that when aiming to distribute patients evenly over the alternative sessions, the sessions are wrongfully excluded, or classified as being less favourable in terms of utilization, whilst they are not. Especially sessions that are further ahead in time are then not returned in the Topsis method. The LightGBM model is therefore a better choice not only in terms of rmse, but also in relation to scheduling.

5.2 Simulation model

We verify the performance of the model constructed in chapter 4 using a simulation. The simulation was initially constructed by Westerink, 2021. We adapt the simulation such that the simulation uses a prediction of utilization by means of the LightGBM model developed in chapter 4. This is the only part that we alter. Figure 5.2 gives an overview of the scheduling process that we simulate. We do this for every appointment that was scheduled between the first of July 2019 and the 31st of July 2019 in the order of which the appointment was scheduled. The scheduling rule is based on the Topsis method, and we allow the method to schedule to all open appointments during the first of July 2019 up to the 31st of December 2020. So there are options for all three of new patients, recurrent patient and periodic patients.

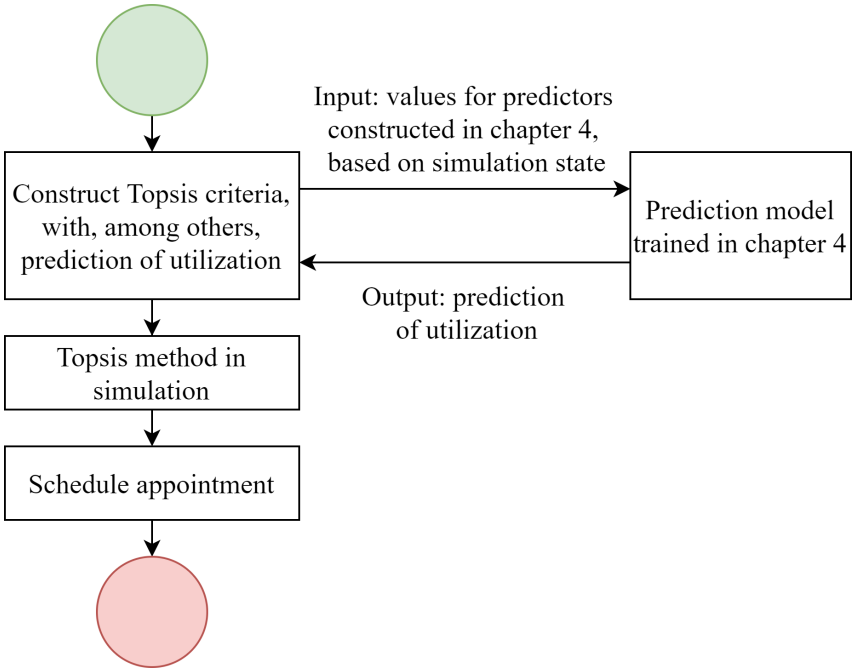

Figure 5.2: Flowchart of relation between simulation and prediction model

During each iteration of the simulation, we construct all the predictors that we developed the LightGBM model on in chapter 4 and were changed based on the previous iteration of the simulation. We send the predictors to the LightGBM method, and return a prediction of utilization to the Topsis method.

Finally, we determine results. For the results we are mainly interested in the first months due to that we only simulate appointments that were originally scheduled during July, 2019. Therefore we will mainly focus on the first months starting from July 2019.

In the simulation we simulate the **online scheduling process** as described in section 1.1.2. We reconstruct the calendar and practitioners' schedules (the sessions) as of the first of July, 2019. Then we retrieve the list of appointments requested by patients, and arrange the list in ascending order on the date and time at which the appointment was scheduled. Then, we schedule the appointments to the schedules, one-by-one, in the order of which appointment was scheduled first (the order of the appointment list). We do this to imitate online scheduling as this mimics the scheduling process at the organization best. The session that the appointment is scheduled to is determined by the Topsis method, described in section 5.1.1. In this section, we discuss the structure and design of the simulation.

5.2.1 Assumptions and simplifications

The simulation has some modelling assumptions and simplifications about the scheduling process (section 1.1.2). They are first discussed so that the modelling choices can be better understood. The assumptions and simplifications are considered to ensure that the simulation runs correctly, and is not too computationally expensive. The assumptions and simplifications are mentioned by Westerink, 2021.

In the simulation, non patient-bound appointments are scheduled before any other type of appointment. This assumption is consistent with how non patient-bound appointments are almost always scheduled before any of the patient-bound appointment in a session. Appointments are always scheduled to a session that is of the same type of session. That is, a diabetics appointment is always scheduled to a diabetics session type.

Currently, there is no data on the considered appointment options. This makes it hard to infer anything regarding patients' preferences during appointment talks. Therefore, patient choice is modeled by taking a random appointment from the 5 best by means of the Topsis method (section 5.1), each with equal probability. This also corresponds to the desire of the organization to limit the number of appointment options offered to the patient. As an example, the organization is currently experimenting with offering only three options. The corresponding assumption is that the difference in effect of patient choice between reality and simulation is not large enough to invalidate the simulation.

We also assume that a patient requires at least 30 minutes to arrive to a location. Therefore, if there is an option for the same date, the feasible time slots are at least 30 minutes later than the observed time at which the appointment was scheduled.

Finally, to be able to generate a range of possible options for recurrent and periodic appointments, it is assumed that the patient corresponding to this appointment desires an access time (in weeks) equal to the actually observed access time in weeks, rounded to the nearest integer, say w weeks. In the simulation, the Topsis method then aims to find an appointment within the week that is w weeks ahead.

5.2.2 Simulation layout

The simulation consist of several stages. Figure 5.3 depicts the simulation and the flow. These are, in chronological order

1. Loading and transforming historical session data and patient appointment data.

2. Ordering list of patient appointments by time that the appointment was originally scheduled from earliest to latest.
3. Finding feasible sessions.
4. For all feasible session generating session and appointment information.
5. Generating predictors and predicting utilization.
6. Performing the first stage of Topsis in which the 30 best feasible sessions are chosen. One of the steps here is the prediction of utilization, based on either the logistic regression method as implemented by Westerink, 2021 or the LightGBM model as trained in chapter 4.
7. For all feasible sessions, finding the feasible slot options.
8. Performing the second stage of Topsis in which the slot options and corresponding sessions are ranked. Here, the prediction is again used.
9. Scheduling the appointment to one of the best sessions
10. Updating the simulation state.
11. After all appointments have been scheduled, writing the results and stop the simulation.

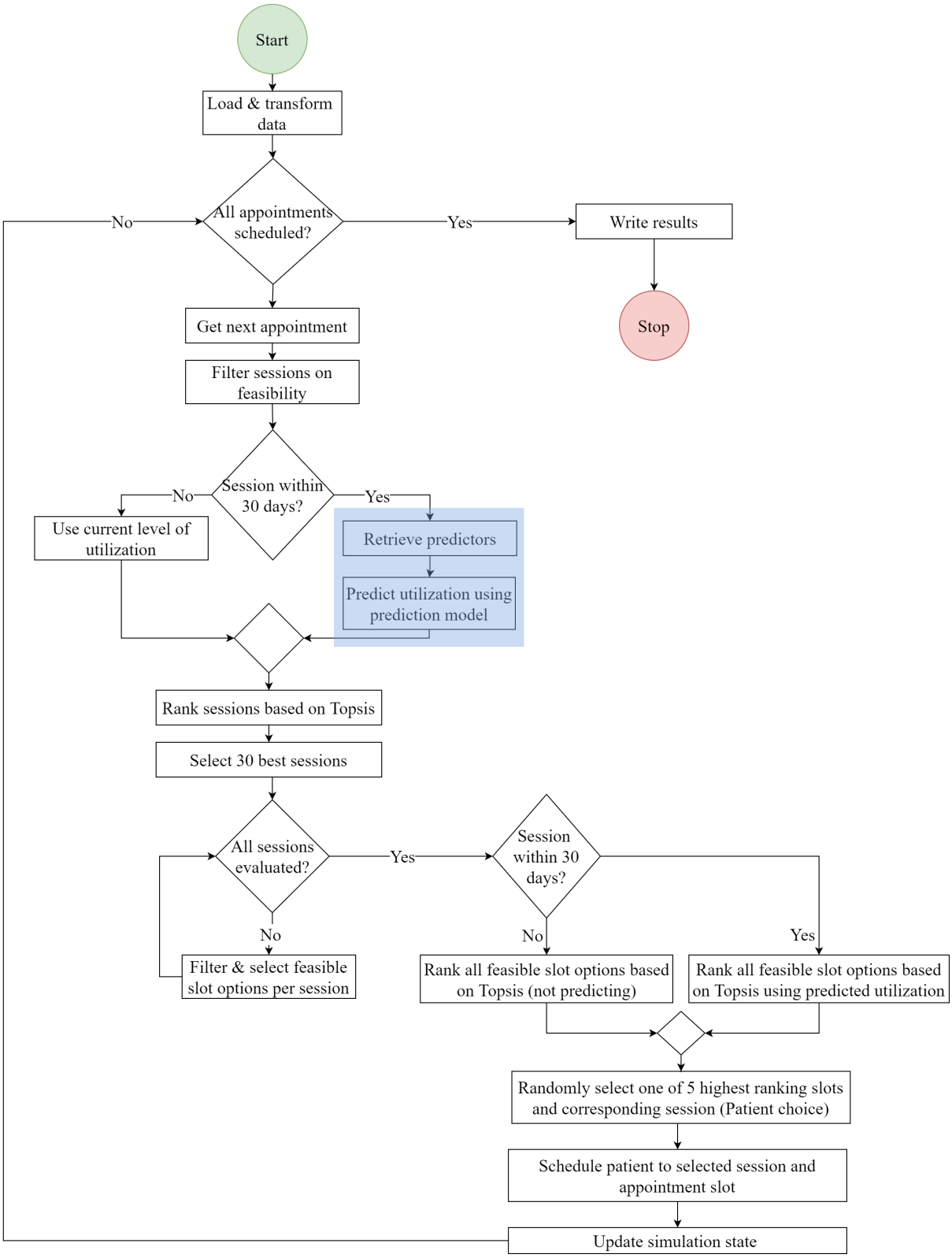

Figure 5.3: Simulation flowchart

In figure 5.3, recall that the blue square area is the only part of the simulation flow that we alter. Every simulation run returns two results. One of which is the simulation results based on using the logistic regression prediction model model as implemented by Westerink, 2021 and the other is the simulation result based on the LightGBM model as

trained and tested in section 4. The second stage of the scheduling procedure does use the prediction of utilization, but in this step, the value is not different from the value in stage 1. Yet, due to that other criteria such as the fragmentation are added here (since they are based on specific appointments, not on session), the predicted utilization is used here again.

We leave the largest part of the simulation and the scheduling process in place. The most important data in the simulation are the state of the sessions as of the first of July, 2019 as well as all the appointments scheduled from the first of July, 2019 until the first August 2019. That are 23283 sessions and 13522 appointments, of which 1234 are non patient-bound appointments.

The first stage of the simulation consists of loading and transforming the input data. During transformation, some of the appointment and session data are transformed into usable predictors for instance, to speed up the simulation. Examples are the break ratio and whether the week is even or odd, but for predictors that depend on the state of the simulation such as the current level of utilization, this cannot be done in the first stage.

5.2.3 Finding feasible sessions

Then the simulation starts scheduling appointments. The simulation starts with the non-patient-bound appointments. These have a predefined session and time slot, and are scheduled in that time slot. After all non-patient-bound appointments have been scheduled, the simulation continues with patient-appointments. The simulation starts with the appointment that was registered first.

For the appointment that is to be scheduled, first the feasible sessions are retrieved. These are sessions that are regarded as options for the appointment. The appointment can have the options "As soon as a possibility occurs", "Today", "Within 24 hours", "Within one week" or "X weeks". As a result of regulations, some types of care require the patient to have an appointment within 24 hours for instance. In the simulation, the corresponding feasible sessions are thereby required to take place within the defined window. "X weeks" also means that the appointment is scheduled not sooner than the number of weeks that is specified. Other constraints are a predefined date, a predefined time slot, a predefined practitioner or a predefined location. A predefined location may be required due to the availability of specific equipment only available at the predefined location. The appointment could also be required to be scheduled in a specialized session, such as an appointment specifically for diabetics or specifically for children. Finally, sessions for which the remaining available time is shorter than the length of the appointment are also filtered as no feasible option is within the session.

For each of the remaining feasible sessions, the predictors are constructed and a prediction is made using the boosting procedure, developed in chapter 4 if the session takes place within 30 days from the schedule date. Then the first step of Topsis is performed. During the step, the sessions are ranked on multiple criteria (corresponding to the weights presented in table 5.1). These are the distance to the location, either the predicted utilization or the current utilization, the days of deviation between the session date and the target date, and whether the type of the appointment, diabetics for instance, is also the type of the session. This results in a ranked list of which the 30 best options now enter the second stage of the simulation.

5.2.4 Generating time slot options

In the next stage of the simulation, the best combination of a time slot and session among the 30 best sessions is chosen. First, for the 30 sessions, the feasible starting time slots are retrieved. The slots are a representation of time.

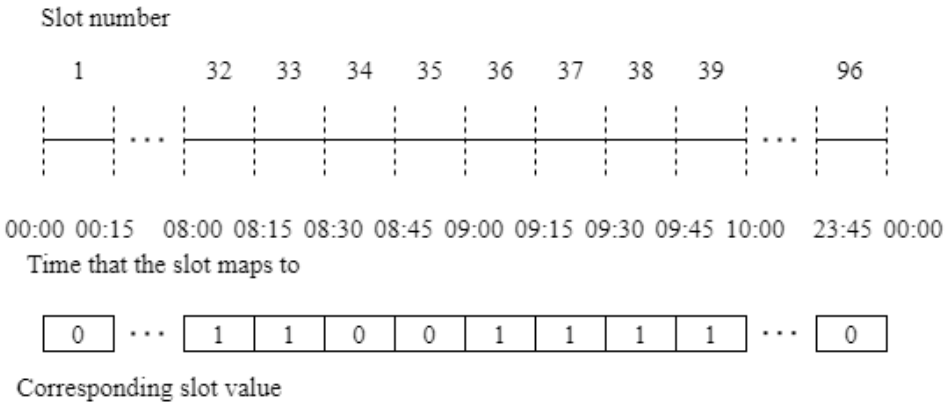

Figure 5.4: Timeslot design

Figure 5.4 depicts the design of the time slots. A day is divided into 96 time slots of 15 minutes, where slot 1 maps to the time slot 00:00 to 00:15 and slot 96 maps to time slot 23:45 to 00:00. Every session has such a mapping of slot options and a corresponding 0 or 1 value representing whether or not the option is open. An appointment of length 30 minutes requires two time slots. Note that the dots indicate that the day also has time slots in between but it is not helpful to show them all.

The **starting** time slots are **not** feasible if

- They are outside of the start and end time of the session. This is represented by a value 0 in the time slot prior to the first open time slot, and a value 0 in the time slot following the last open time slot.
- There already is an appointment scheduled to the time slot, represented by a 0.
- The number of open time slots succeeding the starting time slot is smaller than the number of time slots required by the appointment minus the starting time slot. That is, the range of open slots with value 1 should be at least as long as the number of slots that the appointment requires. Say that an appointment requires one hour, so 4 time slots, then in the schedule represented by figure 5.4, only the slots between 09:00 until 10:00 are an option.
- The time slot is not a predefined time slot when this is required.
- The session date is the same as the schedule date and (the appointment is today) and the time slot is not at least 2 slots later than the time that the appointment is scheduled. It is assumed that the appointment is at most two slots, so one half hour later than the schedule time in order for the patient to travel to the location.

Next, for the sessions and the starting time slots, the fragmentation factor is updated and access time violation are constructed. The fragmentation factor is increased with 1 when the proposed time slots that form the appointment are surrounded by two open time slots (the appointment creates a new block of appointments). The fragmentation

factor is decreased with 1 when the proposed time slots that form the appointment are surrounded by two filled time slots (the appointment fills a gap between two blocks of appointments). The fragmentation factor remains the same when the proposed time slots that form the appointment are surrounded by one filled time slots and one open time slot (the appointment extends a block of appointments). The access time violation criterion is increased with 1 if the proposed time slot does not meet the required access time for the appointment. The combination of all access time violations can be seen as a service level for patients being served in time.

Then the starting time slots and corresponding sessions are again ranked using the Topsis. In this step, beside the distance, prediction of utilization, the days deviation for the targeted date, and whether the schedule type corresponds to the appointment type, also the fragmentation factor and the access time violation are used.

From the ranked starting time slot and corresponding sessions, one of the five best options is randomly chosen. This is done randomly in order to simulate patients' choice in interaction with the service agent or the practitioner. The appointment is scheduled to the session and the simulation state is updated in the sense that the predictors that are subject to change in the schedule, for example the average utilization in a region, are updated.

Finally, when all patients are scheduled, then the results are written to an output file and the simulation is stopped.

5.3 Experiments

We want to assess the validity of the prediction model in a more real environment using the simulation model. It is the idea to compare the performance of the in chapter 4 proposed model with the performance of the currently proposed (logistic regression) model by Westerink, 2021 in practice. We argued that this can be measured in terms of

1. Variance in session utilization
2. Waiting time for new patients

The variance in session utilization is a measure of dispersion of appointments over the total available work time. A good distribution of appointments over the sessions leads to that the overall level of utilization can be increased. It is one of the aims of the proposed Topsis method to reduce the variance in utilization.

Westerink, 2021 also use fragmentation as simulated performance measure, but since the fragmentation index is not something that the prediction method alters, we do not regard this an important feature for this study, but that does not mean that for the organization this can be an important factor.

Since the influence of the performance of the model depends on the weights used in the Topsis method, we propose experiments to assess the performance. Table 5.3 shows the experiments with corresponding weights. We should note that in the two stage approach, the first stage has all influence on forming the set of sessions from which the appointment is eventually chosen. That also means that if we experiment, putting more weight or less weight on utilization should be done in the first stage.

- Experiment 1: Run the default configuration with weights proposed by Westerink, 2021
- Experiment 2: Put 50% less weight on utilization criteria
- Experiment 3: Put 50% more weight on utilization criteria
- Experiment 4: Give all weight to utilization criteria

First of all, in experiment 1 we assess the performance using the weights proposed by Westerink, 2021 to directly compare the two studies and to let is serve as a baseline. Then in experiment 2 we give the utilization criteria 50% less weight, and in experiment 3 experiment we give utilization criteria 50% more weight. We compare that to the baseline. For the other criteria we leave the relative proportion of weights to each other the same in these two experiments. In experiment 4, we give all weight to the utilization criteria as an extreme scenario.

Required access time ≤ 30					**Required access time > 30 days**				
Stage 1									
	Experiment weight					**Experiment weight**			
Criterion	1	2	3	4	**Criterion**	1	2	3	4
Distance	0.83	0.876	0.784	0	Distance	0.83	0.876	0.784	0
Predicted utilization	0.10	0.050	0.150	1	Utilization	0.10	0.050	0.150	1
Days deviation from target	0.05	0.053	0.047	0	Days deviation from target	0.05	0.053	0.047	0
Appointment same type as session	0.02	0.021	0.019	0	Appointment same type as session	0.02	0.021	0.019	0
Stage 2									
	Experiment weight					**Experiment weight**			
Criterion	1	2	3	4	**Criterion**	1	2	3	4
Distance	0.30	0.374	0.226	0	Distance	0.33	0.427	0.233	0
Predicted utilization	0.33	0.165	0.495	1	Utilization	0.37	0.185	0.555	1
Days deviation from target	0.08	0.100	0.060	0	Days deviation from target	0.10	0.129	0.071	0
Fragmentation index delta	0.09	0.112	0.068	0	Fragmentation index delta	0.10	0.129	0.071	0
Appointment same type as session	0.10	0.125	0.075	0	Appointment same type as session	0.10	0.129	0.071	0
Access time violation days	0.10	0.125	0.075	0					

Table 5.3: Experiment weights

Note that in table 5.3 the weights are rounded to three digits. Therefore the sum of the weights can be a little less or a little more than 1. But since the Topsis method ranks according to the relative difference between the weights, this is not an issue as long as the difference is not too large.

		July		August		September		July and August		August and September		July, August and September	
Exp Nr	Model	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1	Logistic	0.871	0.141	0.611	0.238	0.338	0.164	0.754	0.23	0.465	0.243	0.612	0.288
	LightGBM	0.862	0.167	0.617	0.246	0.344	0.167	0.752	0.24	0.47	0.248	0.612	0.291
2	Logistic	0.878	0.153	0.612	0.261	0.34	0.181	0.758	0.247	0.466	0.26	0.615	0.301
	LightGBM	0.875	0.172	0.617	0.27	0.342	0.185	0.759	0.256	0.47	0.266	0.616	0.306
3	Logistic	0.865	0.136	0.61	0.226	0.335	0.158	0.751	0.222	0.462	0.237	0.608	0.283
	LightGBM	0.842	0.164	0.635	0.224	0.341	0.159	0.749	0.219	0.477	0.241	0.609	0.279
4	Logistic	0.848	0.127	0.615	0.201	0.333	0.149	0.742	0.201	0.464	0.224	0.603	0.268
	LightGBM	0.802	0.157	0.661	0.202	0.344	0.148	0.739	0.192	0.491	0.236	0.603	0.259

Table 5.4: Mean and standard error in utilization resulting from the simulated experiments

We report the results in terms of utilization from the simulated experiments in table 5.4. The results on all criteria are shown in table I.1. Due to that the simulation has an initial state, we report multiple months of the simulation. Showing only July would result in a distorted view on the actual situation. That is due to that in the simulation, appointments can be scheduled in August and September. On the other side, since we limit the number of appointments to those that were requested in July 2019, the sessions in the months after September will almost solely consist of non-patient bound appointments. Hence we will not observe a large difference between the two prediction models there and therefore we do not report the results of these months.

We also see that the more weight we put on the prediction of utilization criteria, the lower the variance in utilization gets and, the larger the difference between the Logistic regression model and the LightGBM model is. For experiment 3 ($F_{1,6435} = 5.536$, p = .0187) and for experiment 4 ($F_{1,6435} = 24.072$, p < .001), the differences are significant. Obviously this comes at a cost of the other criteria. For those two experiments the average distance increases from 3.6 km on average in the default experiment to 4.6 km on average in experiment 3 and to 79.5 km on average in experiment 4. The extreme result in experiment 4 is clearly due to that distance is not considered an important criterion in that experiment. As a result, appointments could be scheduled to any location which is of course impractical in reality. For experiment 3 however, distance increases with 1 km, but the variance is significantly reduced. This brings up a relevant question to the organization. Whether or not an increase in distance is tolerated if that leads to a reduction in variance in utilization, and by how much.

We see that in general for the month July, the simulations with the logistic regression models result in higher level of utilization and lower standard deviations when compared to the LightGBM models. We see that the image shifts in the later months. In September we report higher levels of utilization for the LightGBM model and we see that the difference in standard deviation is relatively smaller than for the month July. For some of the experiments, the standard deviation is even lower. Figure 5.5 depicts the results for every experiment in weekly average utilization between the two models.

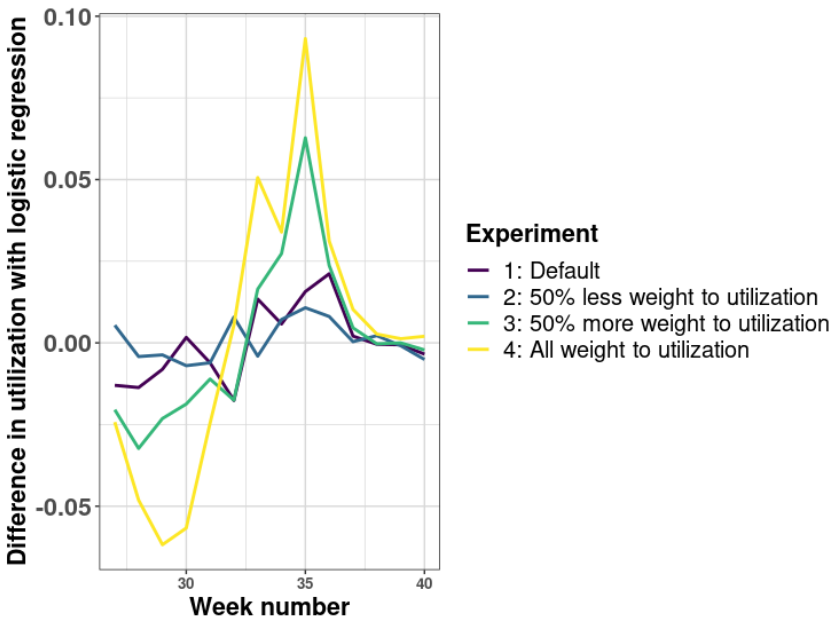

Figure 5.5: Difference in utilization compared to logistic regression model for each of the experiments

We identify a couple of reasons for this behaviour. First of all, the logistic regression model predicts the probability that the session utilization will be larger than average. In consultation with the authors of Westerink, 2021 we learn that this probability can still turn out to be relatively low for higher levels of utilization. The LightGBM is more likely to predict larger levels of expected utilization due to that in the initial state, the most important predictor for the LightGBM model, the current level of utilization, is also relatively large. Therefore the relative difference between the different sessions is small and the Topsis method regards these appointments to be relatively bad. Especially when there is more weight on the predicted level of utilization. Therefore, the Topsis method will schedule an appointment further ahead in time. This is exactly what we see in the simulation.

A second reason is that by shifting the weights more to the utilization criteria, we automatically put less weight on being served quickly due to that we make the weights "Days deviation from target" and "Access time violation days" smaller. Therefore one of the interesting experiments to consider is experiment 2, where we put less emphasis on the utilization criteria but put more emphasis on being served quickly. We expect to see a decrease in the expected waiting time for new patients in experiment 2. For the other experiments we can also see an increase due to the shifted weights. We show the results in table 5.5.

		July		August		July and August	
Exp Nr	Model	μ	σ	μ	σ	μ	σ
1	Logistic	11.6	6.88	18.8	4.86	14.8	7.03
	LightGBM	12.5	6.83	19.2	4.96	15.6	6.91
2	Logistic	11.1	6.49	17.7	5.5	13.8	8.65
	LightGBM	11	6.7	17.8	5.28	13.8	5.47
3	Logistic	11.9	6.93	20.1	5.09	15.7	6.92
	LightGBM	15	6.68	22.5	4.67	19.1	7
4	Logistic	13.3	8.27	23.6	5.75	19	7.37
	LightGBM	19.1	6.28	27.6	2.33	25.2	6.82

Table 5.5: Waiting time for new patients

Note that due to that we have the requirement that new patients are seen within 30 days we do not schedule any new patient in the months after August. The results show that waiting time for new patients tends to increase when more emphasis is being put on the prediction of utilization. This is in line with what we expect, and also with the increase in overall utilization we see for the LightGBM models since it speaks to the resource utilization paradox, where large utilizations come at the cost of large waiting times.

Another interesting result is that we see that for both the logistic regression as for the LightGBM model, the 13.8 days of waiting time is actually under the 14.8 days of waiting time for the model where 50% less weight was put on utilization. Recall that a waiting time of less than 14.8 days for new patients is reported as the research goal. For that experiment we also report the largest average utilization for all of the models and experiments when the LightGBM model is used. These two results suggest that the weights can be set such that an increase in utilization and a decrease in waiting time for new patients are possible.

The results for both the utilization and the waiting time for new patients also suggest to run extra experiments to investigate the influence of the weights of being served quickly on both the utilization and the waiting time. We propose three experiments in which we:

- Experiment 5: Put two times as much weight on being served quickly and leave the weights on the utilization criteria the same.
- Experiment 6: Put 50% more weight on the utilization criteria and put two times as much weight on the being served quickly criteria
- Experiment 7: Put two times as much weight on being served quickly and put 50% less weight on the utilization criteria.

The weights for the experiments are shown in table 5.6

Required access time $\leq$ 30				Required access time $>$ 30 days			
Stage 1							
	Experiment weight				**Experiment weight**		
Criterion	5	6	7	**Criterion**	5	6	7
Distance	0.786	0.732	0.830	Distance	0.786	0.732	0.830
Predicted utilization	0.095	0.150	0.050	Utilization	0.095	0.150	0.050
Days deviation from target	0.100	0.100	0.100	Days deviation from target	0.100	0.100	0.100
Appointment same type as session	0.019	0.018	0.020	Appointment same type as session	0.019	0.018	0.020
Stage 2							
	Experiment weight				**Experiment weight**		
Criterion	5	6	7	**Criterion**	5	6	7
Distance	0.234	0.089	0.291	Distance	0.293	0.152	0.383
Predicted utilization	0.258	0.495	0.165	Utilization	0.329	0.555	0.185
Days deviation from target	0.160	0.16	0.160	Days deviation from target	0.200	0.200	0.200
Fragmentation index delta	0.070	0.027	0.087	Fragmentation index delta	0.089	0.046	0.116
Appointment same type as session	0.078	0.029	0.097	Appointment same type as session	0.089	0.046	0.116
Access time violation days	0.200	0.200	0.200				

Table 5.6: Weights for new experiments 5, 6 and 7

The results for experiments 5, 6 and 7 are listed in table 5.4. When we put more weight on being served quickly, and leave the proportions between the other criteria the same (experiment 5) we find that waiting time for new patients can decrease to 11.3 days compared with the weights according to Westerink, 2021. Also, we find that using the LightGBM model results in lower waiting times than the logistic regression model, with 11.3 and 11.8 days of waiting time for new patients respectively. The reduction in access time comes at the cost of increased distance, which increases from 3.6 km to 4.4 km for the logistic regression model and 4.2 km for the LightGBM model. We see an increase in the schedule fragmentation and the variance in utilization, but we also see a slight increase in utilization. The increase in variance is therefore likely due to the increase in weights of being served quickly. This is reflected in the total number of scheduled appointments per experiment, where we see that the number of appointments scheduled in the later weeks is higher for the experiments where less weight is put on being served quickly criteria.

Therefore, in experiment 6 we put more weight to avoiding larger expected levels of utilization. The result is that the variance in utilization decreases compared to the default experiment and also compared to logistic regression model in the same experiment. We also see a decrease in waiting time for new patients, but the decrease is larger for the logistic regression model than for the LightGBM model. This is arguably also due to that the utilization criteria in Topsis are regarded larger in the LightGBM model than in the logistic regression model. Therefore the options corresponding to larger expected utilizations are regarded worse than the ones with smaller expected utilizations. The Topsis ranking favors those with smaller utilizations which are positioned further ahead in time, resulting in larger waiting times for new patients. This also comes at the cost of distance, which increases to 8.2 km for both the logistic regression and the LightGBM model in this experiment.

In experiment 7, we investigate the effect of reducing the weight of the utilization criteria, while we increase the weight of the being served quickly criteria. The effect is an expected increase in variance in utilization for both the logistic regression and the LightGBM model, but almost all of the other KPIs perform better. Using the LightGBM model we report a waiting time of 11.3 days, and 11.6 for the logistic regression model. The utilization

for the LightGBM model also is 0.616. This is higher than the 0.612 reported in the default experiment. In experiment 7, this does not come at the cost of distance, which actually decreases to 3.4 for the logistic regression model and to 3.3 for the LightGBM model. A possible reason for this is that relatively, weight shifts from the utilization criteria to the distance criteria. This is also true for the other criteria. We see that the average fragmentation decreases as well, and the average number of days deviation from the target date decreases from 5.3 to 2.6. The result on this experiment suggests that the Topsis method actively spreads appointments to sessions that are expected to be less utilized, and schedules the time left open with new patient appointments.

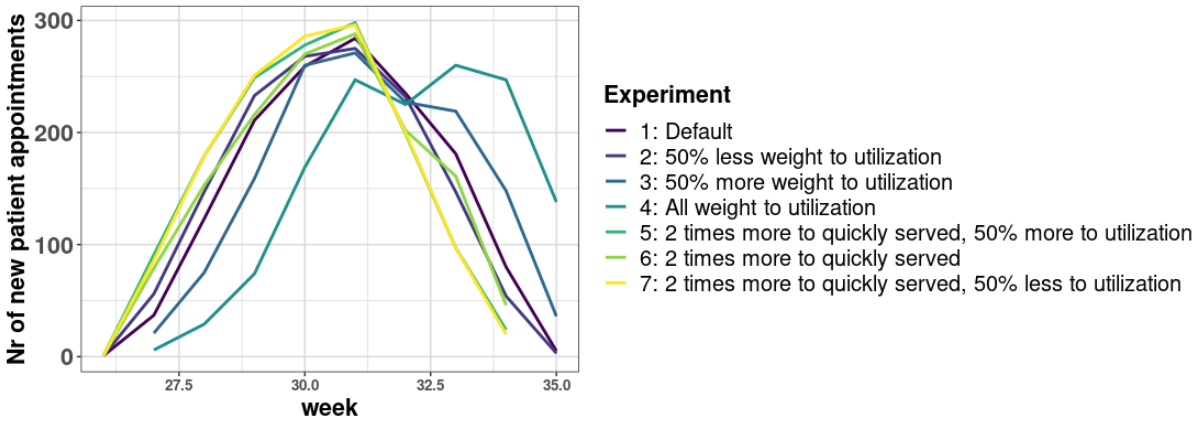

Figure 5.6: Number of new patient appointments per week for each experiment using boosting

Figure 5.6 shows the number of new patients scheduled to the session corresponding to the week on the horizontal axis. It is seen that in experiment 4, the new patients are scheduled more to later weeks, whereas in experiment 7, new patient appointments are scheduled to sessions taking place more in the earlier weeks. New patients are not the only type of appointments, therefore, the ratio of new patient appointments to other type of appointments is shown in figure 5.7.

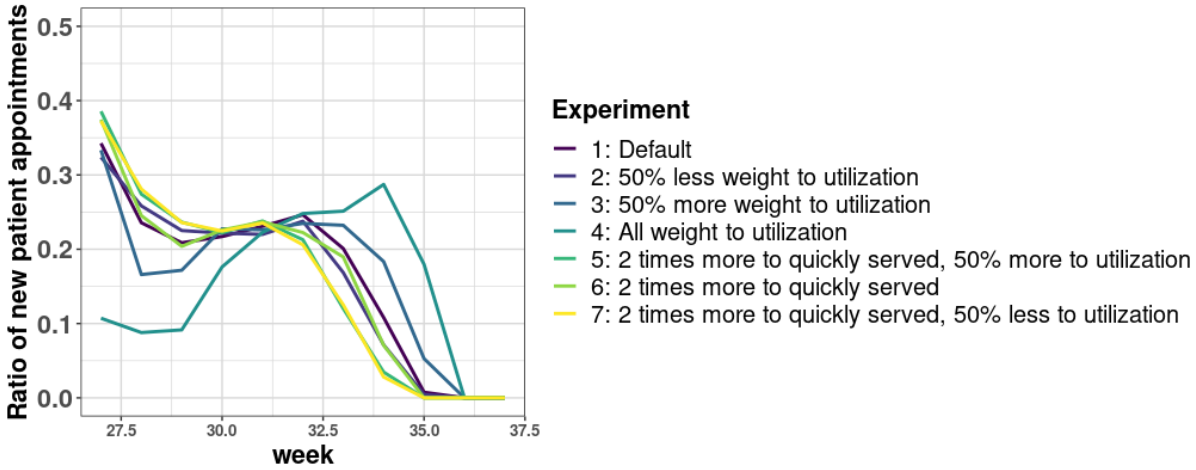

Figure 5.7: Fraction of scheduled appointments that were new patient appointments per week for each experiment using boosting

In figure 5.7 we see that if we put more weight to scheduling appointments quickly, the appointments in earlier weeks are scheduled with relatively more new patients (see experiments 5-7) than when the weight is lower (experiments 1-4). If we put less emphasis on utilization we find that the level is slightly lower (experiment 7), and if we put more emphasis on utilization, the level is slightly higher (experiment 5). In those three experiments, the ratio of new patient appointments in then obviously lower in the later weeks. In the experiments in which no extra weight was put on being served quickly, the ratios are smaller in the first weeks and then higher later on. Note that we let the figure start at week 27 instead of week 26 due to that the number of scheduled appointments is very low (max 6) in week 26, which does not provide meaningful insights. We conclude that the behaviour of the Topsis model is as expected.

In conclusion, we answer research questions 4a and 4b. We start with research question 4a, how can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the variance in utilization? We find that in the simulation the LightGBM model performs better than the logistic regression model depending on the used weights and the KPIs. We see that if we put all weight to the level of utilization criteria, using the LightGBM model significantly ($p < .001$) reduces the variance in utilization compared to the logistic regression from 0.083 to 0.067, which serves as evidence that the LightGBM model is a better model for the prediction of utilization than the logistic regression model. However, putting full weight to utilization comes at the cost of distance.

The answer to research question 4b, to what degree can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the waiting time for new patient appointments, is that we are under the impression that in the by Westerink, 2021 proposed set of weights, there is too much emphasis on the prediction of utilization when considering the results seen in experiment 7. The results in that experiment show that waiting time for new patients can be decreased from 14.8 to 11.3 days if the weights are set such that more emphasis is put on being served quickly rather than on the predicted level of utilization. The experiment also shows a reduction in distance, fragmentation and maintaining the target date. However, the variance in utilization is increased from 0.083 to 0.096. We identify a possible reason in that the time left open in more utilized sessions is filled with appointments for new patients, when appointments of that type occur. Finally, we conclude, since reduction in variance should not be a goal on its own but rather something to exploit, that the LightGBM model produces better results on the KPIs in the simulation than the logistic regression model.

5.4 Conclusion

In this chapter we used a simulation model to find the answer to the research question 4a and 4b

We summarize the chapter with the research questions and the corresponding answers.

- 4a) How can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the variance in utilization? We find that in the simulation the LightGBM model performs better than the logistic regression model depending on the used weights and the KPIs. We see that if we put all weight to the level of utilization criteria, using the LightGBM model significantly ($p < .001$) reduces the

variance in utilization compared to the logistic regression from 0.083 to 0.067, which serves as evidence that the LightGBM model is a better model for the prediction of utilization than the logistic regression model. However, putting full weight to utilization comes at the cost of distance.

- 4b) To what degree can the prediction model influence the by Westerink, 2021 proposed scheduling method to reduce the waiting time for new patient appointments? We are under the impression that in the by Westerink, 2021 proposed set of weights, there is too much emphasis on the prediction of utilization when considering the results seen in experiment 7. The results in that experiment show that waiting time for new patients can be decreased from 14.8 to 11.3 days if the weights are set such that more emphasis is put on being served quickly rather than on the predicted level of utilization. The experiment also shows a reduction in distance, fragmentation and maintaining the target date. However, the variance in utilization is increased from 0.083 to 0.096. One possible hypothesis for this is that new patient appointments are scheduled in the time left open by earlier iterations for other appointments of the Topsis method, whereas the Logistic regression model might still predict a large level of utilization for the session similar to 5.1.2 causing other sessions to be listed higher in the returned Topsis list. Finally, we conclude, since reduction in variance should not be a goal on its own but rather something to exploit, that the LightGBM model produces better results on the KPIs in the simulation than the logistic regression model.

6 Discussion

This chapter is dedicated to answering the research questions and giving a concise answer to the research goal. The chapter presents the research limitations, recommendations for the organization, recommendations for future research. The chapter also discusses the contribution to scientific literature and the generalizability of the study.

6.1 Conclusion

In this section we answer the research goal using the research questions. The research goal was:

How can we develop a prediction model that reduces the difference between observed and predicted utilization, in order to influence the proposed scheduling method such that the access time for new patients is lower than 14.8 days?

To find answers to this research goal we used a group of 4 research questions. In this section we describe each research question separately, and we put together the answers in the final part of this section.

Research question 1: What is the current performance of the scheduling system and the proposed prediction method?

We measure the current performance of the scheduling system and the currently proposed prediction method in terms of access time, variance in utilization and predictive quality of the currently proposed prediction method.

We find that the access time is 20.5 days. This value is the access time for appointments scheduled in 2019, which we regard the most representative year, and we regard the period of one year long enough to be able to observe the characteristics in access time showing up over time. For the total period of time, the access time was 18.8 days. The results are in line with the increase in number of appointments that the organization faced over the last years, coming at the cost of increased access time. We also found that the access time from the perspective of the individual appointments follows a somewhat remarkable pattern. We found that the access times are often recurring around a $7, 14, 21, ...$ series. Two reasons were identified. First of all, scheduling a multiple of 7 days ahead is always an option. This is due to that patients call on weekdays, and not in the weekend. Secondly, practitioners tend to schedule a multiple of 7 days ahead due to that the practitioner's work schedule is either weekly or two-weekly recurrent. Therefore the practitioner might not be available any other number of days than 7 or 14. Finally, the access time distribution for sessions is positively skewed. The group of recurrent appointments are responsible due to the often large access times.

For the level of variance we also regard 2019 to be the most representative year. Since variance is a measure of dispersion around a mean we also report the mean utilization. We know that high levels of variance in utilization make it to maintain high levels of utilization. This speaks to the resource utilization paradox. We find that the level of utilization during 2019 was 89.7% and the variance was .0173 We see that the variance heavily increased in the total period of time as we see that the all time utilization mean and variance are 87.8% and the variance was 0.024. When we consider the variability in utilization over time, we find that The second quarter shows the highest levels of

utilization, and the first quarter the lowest. However the differences are small. In terms of months and weeks we see that holiday-filled periods of time like August, or the last week of December, leads to lower utilizations and in terms of days we see that sessions on Saturdays are by far less utilized. We also find that utilizations are differently distributed for different locations.

Due to that different models using multiple interaction effects have been performed we tested all of these models and reported the results for the model with the best performance. The model shows a classification error rate of .362 and a cross validated error of .217. The model has difficulties with predicting an accurate probability that the utilization is higher than average when the current level of utilization is low. Also interaction effects were included to some extend. At that point, the prevailing idea was that the relation between the predictors and utilization is often nonlinear, and that the prediction quality could benefit from more flexible modeling techniques.

In conclusion, the access time level gives a benchmark that we can use. We find information regarding the scheduling process in the distributions of access time. We find that utilizations are variable both over time as well as geographically. The found variance in utilization is not only a benchmark, but also an instrument. We also believe that the currently proposed prediction model has difficulty predicting utilization directly and the prediction quality could benefit from more flexible modeling techniques.

Research question 2: Which methods are available in literature regarding the prediction of utilization?

We discussed three alternative approaches towards the prediction of utilization. The first approach is to use time series. Time series methods aim to explain the variance in utilization by means of differences time. Since we saw that there was variability over time, considering the time series approach was a reasonable choice. We also investigated the ability of time series models to non time related predictors since we saw that there we also differences in utilization per region for instance. We found such methods, but due to that time series also require a regular series of observations we did not choose to use time series, as our series are generally not regular. A second considered approach was to use similar literature and to use the modeling techniques described in that literature. However, we found a very limited number of studies that are similar. We found only one similar study, that has a different unit of measure. Therefore choosing the similar literature approach was not an option. The third approach was to use machine learning models. Like time series, machine learning methods offer a general approach towards prediction modelling, but do not necessarily require a regular series. Therefore we chose to use machine learning.

Due to that machine learning modelling has many facets and aspects, we follow a methodology that covers the different aspects. We also describe the main challenges that are found in machine learning modelling. These include dealing with the bias-variance trade-off and using cross-validation in order for models to generalize well, handling specifically typed attributes, transformation and scaling to enhance model predictive accuracy, using model selection and regularization approaches to increase model interpretability and to handle the curse of dimensionality and the use of cross-validated hyperparameter tuning and making use of ensemble methods for models to fit well.

We describe different types of machine learning models and list their characteristics. From this list, we aimed to find models that we believe are a good fit for our prediction problem. We found that the multivariate adaptive regression splines, Light gradient boosting machines and Extreme gradient boosting machines give us three models to test on our data and believe that these three models are a good fit for the prediction problem due to the flexibility the models offer, the amount of data that these models require and the complexity the models exhibit. From the more complex and flexible models, we believe that neural networks are not only more complex, but also require more data to fit well, whilst we learn from literature that neural networks do not necessarily make better predictions. From the less complex and flexible models, we think that the models might not offer enough flexibility, which was a problem with the models tested by Westerink, 2021. Following that reasoning we chose to use the Mars, XGBoost and LightGBM models. Other advantages are that the three models can handle predictor interaction automatically, are flexible and the difficulty is in the parameter tuning. The LightGBM and XGBoost models have been developed for usage in production environments and require little extra work. The Mars model is not that much adapted to production environments, but is included as it can be used as a benchmark for the two boosting models.

We conclude that among the methods found in literature, for us the best methods to use are the Mars, XGBoost and LightGBM model. The main challenges are to tune the model parameters, and to prevent underfitting and overfitting. For both of the methods, cross validation is a solution.

Research question 3: How can we develop a prediction model for session utilization that produces accurate results?

One of the most challenging aspects of the prediction of utilization, is the insight that multiple appointments together form one session utilization. That implies that we cannot directly predict the session utilization, but instead we predict the part of the session utilization that is added to the session after scheduling the appointment. This gives a unique value to train and test on for each of the appointments in a session.

Since one of the challenges in model development is to prevent underfitting as well as overfitting, we use a train-test-validate approach. We start by randomly splitting the data. 20% of all of the appointments are left out of the training process. The other 80% enters the parameter tuning process. For each model, we test 500 different configurations of parameters. For the parameters we use a range of values found in studies that report the tuning of the same model. We use the TPE sampling method, in which we sample values for each of the model hyperparameters in such a way that we search more in promising neighborhoods.

The data comprises a group of predictors. We developed three groups of predictors. Static predictors, stateful predictors and aggregation. Static predictors do not depend on the other sessions or appointments. Examples include the distance to a location or the practitioner bound to the appointment. The second group consists of stateful predictors. This group does depend on other appointments. Examples include the current level of utilization, or the current access time in the session. Finally, there are aggregations. Aggregations are based on multiple sessions. For instance the current level of utilization in a region. To take time variability in consideration we add information regarding public holidays, and to take the pattern in access time into consideration, we use lagged variables in multiples of seven. Finally, since there is a large group of elderly patients we implement

the influenza incidence at the time of scheduling the appointment as a predictor to model illness as a factor for a reduced utilization.

We fit parameter configurations for the three proposed prediction models. After cross-validating 500 tuning parameter configurations for each of the models, we train a model on the full training set using the best tuning parameters. We then make predictions using a testing set on these models, and compare the predictions with the actually achieved values. We find that the LightGBM model, produces significantly lower absolute errors, and a lower variance in prediction errors than the Mars and XGBoost model, with a root mean squared error of 0.041 ($R^2 = 0.982$) on the test set, compared to Mars, 0.079 ($R^2 = 0.929$) and XGBoost 0.048 ($R^2 = 0.975$). Therefore we conclude that the LightGBM model, should be used for the prediction of added utilization, which can then be transformed to a prediction of utilization.

We have seen that a small group of predictors explains the largest share of the variance in the added utilization. These predictors are the current level of utilization, the degree of session time spent on non patient bound appointments, the location of the appointment and the employee of the appointment. This is consistent across the three methods. The predictors of the model should therefore at least include these. The current level of session utilization is an intuitive predictor, as it directly gives an upper bound of total utilization that can be added to the session. For example, if currently the level of utilization is 0.9, then only 0.1 can physically be added. Therefore high current levels of utilization narrows the range of possible outcomes and therefore makes more accurate predictions. The practitioner of the session is the an intuitive predictor due to that inherently, practitioners see different patients. This is not always registered. Some practitioners might put diabetic type patients in their regular session. Others only schedule specific cases in their session themselves. The location of the appointment is intuitive as some locations are more heavily utilized than others due to larger capacity (more practitioners working in one location), or due to that another location is in the neighborhood for instance. It also helps identifying locations that are in locations with less demand. For instance in smaller villages.

The other predictors have shown - be it to a lesser extend - to contribute to the explanation of variance in the added utilization. These are used in simulation, but may require more implementation time and therefore it can be wise not to use these in production or to introduce these later.

To conclude, by cross validating the hyperparameter-tuning process, then training the model with the best parameters in terms of lowest cross-validated root mean squared error, we find that the LightGBM model performs best on the test set with a found root mean squared error of 0.041 and that the model produces significantly lower errors, and significantly lower variances in root mean squared added utilization error than the Mars and LightGBM model. All models perform best using at a combination of at least the current level of utilization, the degree of session time spent on non patient bound appointments, the location of the appointment and the employee of the appointment.

Research question 4: How would such a prediction model perform in relation to the by Westerink, 2021 proposed scheduling method?

We find that in the simulation the LightGBM model performs better than the logistic regression model depending on the used weights and the KPIs. We see that if we put all

weight to the level of utilization criteria, using the LightGBM model significantly ($p < .001$) reduces the variance in utilization compared to the logistic regression from 0.083 to 0.067, which serves as evidence that the LightGBM model is a better model for the prediction of utilization than the logistic regression model. However, putting full weight to utilization comes at the cost of distance. We have also seen that if we put 50% more weight to utilization than the weights proposed by Westerink, 2021, and leave the proportions between the other criteria untouched, we find that there still is a significant ($p = .019$) reduction in utilization between the LightGBM and the logistic regression model. This serves as evidence that the LightGBM model is also better at predicting utilizations. The effect disappears when weights are shifted more towards other criteria. Therefore, we are under the impression that the reduced variance is exploited, as we do see an improvement on the other KPIs, or that in the currently proposed model, the predicted values form a larger range. The effect of this is that in Topsis, the relative importance of the utilization criteria increases compared to the LightGBM model. Therefore the list of session and slot options will be more focused on picking low currently utilized sessions, by which variance in utilization will be decreased compared to the LightGBM model.

We are under the impression that in the by Westerink, 2021 proposed set of weights, there is too much emphasis on the prediction of utilization when considering the results seen in experiment 7. The results in that experiment show that waiting time for new patients can be decreased from 14.8 to 11.3 days if the weights are set such that more emphasis is put on being served quickly rather than on the predicted level of utilization. The experiment also shows a reduction in distance, fragmentation and maintaining the target date. However, the variance in utilization is increased from 0.083 to 0.096. We identify a possible reason in that the time left open in more utilized sessions is filled with appointments for new patients, when appointments of that type occur. Another possibility is that the larger range following from the logistic regression is responsible for that relatively the utilization criteria are regarded more important.

Finally, we conclude, since reduction in variance should not be a goal on its own but rather something to exploit, that the LightGBM model produces better results on the KPIs in the simulation than the logistic regression model.

6.2 Limitations

The limitations in our study are primarily data related. First of all, only predictions were made from appointments that were registered as taking place. Our study does not consider cancelled appointments, appointments that were rescheduled, or no-show appointments as these three limitations are not registered in a structured way. These might be very relevant, since cancelling and rescheduling is an integral part of the process at the organization and can be very relevant for the prediction of sessions.

A second limitation is that limited training resources lead to that the reported performance for each of the models can be incorrect. Especially for the Mars model, there possibly is improvement potential if we had better resources, for instance more training time or more computing power. However we should note that it can also be seen as a quality of the other models that they are able to make better predictions when given the same training time.

Due to the computational effort it takes to run the simulation, and the limited amount of good, representative data due to both a changing process which can be attributed to growth in the number of patients and practitioners over the last years, and the measures taken against the Covid-19 virus, we acknowledge that the simulation is possibly not representative for today's situation.

A fourth limitation that is related to the third limitation is that we rely on a theoretical validation by means of the train-test-validation approach and the simulation. We do not implement the method in practice and therefore do not report the performance in practice.

6.3 Recommendations

We propose a set of recommendations that the organization can follow in order to parse the findings of the study into a usable system.

1. Structural registration of the state of an practitioner workday, especially the current level of utilization and the ratio of time spent on other activities. This requires that appointments are explicitly connected to practitioner workday schedules, which is currently not the case. Beside these, we also recommend to choose among the set of predictors used to train the LightGBM model. Some predictors contribute more than others, therefore one should prioritize based on the predictor gain for instance.
2. Structural registration of data on cancellations, rescheduled appointment and no-show appointments. We expect these factors to have predictive quality.
3. The implementation of the new scheduling method in a testing environment. We are working with actual patient care. A mediocre implementation can come at the expense of patient quality perception. Special attention should be given to the interaction of the scheduling tool with service agents. In the simulation, an appointment from a group of five appointments is randomly chosen, but this might not include an appointment which suffices a patient's requirements. As a backup, the currently functioning system should still be available.
4. Setting the weights of the criteria based on the test environment. The simulation suggests that lowering the waiting time can come at the cost of distance in one experiment, and at the cost of increase variance in utilization in another experiment. The combination of weights should therefore be carefully tested in practice.
5. The scheduling method has a clear ranking in the options it provides. The best option is on top. We recommend to store the number corresponding to the chosen option to evaluate the number of appointments required to show. We suggest to start with a larger group of options and scale down over time. This also gives an option to investigate patient preferences.
6. The LightGBM method should be implemented as part of the scheduling method. The method has shown to be able to predict utilizations well, and to be able to reduce the variances in utilization. The scheduling method has direct influence on the quality of the prediction made by the LightGBM method. Therefore, the model should be retrained over time, also to include new practitioners and locations.

6.4 Scientific contribution

This study presents an approach to the prediction of session utilization from the perspective of individual appointments. We learned from a literature study, that there currently is to the best of our knowledge no study towards the prediction of utilization in a setting in which utilization is defined as the combination of multiple appointments in one schedule. As discussed in section 3.2.1, Schiele et al., 2021 present the only similar study as they aim to predict operating room bed utilization in order to schedule patients to operating room beds, but there is a fundamental difference in the unit of measure. This study extends the knowledge on more flexible and complex prediction methods by evaluating three different methods and show that the methods work well. One of the methods is also validated by means of simulation with a group of experimental configurations.

6.5 Generalizability

In this study we have taken actions to enhance the generalizability to other context. In chapter 3 we first discuss the research approach, from which we learn that there is little similar literature that addresses the prediction of utilization from the perspective of the individual appointment. Therefore we choose to follow a general approach towards prediction problems and explain why we choose machine learning over time series. We propose a methodology that we followed, and also describe what the challenges are that can confine a reliable and good implementation. That approach can directly be followed by others.

In chapter 4 we present a set of predictors that we train our models on. There are predictors that are specifically present in our context such as the appointment types or predictors that are specific to the Netherlands, such as whether or not it was Kingsday, and in what region what location lies. There are also predictors that are more general, such as distance to a location and the current level of utilization. The dynamics of the predictors are arguably very specific to our context, but the general ideas can directly be used in other context. Many countries have their own set of public holidays that can be included.

We also describe how we can train the proposed models. The cross validation procedure is a very general and well covered technique in literature. Training the specific models is covered to a lesser extend, therefore we aim to combine the existing literature on training these models by describing the parameters that the models use, and what values are commonly used in tuning the models. We show the results of the tuning process in the appendices F, G and H. Finally, we also address how the models are compared and how we compare the used predictors.

In chapter 5 we describe how we validate the model in reality by means of simulation. We aim to describe the exact simulation process which adds value to the generalizability. We also show the used weights in the experiments and how the results are compared. However we also note that the set of appointments used in the simulation is limited, which can make it less representative.

6.6 Recommendations for future research

There are multiple questions on what to do for future research. In the machine learning approach we chose three prediction methods from the larger group of possible methods. The models were chosen for multiple reasons among which one is the possible shortage on usable data for for instance the usage of deep learning. Assessing multiple other models not only in training but also in simulation is one way that the research can be improved. This is also relevant for scientific literature and can be applied directly to assess the performance. We mention the lack of data on cancelled and rescheduled appointments as a limitation of this study, and we recognize that these are relevant data which could explain a share of the variance. Sources of cancelled and rescheduled appointments might have to be investigated for this. We also acknowledge that using practitioners and locations as predictors to identify for instance practitioner specific patterns, is a short but inconvenient modelling choice. We are sure that there are underlying reasons such as the activities scheduled in the session or geographic dispersion of patients and we therefore recommend to operationalize the practitioner-specific and location-specific patterns. One option is to categorize the practitioners into new, moderate and experienced practitioners. To categorize locations, we might use a density factor.

Finally, due to a gap in scientific literature we cannot benchmark our trained models with comparable studies that report the implementation of the same models, therefore if similar studies are published, we recommend to compare the reported performance of prediction models. This also speaks to the generalizability of this study. Finally, we do not report the performance in practice and therefore we recommend the evaluation of the proposed prediction model when being used in production.

One appropriate question that can be asked is how other type of models perform on the researched data. A second question is whether more data as well as other predictors will improve the predictions. This directly relates to one relevant question for our study, which is what the reason is for cancelled or rescheduled appointments and how these can be included in our model. And a relevant question is whether similar studies show comparable results. Finally a very relevant and important question is how the models perform in reality.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). "Optuna: A next-generation hyperparameter optimization framework". *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (page 42).

Anghel, A., Papandreou, N., Parnell, T., Palma, A. D., & Pozidis, H. (2018). "Benchmarking and optimization of gradient boosting decision tree algorithms". https://doi.org/https://doi.org/10.48550/arXiv.1809.04559. (Pages 63, 65, 118, 120)

Behzadian, M., Khanmohammadi Otaghsara, S., Yazdani, M., & Ignatius, J. (2012). "A state-of the-art survey of topsis applications". *Expert Systems with Applications*, *39*(17), 13051–13069. https://doi.org/https://doi.org/10.1016/j.eswa.2012.05.056 (page 31)

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). "Algorithms for hyper-parameter optimization." In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, & K. Q. Weinberger (Eds.), *Nips* (pp. 2546–2554). http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization. (Page 42)

Bishop, C. M. (2006). "Pattern recognition and machine learning (information science and statistics)". Springer-Verlag. (Pages 42, 43, 47).

Boehmke, B. C., & Greenwell, B. M. (2019). *Hands-On Machine Learning with R* (pages 46–48, 60, 61, 117).

Box, G. E. P., & Cox, D. R. (1964). "An analysis of transformations". *Journal of the Royal Statistical Society Series B Methodological*, *26*(2), 211–252. http://www.jstor.org/stable/2984418 (page 40)

Box, G., & Jenkins, G. (1970). "Time series analysis: Forecasting and control". Holden-Day. https://books.google.nl/books?id=5BVfnXaq03oC. (Page 34)

Box, G., Jenkins, G., & Day, H. (1976). "Time series analysis: Forecasting and control". Holden-Day. https://books.google.nl/books?id=1WVHAAAAMAAJ. (Page 34)

Breiman, L. (2001). "Random forests". *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324 (page 47)

Calin, O. (2020). "Deep learning architectures: A mathematical approach". Springer International Publishing. https://books.google.nl/books?id=KXtdywEACAAJ. (Page 49)

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). "Crisp-dm 1.0 step-by-step data mining guide" (tech. rep.). The CRISP-DM consortium. https://maestria-datamining-2010.googlecode.com/svn-history/r282/trunk/dmct-teorica/tp1/CRISPWP-0800.pdf. (Page 38)

Chen, T., & Guestrin, C. (2016). "Xgboost: A scalable tree boosting system". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785 (page 48)

Craven, P., & Wahba, G. (1978). "Smoothing noisy data with spline functions". *Numerische Mathematik*, *31*(4), 377–403. https://doi.org/10.1007/BF01404567 (page 70)

De Gooijer, J. G., & Hyndman, R. J. (2006). "25 years of time series forecasting" [Twenty five years of forecasting]. *International Journal of Forecasting*, *22*(3), 443–473. https://doi.org/https://doi.org/10.1016/j.ijforecast.2006.01.001 (pages 33, 34, 39)

Derrick, B., Ruck, A., Toher, D., & White, P. (2018). "Tests for equality of variances between two samples which contain both paired observations and independent observations". *Journal of Applied Quantitative Methods*, *13*. https://uwe-repository.worktribe.com/output/865549 (page 69)

Dev, V. A., & Eden, M. R. (2019). "Formation lithology classification using scalable gradient boosted decision trees". *Computers & Chemical Engineering*, *128*, 392–404. https://doi.org/https://doi.org/10.1016/j.compchemeng.2019.06.001 (pages 65, 120)

Dey, P., & Das, A. K. (2016). "Application of multivariate adaptive regression spline-assisted objective function on optimization of heat transfer rate around a cylinder". *Nuclear Engineering and Technology*, *48*(6), 1315–1320. https://doi.org/https://doi.org/10.1016/j.net.2016.06.011 (pages 61, 117)

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). "Support vector regression machines". In M. C. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems*. MIT Press. https://proceedings.neurips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf. (Page 47)

Fisher, W. D. (1958). "On grouping for maximum homogeneity". *Journal of the American Statistical Association*, *53*(284), 789–798. https://doi.org/10.1080/01621459.1958.10501479 (page 65)

Freund, Y., & Schapire, R. E. (1995). "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting". In P. M. B. Vitányi (Ed.), *Second european conference on computational learning theory (eurocolt-95)* (pp. 23–37). citeseer.nj.nec.com/freund95decisiontheoretic.html. (Page 47)

Friedman, J. H. (1991). "Multivariate adaptive regression splines". *Ann. Statist* (pages 45, 57).

Friedman, J. H. (1993). "Fast mars" (text $\mathrm{LCS}_1 10$). Laboratory for Computational Statistics, Stanford University. https://purl.stanford.edu/vr602hr6778. (Page 60)

Friedman, J. H. (2002). "Stochastic gradient boosting". *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://EconPapers.repec.org/RePEc:eee:csdana:v:38:y:2002:i:4:p:367-378 (page 48)

García Nieto, P., García-Gonzalo, E., Bové, J., Arbat, G., Duran-Ros, M., & Puig-Bargués, J. (2017). "Modeling pressure drop produced by different filtering media in microirrigation sand filters using the hybrid abc-mars-based approach, mlp neural network and m5 model tree". *Computers and Electronics in Agriculture*, *139*, 65–74. https://doi.org/https://doi.org/10.1016/j.compag.2017.05.008 (pages 61, 117)

Géron, A. (2017). "Hands-on machine learning with scikit-learn and tensorflow : Concepts, tools, and techniques to build intelligent systems". O'Reilly Media. https://www.bibsonomy.org/bibtex/2a91270a3a516f4edaa5d459c40317fcc/achakraborty. (Pages 35–40)

Geurts, M. D., & Patrick Kelly, J. (1986). "Forecasting retail sales using alternative models". *International Journal of Forecasting*, *2*(3), 261–272. https://doi.org/https://doi.org/10.1016/0169-2070(86)90046-4 (page 34)

Graham, R., Lawler, E., Lenstra, J., & Kan, A. (1979). "Optimization and approximation in deterministic sequencing and scheduling: A survey". In P. Hammer, E. Johnson, & B. Korte (Eds.), *Discrete optimization ii* (pp. 287–326). Elsevier. https://doi.org/https://doi.org/10.1016/S0167-5060(08)70356-X. (Page 4)

Hardy, M. (1993). "Regression with dummy variables". https://doi.org/10.4135/9781412985628. (Page 28)

Hastie, T., & Tibshirani, R. (1984). "Generalized additive models". (Page 46).

Heerkens, H., & van Winden, A. (2017). "Solving managerial problems systematically" [Translated into English by Jan-Willem Tjooitink]. Noordhoff Uitgevers. (Pages 7, 10, 11).

Ho, T. K. (1995). "Random decision forests". *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282 vol.1. https://doi.org/10.1109/ICDAR.1995.598994 (page 46)

Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory". *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 (page 49)

Hoerl, A. E., & Kennard, R. W. (2000). "Ridge regression: Biased estimation for nonorthogonal problems". *Technometrics*, *42*(1), 80–86. http://www.jstor.org/stable/1271436 (page 44)

Holt, C. C. (2004). "Forecasting seasonals and trends by exponentially weighted moving averages". *International Journal of Forecasting*, *20*(1), 5–10. https://doi.org/https://doi.org/10.1016/j.ijforecast.2003.09.015 (page 34)

Hwang, C., & Yoon, K. (1981). "Multiple Attribute Decision Making" (1st ed.). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-48318-9. (Page 31)

Hyndman, R., & Athanasopoulos, G. (2021). "Forecasting: Principles and practice" (3rd). OTexts. (Pages 33, 34, 39).

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). "A state space framework for automatic forecasting using exponential smoothing methods". *International Journal of Forecasting*, *18*(3), 439–454. https://doi.org/https://doi.org/10.1016/S0169-2070(01)00110-8 (page 34)

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). "An introduction to statistical learning: With applications in r". Springer. https://faculty.marshall.usc.edu/gareth-james/ISL/. (Pages 35–37, 39, 41, 43, 46)

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). "An introduction to statistical learning: With applications in r, second edition". Springer. https://www.statlearning.com/. (Page 49)

Kartal Koc, E., & Bozdogan, H. (2015). "Model selection in multivariate adaptive regression splines (mars) using information complexity as the fitness function". *Machine Learning*, *101*(1), 35–58. https://doi.org/10.1007/s10994-014-5440-5 (pages 61, 117)

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). "Lightgbm: A highly efficient gradient boosting decision tree". In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf. (Page 48)

Kim, B., Khanna, R., & Koyejo, O. O. (2016). "Examples are not enough, learn to criticize! criticism for interpretability". In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf. (Pages 41, 46)

Kuhn, M., & Johnson, K. (2013). "Applied predictive modeling". Springer. https://www.bibsonomy.org/bibtex/2c1847b13c51297368ea3b841fd80fb22/derek-jones. (Page 50)

Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Oueslati, F. S., & Abu-Rub, H. (2021). "A novel stacked generalization ensemble-based hybrid lgbm-xgb-mlp model for short-term load forecasting". *Energy*, *214*, 118874. https://doi.org/https://doi.org/10.1016/j.energy.2020.118874 (pages 65, 120)

Meinshausen, N. (2007). "Relaxed lasso". *Computational Statistics and Data Analysis*, *52*(1), 374–393. https://doi.org/https://doi.org/10.1016/j.csda.2006.12.019 (page 45)

Microsoft Corporation. (2021). "LightGBM 3.3.2.99 documentation". https://lightgbm.readthedocs.io/en/latest/index.html. (Page 58)

Milborrow, S. (2021). "Notes on the earth package". http://www.milbo.org/doc/earth-notes.pdf. (Page 70)

Miller, T. (2017). "Explanation in artificial intelligence: Insights from the social sciences". *CoRR*, *abs/1706.07269*. http://arxiv.org/abs/1706.07269 (page 41)

Molnar, C. (2019). "Interpretable machine learning: A guide for making black box models explainable". (Page 41).

Probst, P., Bischl, B., & Boulesteix, A.-L. (2018). "Tunability: Importance of hyperparameters of machine learning algorithms". https://doi.org/https://doi.org/10.48550/arXiv.1802.09596. (Pages 63, 118)

Rashmi, K. V., & Gilad-Bachrach, R. (2015). "DART: dropouts meet multiple additive regression trees". *CoRR*, *abs/1505.01866*. https://dblp.org/rec/journals/corr/RashmiG15.bib (page 48)

Reinsch, C. (1967). "Smoothing by spline functions" [cited By 1412]. *Numerische Mathematik*, *10*(3), 177–183. https://doi.org/10.1007/BF02162161 (page 45)

Rijksinstituut voor Volksgezondheid en Milieu. (2021). "Griep feiten en cijfers". Retrieved November 23, 2021, from https://www.rivm.nl/griep-griepprik/feiten-en-cijfers. (Page 55)

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors". *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0 (page 49)

Schiele, J., Koperna, T., & Brunner, J. O. (2021). "Predicting intensive care unit bed occupancy for integrated operating room scheduling via neural networks". *Naval Research Logistics (NRL)*, *68*(1), 65–88. https://doi.org/https://doi.org/10.1002/nav.21929 (pages 33, 100)

Smola, A. J., & Schölkopf, B. (1998). "A tutorial on support vector regression". (Page 47).

Stigler, S. (1986). "The history of statistics: The measurement of uncertainty before 1900". Harvard University Press. https://books.google.nl/books?id=M7yvkERHIIMC. (Page 44)

Tang, M., Zhao, Q., Ding, S. X., Wu, H., Li, L., Long, W., & Huang, B. (2020). "An improved lightgbm algorithm for online fault detection of wind turbine gearboxes". *Energies*, *13*(4). https://doi.org/10.3390/en13040807 (pages 65, 120)

Taylor, J. W. (2003). "Exponential smoothing with a damped multiplicative trend". *International Journal of Forecasting*, *19*(4), 715–725. https://doi.org/https://doi.org/10.1016/S0169-2070(03)00003-7 (page 34)

Thomas, J., Coors, S., & Bischl, B. (2018). "Automatic gradient boosting". https://doi.org/https://doi.org/10.48550/arXiv.1807.03873. (Pages 63, 118)

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso". *Journal of the Royal Statistical Society Series B Methodological*, 267–288. https://www.bibsonomy.org/bibtex/215bd0e5f71c50ccd52bc5d3f09a58386/aude.hofleitner (page 44)

Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T. T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K.-C., Selkrig, J., Zhou, T., Song, J., & Lithgow, T. (2018). "Bastion3: a two-layer ensemble predictor of type III secreted effectors". *Bioinformatics*, *35*(12), 2017–2028. https://doi.org/10.1093/bioinformatics/bty914 (pages 65, 120, 121)

Wang, X., & Wang, J. (2013). "Scheduling problems with past-sequence-dependent setup times and general effects of deterioration and learning" [cited By 26]. *Applied Mathematical Modelling*, *37*(7), 4905–4914. https://doi.org/10.1016/j.apm.2012.09.044 (page 55)

Wang, Y., & Ni, X. S. (2019). "A xgboost risk model via feature selection and bayesian hyper-parameter optimization". https://doi.org/https://doi.org/10.48550/arXiv.1901.08433. (Pages 63, 118)

Westerink, M. (2021). "Online operational planning in a multi-provider ambulatory allied healthcare organisation : Developing an online appointment scheduling support procedure employing predictive modelling of location occupancy." http://essay.utwente.nl/86198/. (Pages 1, 2, 4–7, 10, 12, 15, 21, 27–29, 31, 32, 39, 50, 52, 53, 69, 76, 77, 79–82, 85, 86, 88, 90, 92, 93, 96–98, 123, 125)

Winters, P. R. (1960). "Forecasting sales by exponentially weighted moving averages". *Management Science*, *6*(3), 324–342. https://EconPapers.repec.org/RePEc:inm:ormnsc:v:6:y:1960:i:3:p:324-342 (page 34)

World Health Organization. (2021). "Global influenza surveillance and response system (gisrs)". Retrieved November 23, 2021, from https://www.who.int/initiatives/global-influenza-surveillance-and-response-system. (Page 55)

XGBoost developers. (2022). "Xgboost documentation - 1.6.0". https://xgboost.readthedocs.io/en/latest/index.html. (Page 58)

Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). "A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring". *Expert Systems with Applications*, *78*, 225–241. https://doi.org/https://doi.org/10.1016/j.eswa.2017.02.017 (pages 63, 118)

Zhang, J., Mucs, D., Norinder, U., & Svensson, F. (2019). "Lightgbm: An effective and scalable algorithm for prediction of chemical toxicity–application to the tox21 and mutagenicity data sets". *Journal of Chemical Information and Modeling*, *59*(10), 4150–4158. https://doi.org/10.1021/acs.jcim.9b00633 (pages 65, 120, 121)

Zhou, J., Qiu, Y., Zhu, S., Armaghani, D. J., Khandelwal, M., & Mohamad, E. T. (2021). "Estimation of the tbm advance rate under hard rock conditions using xgboost and bayesian optimization". *Underground Space*, *6*(5), 506–515. https://doi.org/https://doi.org/10.1016/j.undsp.2020.05.008 (pages 63, 118)

Zou, H., & Hastie, T. (2005). "Regularization and variable selection via the elastic net". *Journal of the Royal Statistical Society Series B Statistical Methodology*, *67*(2), 301–320. http://www.jstor.org/stable/3647580 (page 45)

A From sessions to utilization

This section describes section 2.1 more in depth. The section describes how sessions are constructed from data, and how session utilization can be measured from the sessions, in such a way that tidy sessions remain, with valid utilization.

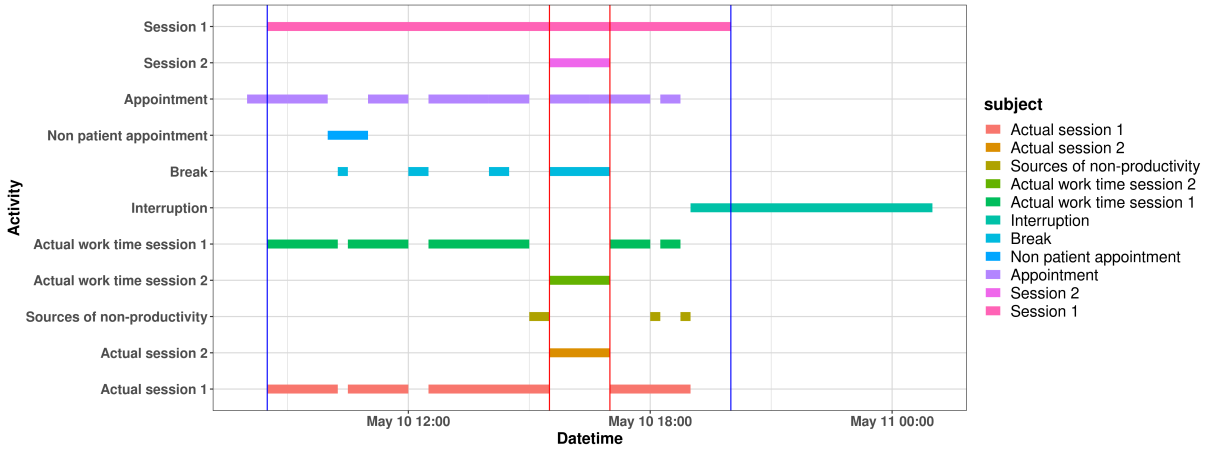

Figure A.1: Overview of the session, and the activities it includes

Figure A.1 depicts the way the organization designed the session, appointments and breaks system. A short explanation is in place. In figure A.1, there are the two sessions, named 1 and 2. Session 1 is at location X and session 2 is at location Y. session 2 overlaps completely with session 1. Session 1 has a break during the hours that session 2 is active so that the employee can adhere to session 2. The appointments that are scheduled during session 2 should be assigned to session 2, but there is no explicit link. Both sessions can have their own breaks. In this particular case, session 2 does not have breaks. The breaks belong to session 1. The non-patient appointment also belongs to session 1. It overlaps with a break, the break time is completely subtracted from the schedule, but the non-patient appointment contributes to productive time. The first appointment starts before the session starts. Recall that we define productive time such that only the time that falls within the session is counted, hence the part that lies outside of the session does not contribute towards utilization. Session 1 is demarcated by the blue vertical lines, and the productive hours are 'actual work time session 1' and 'actual work time session 2'. Session 2 has a utilization of 1. Session 1 has a lower utilization due to non-productive time. There are three sources. The first one is just before session 2 starts. Session 2 is demarcated by the red vertical lines. The second source of non-productivity is just before the last appointment. The last source is the time between the final appointment and the start of the interruption. The line 'actual session 1' continues there, but there is no appointment. The idea is that the practitioner is not productive there and therefore we consider this to be non productivity. Finally, there is an interruption. Recall that the interruption is employee bound, and not bound to a session. Session 1 overlaps with the interruption, hence actual work time session 1 is cut off at the start of the interruption. Session 2 does not overlap with the interruption, and therefore the interruption is not of importance to session 2.

Appendix B illustrates how we get from the database to a structure that is usable for analyses. The illustration contributes to the assessment of the validity of the study. From the database we construct metadata, which we continue the study with.

B Data extraction, transformation and loading

The architecture of the database used by the organization is limited in its usability with respect to the analysis of the sessions. One of the issues is that there is no explicit connection between the appointments and sessions. As a result, we can only assume that some of the appointments are connected to a session and this becomes problematic when multiple sessions are on the same date. This section explains how the raw data originating from the database is processed towards usable variables.

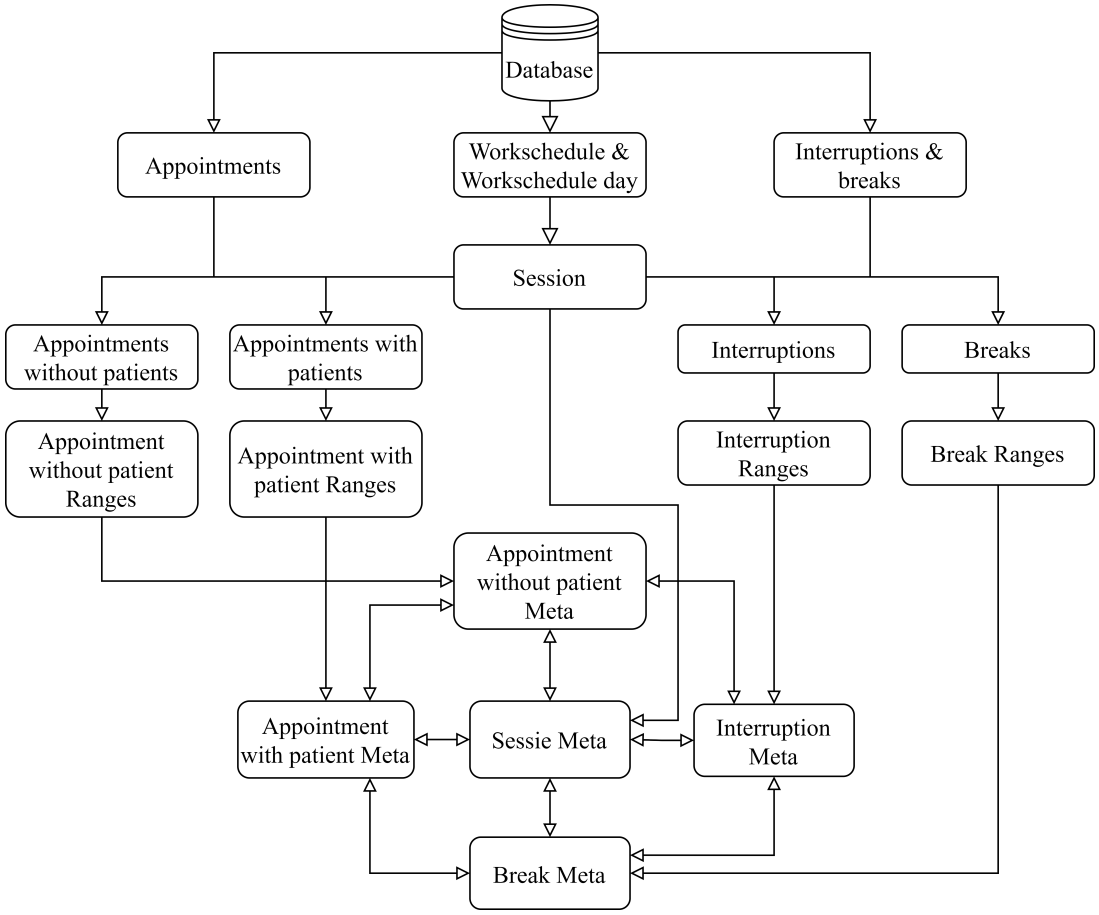

Figure B.1: The ETL process, how we get from the database to usable data

Illustrated by figure B.1, we take the session as a starting point. Recall that the session is the combination of an employee, a location, and a date. The combination of these is the unique identifier. As an example, 20150915-010-008 is a session on September the 15th, 2015, for the employee with id 10 at the location with id 8). It is however not straightforward to obtain the session key, as sessions are not explicitly defined as well, but exist in so called work schedules. The work schedule defines a range of dates at which a session is active. The session in this case defines the start and end time of the day. The work schedule can span one day, but also multiple years. The work schedule is always defined for a location, an employee, a day of the week, and an even or odd week. Home visits are not considered as home visits are scheduled at once, and do not use predicted utilizations. Other entities that are not considered are employees that represent a location, such as a store. The reason that these exist is software related. We also do not consider

retirement homes, as appointments at these locations are, similar to the home visits, scheduled at once, and therefore do not follow the process we try to model.

We obtain the session keys by iterating over a date range starting on the first observed work schedule date, and the current date, but it also occurs that multiple work schedules on the same location and date are active. This is either a scheduled overruling or an issue with the scheduling system. In the latter case, the most recent session is chosen.

We then obtain the breaks, interruptions and appointments. The appointments data should be subdivided into appointments involving a patient, or not. The former is found by joining on the patients, the latter by anti joining on the patient. Breaks and interruptions originate from the same source, and we distinguish between the two. Breaks are connected to the work schedule, interruptions are connected to the employee. Breaks never exceed one day and are not explicitly defined for a date but inherit the date from the work schedule. Interruptions might span multiple days, and have therefore a defined date range.

For the appointments, interruptions and the breaks, we construct ranges. A range is an extended period of the day on which one of the activities takes place. For instance, if a session contains four appointments, with the following intervals [09:00; 10:00], [10:00; 11:00], [11:00;12:00], [13:00; 14:00] this is reduced to one range from 09:00 to 12:00 and one range from 13:00 to 14:00. One advantage is that analysis takes considerably less time due to that there are fewer entries to process. The other advantage is that overlapping activities of the same type are considered to be in the same range and thereby does not occur. The example is illustrated by figure B.2.

Appointment id	Start time	End time
1	09:00	10:00
2	10:00	11:00
3	11:00	12:00
4	13:00	14:00

Appointment Ranges	Start time	End time
1	09:00	12:00
2	13:00	14:00

Figure B.2: An illustration of how sessions relate to the work schedule

Using the ranges and the sessions, metadata is constructed. The metadata for each of the activities appointments, breaks or interruptions, comprise among other things an overview of a session in terms of the overlap with the other activities, and the overlap with the session. The total time by which an activity overlaps with the session can be seen as the total number of productive hours during the session, but should be corrected for non productive hours. The metadata also contains the total number of activity types per session, and the sub specification of the type of activity. For instance the number of patient bound activities, and the total number of new patients activities. The average access time of the appointments in the session is also stored.

C From sessions to utilization

In this section we show how the individual components (interruption, breaks, patient appointments and non-patient appointments are transformed into utilization) in a structured way.

We define the following time data sets, which all overlap entirely with the length of the session.

$$B := Break$$

$$I := Interruption$$

$$P := Patient\,appointment$$

$$NP := Non\,patient\,appointment$$

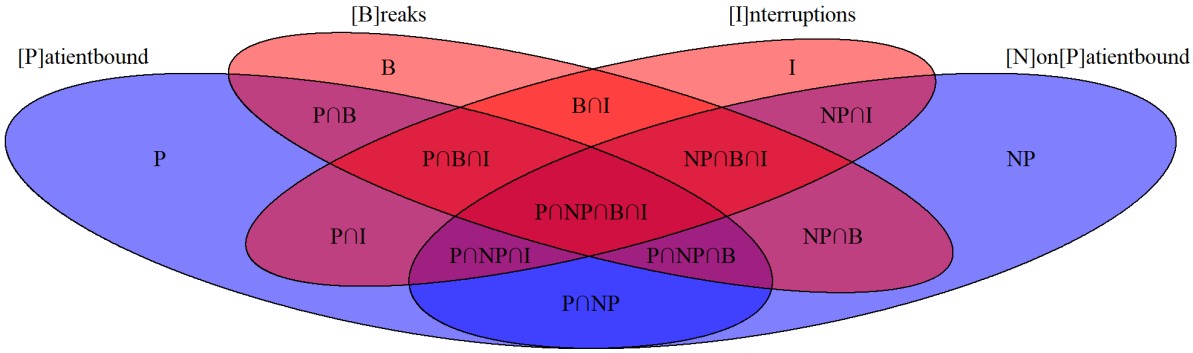

Figure C.1: Venn diagram representing the activities found at the organization. Blue: productive time, red: non-productive time

Figure C.1 shows a Venn diagram that represents the components from which we determine the utilization. The diagram helps visualizing the areas we wish to construct utilization from, and which we leave out. The blue area represents productive time, the red area the breaks and interruptions. All overlap the session. To obtain the session utilization, we need to account for the overlapping time.

$$Session\ utilization = \frac{Productive\ time}{Session\ length} \tag{1}$$

Following (1) the numerator, productive time, is composed of three components that together define the productive time. These are:

Practice time, the time spent on treating patients and the time spent on side activities, but only once the time that both occur, which is already reflected in P and NP:

$$P + NP - P \cap NP \tag{2}$$

Breaks and interruptions during patient appointments, accounted for both occurring at the same time:

$$P \cap B + P \cap I - P \cap B \cap I \tag{3}$$

Breaks and interruptions during non-patient appointments, accounted for both occurring at the same time, and either one occurring during a patient appointment:

$$NP \cap B - P \cap NP \cap B + NP \cap I - NP \cap B \cap I - P \cap NP \cap I + P \cap NP \cap B \cap I \quad (4)$$

Combining the components:

$$Productive\,time = (2) - (3) - (4)$$

The session length is the time between the start and end time of the session (S), without the breaks and interruptions, compensated for breaks and interruptions occurring at the same time:

$$Session\,length = S - B - I + B \cap I \quad (5)$$

D Systematic literature review

Figure D.1: The systematic literature review

E List of predictors with description

Predictor	Description
Date of appointment request	The date on which the appointment was made.
Date of appointment	The date of the appointment.
Access time	The time between scheduling and the appointment.
Access time aggregation	The average access time per session, region, location, nation.
Access time variance aggregation	The variance per session, region, location, nation.
Travel time	The patients' estimated travel time to location.
Average travel time aggregation	The average estimated patients' travel time to location per session, region, location, nation.
Patient age	The age of the patients.
Average patient age aggregation	The average age of the patients in the session, location, region, nation on the appointment date at the time of scheduling.
Session start time	the time at which the session starts.
Session end time	the time at which the session ends.
Is single session	Whether or not the session is for a single day.
Day of the week	The day of the week that the session is at.
Is odd week	Whether or not the week is odd.
Employee id	The id that specifies the employee.
Location id	The id that specifies the location.
Region id	The id that specifies the region.
Number of breaks	The number of breaks in the session.
Total break time	The sum of all break time in the session.
Ratio break time	The sum of all break time in the session, divided by the session length.
Number of interruptions	The number of interruptions in the session.
Total interruption time	The sum of all interruption time in the session.
Ratio interruption time	The sum of all interruption time in the session, divided by the session length.
Number of appointments without patient	The number of appointments w/o patient in the session.
Total time of appointments without patient	The sum of all appointments w/o patient time in the session.
Ratio of appointments without patient time	The sum of all appointments w/o patient time in the session, divided by the session length.
Year	The year of the session.
Quarter	The year quarter of the session.
Month	The month number of the session.
Week	The week number of the session.
Yearmonth	The combination of year and month of the session.
Yearweek	The combination of year and week of the session.

Utilization of appointment aggregation	The current utilization of the session/location/region/nation at time of scheduling per session.
Session length	The length of the session.
Vacation factor	The intensity factor of vacations of the session.
is 7 days after public holiday	Whether or not the session date is 7 days after a public holiday.
is 14 days after public holiday	Whether or not the session date is 14 days after a public holiday.
is Christmas eve	Whether or not the session date is on Christmas eve (Dec., 24th).
is 31st of December	Whether or not the session date is the last day of the year (Dec., 31st).
is Sinterklaas eve	Whether or not the session date is is on Sinterklaas eve feast (Dec., 12th).
is dead remembrance day	Whether or not the session date is on death remembrance day (May, 4th).
is liberation day	Whether or not the session date is on liberation day (May, 5th).
Total influenza	Reported number of new influenza cases that was available at the time of scheduling the appointment.
Lagged total influenza	Reported number of new influenza cases that was one weekbefore the time of scheduling the appointment.

F Mars parameter tuning

		Study							
		García Nieto et al., 2017		Kartal Koc et al., 2015		Boehmke et al., 2019		Dey et al., 2016	
Parameter	**Range**	LB	UB	LB	UB	LB	UB	LB	UB
Max terms	[0, ∞)	3	200		21	2	200	10	40
Penalty	[-1, ∞)	-1	4					0	4
Interactions	[0, ∞)	1	6		2	1	3	2	4
Threshold	[0, 1]								10^{-4}

Table F.1: Mars parameter optimization studies with corresponding parameter bounds

Table F.1 contains Mars parameters found in literature, that were tuned in corresponding studies. The values consist of a lower bounds (LB) and upper bounds (UB) for the parameter. An empty cell means that the study does not report the usage of the parameter. Cells with an equal value for the LB and UB means that the study reports that it used that exact value. If only a single value is mentioned, only either an upper bound or a lower bound is mentioned.

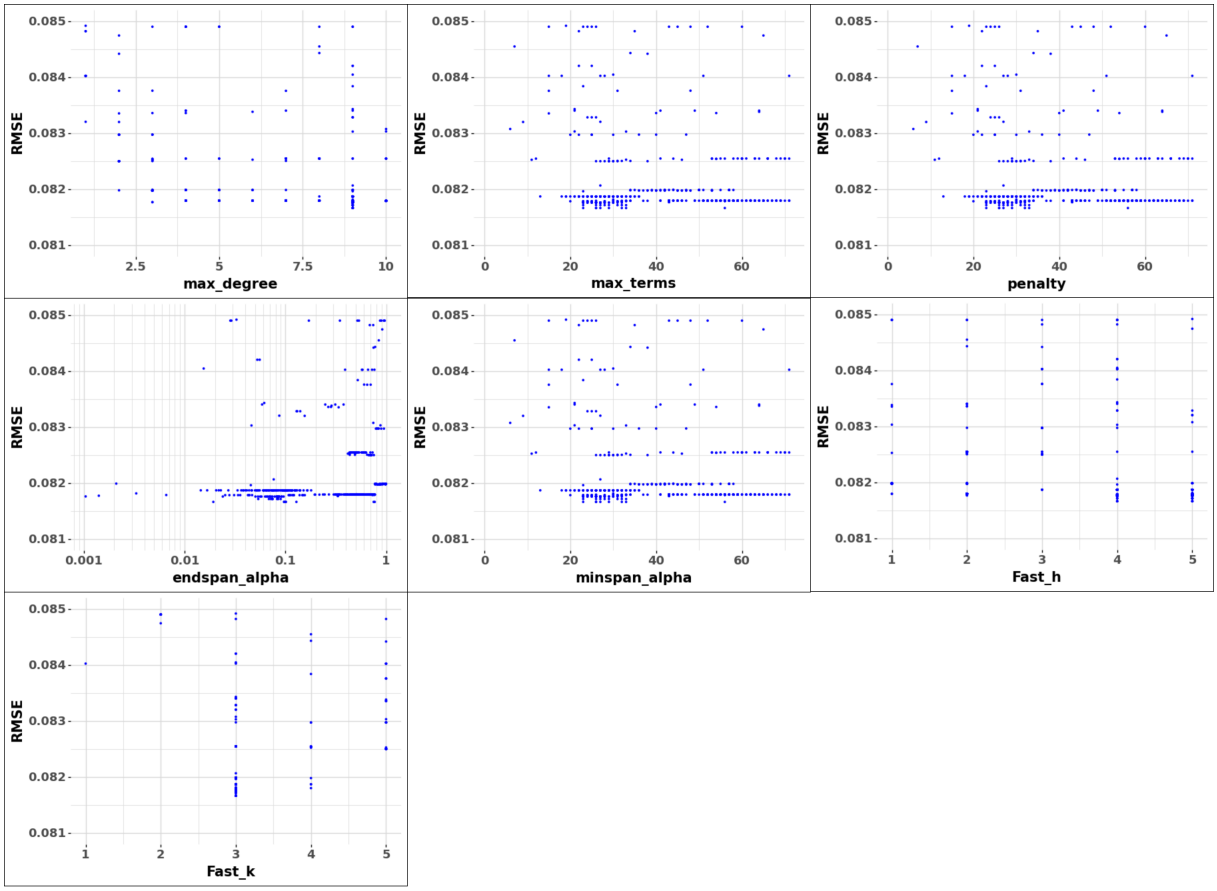

Figure F.1: Results in rmse for the corresponding hyperparameter

G XGBoost parameter tuning

		Study											
		Xia et al., 2017		**Probst et al., 2018**		**Thomas et al., 2018**		**Y. Wang et al., 2019**		**Zhou et al., 2021**		**Anghel et al., 2018**	
Parameter	Range	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
Nrounds	0	60	60	1	5000					1	150	16	1000
Learning rate	[0, 1]	0.1	0.1	2^{-10}	2^{0}	0.001	0.2	0.005	0.2	10^{-5}	1	0.01	1
Subsample	[0, 1]	0.9	1	0.1	1	0.5	1	0.8	1				
Max depth	[0, ∞)	1	12	1	15	3	20	5	30	1	15	2	14
Max delta step	[0, ∞)	0	1										
Min child weight	[0, ∞)	0	4	2^{0}	2^{7}			0	10				
Colsample bytree	(0, 1)	0.9	1	0	1	0.5		0.8	1			0.01	1
Colsample bylevel	(0, 1)					0.5							
Lambda	[0, ∞)			2^{-10}	2^{-10}	2^{-10}	2^{10}			1	15		
Alpha	[0, ∞)			2^{-10}	2^{-10}	2^{-10}	2^{10}			1	15		
Gamma	[0, ∞)	0	0.01			2^{-7}	2^{6}	0	0.2			10^{-2}	10^{5}

Table G.1: XGBoost parameter optimization studies with corresponding parameter bounds

Table G.1 contains XGBoost parameters found in literature, that were tuned in corresponding studies. The values consist of a lower bounds (LB) and upper bounds (UB) for the parameter. An empty cell means that the study does not report the usage of the parameter. Cells with an equal value for the LB and UB means that the study reports that it used that exact value. Studies regularly mention the usage of a log_2 scale. Therefore, these values denoted as an exponent with base 2. If only a single value is mentioned, only either an upper bound or a lower bound is mentioned.

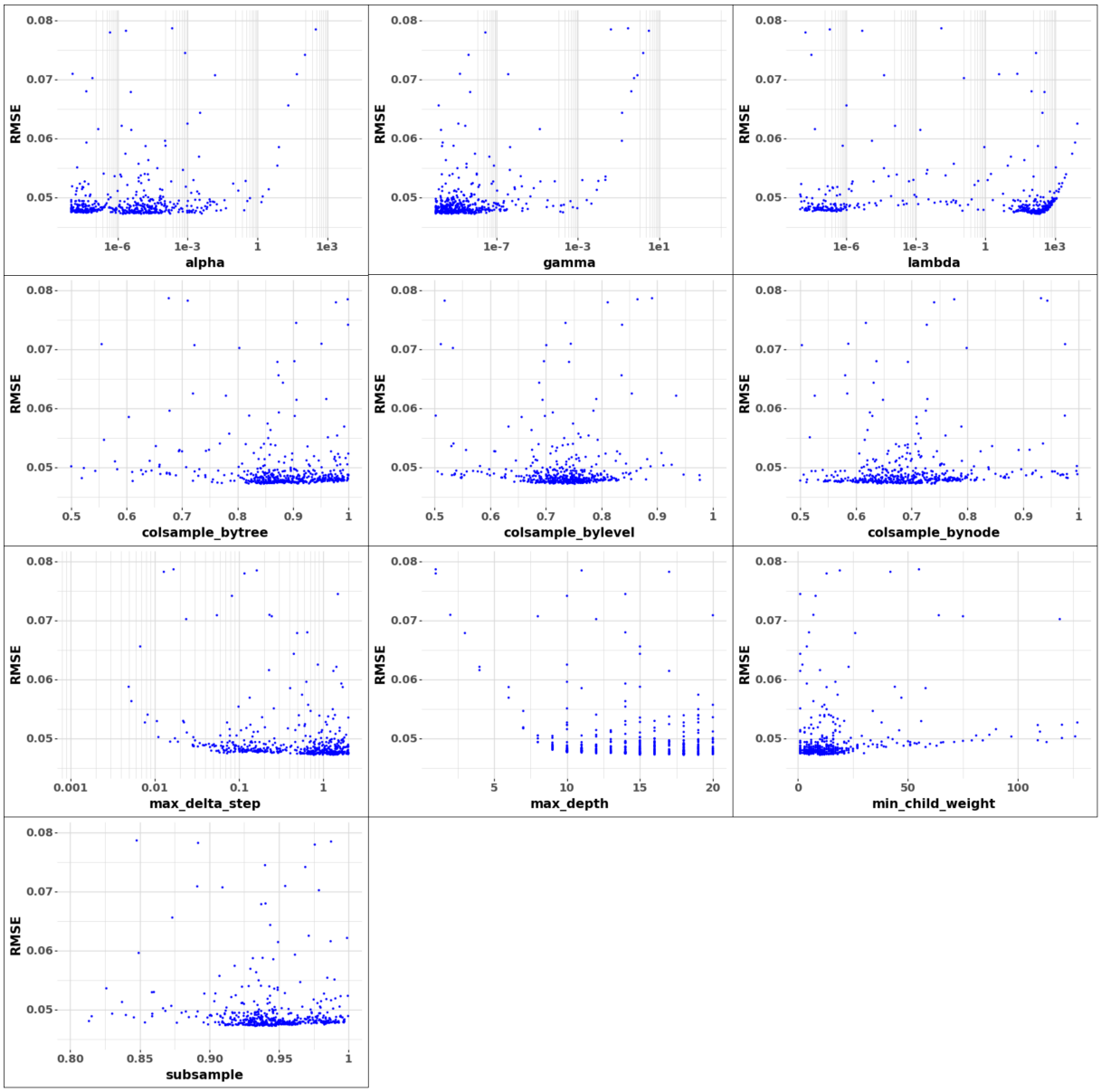

Figure G.1: Results in rmse for the corresponding hyperparameter

Figure G.1 shows the parameter tuning process for the XGBoost model. The resulting cross validated rmse is shown for every configuration of parameters. Some parameters show a clear range of optimal parameter values such as the bagging frequency and bagging fraction. This is also observed for the number of leaves, max depth. min child weight and min data in leaf. The Lambda l_2 parameter colsample bynode do not seem to have much influence as no clear decrease is seen for the range of variables, but the lambda l_1 does to some extend. Also, it seems that we could have reduced the maximum value for the min child weight parameter as the optimal value for the parameter seems to lie far lower.

H LightGBM parameter tuning

		Study											
		J. Wang et al., 2018		Massaoudi et al., 2021		Zhang et al., 2019		Anghel et al., 2018		Dev et al., 2019		Tang et al., 2020	
Parameter	**Range**	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
Nrounds	$0 \geq$			100	3000	50	900	16	1000	1500	2000		
Learning rate	> 0	2^{-10}	2^{-1}	10^{-3}	1	10^{-6}	10^{-1}	0.01	1	0.01	0.2	0.01	1
Subsample	(0, 1]					0.8	0.8					0.5	1
Max depth	$[0, \infty)$	5	10	-2	12	5	12	2	14	1	11	3	20
Min child weight	$[0, \infty)$	0	0.02										
Feature fraction	(0, 1]	0.5	1					0.01	1			0.5	1
Lambda l_1	$[0, \infty)$	0	0.01										
Lambda l_2	$[0, \infty)$	0	0.01										
Num leaves	[1, 131072)	50	800			30	500			3	90	8	40
Min data in leaf	$[0, \infty)$	2^1	2^6			30	500						
Max bins	$(1, \infty)$	2^5	2^{10}			250	500						

Table H.1: LightGBM parameter optimization studies with corresponding parameter bounds

Table H.1 contains LightGBM parameters found in literature, that were tuned in corresponding studies. The values consist of a lower bounds (LB) and upper bounds (UB) for the parameter. An empty cell means that the study does not report the usage of the parameter. Cells with an equal value for the LB and UB means that the study reports that it used that exact value. Studies regularly mention the usage of a log_2 scale. Therefore, these values denoted as an exponent with base 2. If only a single value is mentioned, only either an upper bound or a lower bound is mentioned.

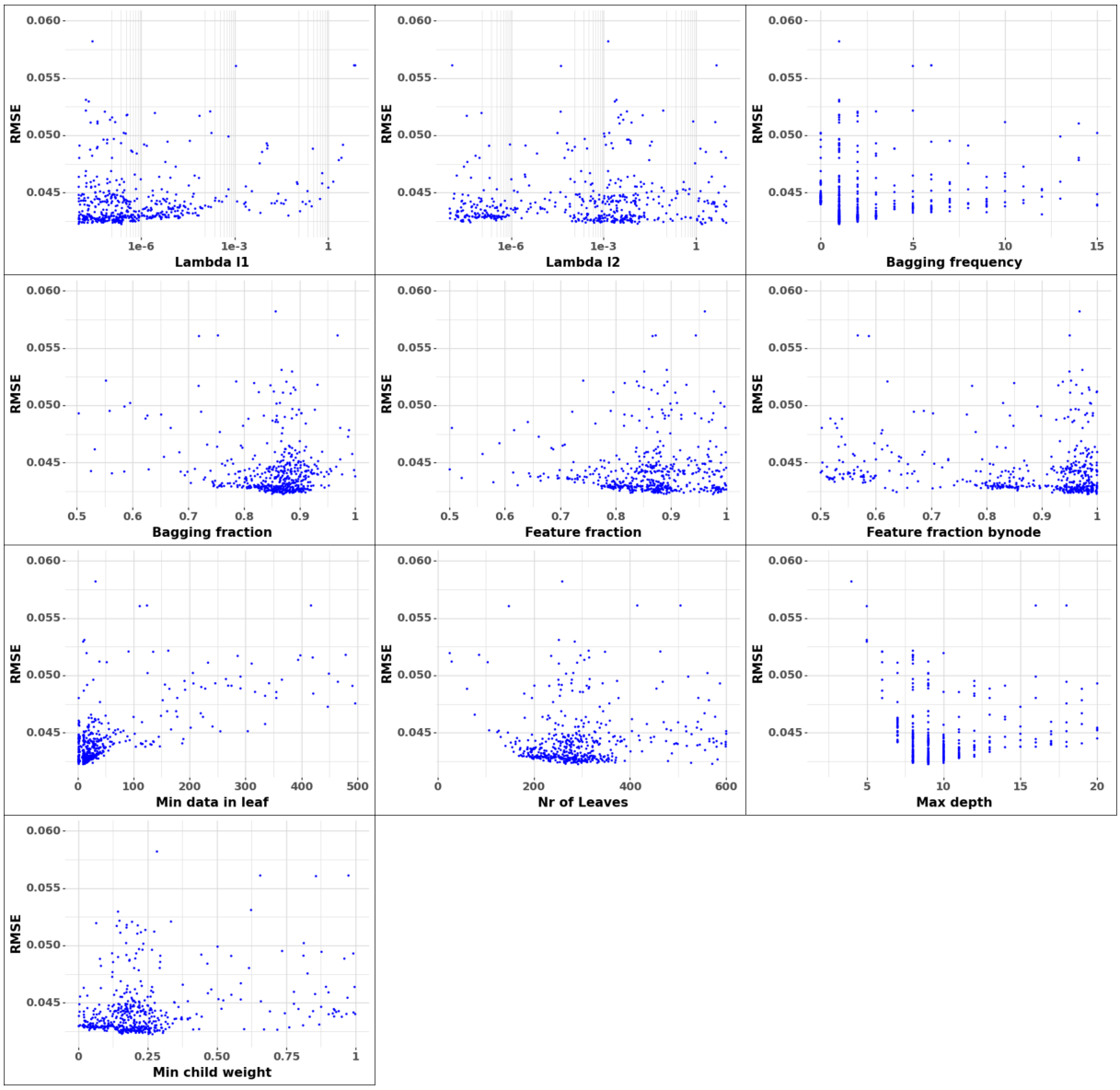

Figure H.1: Results in rmse for the corresponding hyperparameter

Figure H.1 shows the tuned parameters and the corresponding rmse value on each of the configurations. Lambda L2 does not have major influence on the rmse values. The rmse values vary uniformly over the range of Lambda L2 values. The max depth parameter shows an example of the bias-variance tradeoff. Deeper models are able to fit a training set better, but the resulting model can also overfit the training set. First, we observe a decrease in the cross validated rmse values when the depth of the model increases, but after a maximum depth of 9, the model tend to overfit the training data and we start to observe an increase in cross validated rmse values. We observe the same with the parameters min data in leaf, the number of leaves, the bagging fraction and the bagging frequency. In fact, the range of min data in leaf could have been much lower. For this parameter, our results are more in line with J. Wang et al., 2018 than with Zhang et al., 2019. The difference can possibly be attributed to the number of observations available for the underlying predictors.

I Simulation results

Exp Nr	Model	Access time	Distance	Days deviation	Access time violation	Utilization		Fragmentation
		μ	μ	μ	μ	μ	σ	μ
1	Logistic	14.795	3.596	5.328	0.121	0.612	0.288	2.052
	LightGBM	15.574	3.484	5.981	0.154	0.612	0.291	2.005
2	Logistic	13.833	2.918	3.993	0.097	0.615	0.301	1.896
	LightGBM	13.821	2.905	4.095	0.103	0.616	0.306	1.865
3	Logistic	15.737	4.829	6.053	0.196	0.608	0.283	2.153
	LightGBM	19.054	4.629	7.133	0.436	0.609	0.279	2.135
4	Logistic	18.998	79.234	8.324	0.602	0.603	0.268	3.194
	LightGBM	25.207	79.543	10.186	1.297	0.603	0.259	3.213
5	Logistic	12.925	8.156	4.259	0.015	0.610	0.287	2.208
	LightGBM	13.142	8.231	4.563	0.015	0.612	0.286	2.166
6	Logistic	11.835	4.393	2.771	0.010	0.615	0.294	2.148
	LightGBM	11.275	4.214	2.786	0.012	0.615	0.297	2.101
7	Logistic	11.574	3.359	2.549	0.020	0.615	0.305	2.020
	LightGBM	11.317	3.332	2.554	0.015	0.616	0.310	1.973

Table I.1: Simulation results on all parameters

J Implementation report

This report technically describes the minimal steps to take in order to successfully implement the prediction model, more specifically the LightGBM model at the organization as proposed in this study. The report should be read as an independent report. Therefore the report walks through some concepts that have also been defined and described in this report. The report first briefly describes the process and will cover the topics more in-depth later.

Service desk

We name the scheduling methods taking place at organization the system. The first line of contact is the service desk. The service desk is responsible for scheduling patient appointments using the scheduling software.

Session

Secondly, practitioners see the patient on the proposed combination of date, time and location. The unit of measure that is the combination of (Practitioner, Location and Date) is called the Session. By that, the session may or may not be the only session of a practitioner on one date and location. Hence, the practitioner can be present at two locations on the same date. However, it is safe to assume that one practitioner is only present once at the same location and on the same date.

Session state

Consider yourself at the service desk. A patient calls and wants to make an appointment. In this report, this is the state of the system. That is, every time a patient calls to make an appointment is when predictions are required.

Why is that the case? The study Westerink, 2021 recommends that patients are scheduled using a Topsis scheduling methodology, in which alternative appointment options are ranked on a multi-criteria scale in order of best to worst. Part of that implementation is the prediction of session (as defined) utilization, as a measure of the 'crowdedness' of a session.

For session utilization, we define:

- The session length is known as the total time that was scheduled for seeing patients or non-patient bound appointments. That means that the non-overlapping break time (pauze, bound to the work schedule) and interruption time (bound to the employee) occurring through the session is not counted.
- Productive time is then the non-overlapping time spent on patient or non-patient bound appointments within a session.
- Then session utilization is the ratio of productive time and session length.

Non-overlapping means that during the time, one event takes place at most. That is, when both a break and an interruption take place at the same time (or for that matter two breaks at once), the time is not doubled, but only counted once. If a break and an appointment take place at the same time, the time is considered not to be scheduled for

appointments but for breaks and therefore not counted. Should two appointments take place simultaneously, the time is productive only once.

One of the lesser-known challenges in supervised learning methods such as LightGBM, is that it requires the same non-recurrent individual training observations to prevent cycling and thereby overfitting. These observations should therefore also be on the level of the patient appointment, and not on that of the session, which is an aggregation of appointments. Therefore we cannot use session utilization (which we do aim to predict) in the training observations, better known as labels. To overcome this problem, we transform session utilization (at the time of the appointment) into total session utilization added after the appointment. This measure is supposed to be unique for each of the appointments. This is simply

$$\textit{Added utilization} = \textit{Final utilization} - \textit{Current utilization (incl. appointment)} \tag{6}$$

When we predict added session utilization, we can simply add the current level of session utilization in order to construct a predicted value for the final session utilization. This is what the LightGBM model is trained to do, and works remarkably well with a cross-validated root mean squared error of 0.041 (R2 = 0.982).

The LightGBM model has been trained with 71 predictors (also known as features), and in this report we found that the best performing predictors are

- Current level of session utilization including the currently proposed appointment.
- Ratio of session time used for non-patient bound appointments (within the organization also known as Grijze blokken, Grey Blocks).
- Employee ID.
- Location ID.

Employee ID and Location ID are the primary keys from the respective tables 'medewerker' and 'vestiging' as found in the database. The current level of session utilization including the currently proposed appointment is the level of utilization seen just after the appointment is scheduled in the session. Events that alter this level are therefore the creation or removal of appointments from the session. The ratio of non-patientbound appointments is basically the part of utilization coming from the non-patientbound appointments. It is the ratio the total session length that is scheduled with non-patientbound appointments.

These are the four data requirements that should be provided at the time of prediction.

We consider the prediction method to be an independent module, depicted in figure J.1. The data requirements are input to the prediction method module, and the prediction of utilization (defined on the unit interval) is output.

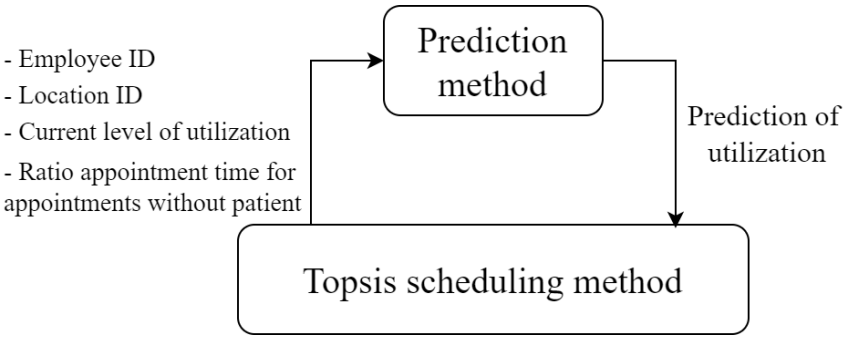

Figure J.1: Prediction method in context of Topsis scheduling method

The implementation of the Topsis scheduling method is described extensively in Westerink, 2021.

LightGBM

One of the qualities of LightGBM is that it comes with a shipped implementation framework. The installation guide can be found here https://lightgbm.readthedocs.io/en/latest/Installation-Guide.html.

LightGBM has an R, Python and C API which makes implementation relatively easy. For example, implementation from a pre-trained model has examples here https://github.com/microsoft/LightGBM/issues/2397.

The model file originating from a trained model is just a text (*.txt) file with specifications for each tree node, along with some metadata like the objective of the optimization model, in this case obviously regression. Loading the model is therefore simply specifying the location of the model file, using the **LGBM_BoosterCreateFromModelfile** function.

The next step is to predict values. Since it is the idea to order a list of n (currently specified at 30) appointments, the easiest way to do this is to combine n predictor vectors originating from the n appointments into an n x m matrix, where m is the cardinality of the vector, id est the number of predictors, currently specified at 4. Predictions can be made using the **LGBM_BoosterPredictForMat** which uses a pointer to the loaded model and to the matrix with predictors.

Since we know from that the organization works with C#, we consider that as well. There is no official LightGBM wrapper for C#, but due to LightGBMs open nature, there are many implementations. Therefore a second and arguably easier option from the perspective of C# / .net is to use Microsofts ML.net framework, which has built-in models for LightGBM, and can load model.zip files originating from Microsoft's nimbusML python library, which provides Python bindings for ML.net. This means that model research and development can be done in the same context as implementation. Now follows a fairly simple and straight forward example class of what that would look like

Listing 1: C# LightGBM prediction class

```
1 using Microsoft.ML;
2 using Microsoft.ML.Data;
3 using MySqlConnector;
4 using Microsoft.Data.Analysis;
```

```
5  using System.Linq;
6
7
8  namespace LightGBM_CSHARP_TEST
9  {
10     public class AppointmentData
11     {
12         public float utilization_w_current_session { get; set; }
13         public float ratio_afspraak_zonder_patient { get; set; }
14         public long medewerker_id { get; set; }
15         public long vestiging_id { get; set; }
16         public double util_added { get; set; }
17     }
18     public class UtilizationPrediction
19     {
20         [ColumnName("Score")]
21         public float utilizationPrediction { get; set; }
22     }
23
24     class LightGbmPredictor
25     {
26         public static void predict_utilizations(string
               databaseConnectionString)
27         {
28             // Create ML context
29             MLContext mlContext = new MLContext();
30
31             // Load model
32             string model_path = @"path/to/lightgbm/model.zip"';
33
34             DataViewSchema modelSchema;
35             ITransformer trainedModel = mlContext.Model.Load(
                   model_path, out modelSchema);
36
37             // Load data (nxm matrix, n rows, m predictors)
38             DatabaseSource source = new DatabaseSource(
39                 MySqlConnectorFactory.Instance,
40                 databaseConnectionString,
41                 "SELECT " +
42                 "utilization_w_current_session, " +
43                 "ratio_afspraak_zonder_patient, " +
44                 "medewerker_id, vestiging_id, " +
45                 "util_added " +
46                 "FROM prediction_matrix");
47             DatabaseLoader loader = mlContext.Data.
                   CreateDatabaseLoader<AppointmentData>();
48             IDataView prediction_matrix = loader.Load(source);
49
50             // Predict added utilizations
51             IDataView predictions = trainedModel.Transform(
                   prediction_matrix);
52             float[] actual_predictions = predictions.GetColumn<float
                   >("Score").ToArray();
53
54             PrimitiveDataFrameColumn<float> prediction_arr =
55                 new PrimitiveDataFrameColumn<float>("predicted");
56             for (var i = 0; i < actual_predictions.Length; i++)
57             {
```

```
58                  prediction_arr.Append(actual_predictions[i]);
59              }
60
61              // Transform added utilizations to utilizations
62              DataFrame prediction_matrixDf = prediction_matrix.
                    ToDataFrame(-1);
63              prediction_matrixDf.Columns.Add(prediction_arr);
64              DataFrameColumn current_util = prediction_matrixDf
65                  .Columns.GetSingleColumn("
                        utilization_w_current_session");
66              prediction_matrixDf["predicted"].Add(current_util,
                    inPlace: true);
67              prediction_matrixDf["predicted"].Clamp(0, 1, inPlace:
                    true);
68              //DataFrameColumn dataColumn = prediction_matrixDf["
                    util_added"].Add(actual_predictions, inPlace: false);
69              System.Console.WriteLine("Done.");
70          }
71          public static float predict_utilizations(
72              float current_util, float ratio_no_patient_aptmnt, int
                    medewerker_id, int vestiging_id)
73          {
74              // Create ML context
75              MLContext mlContext = new MLContext();
76
77              // Load model
78              string model_path = @"path/to/lightgbm/model.zip"';
79
80              DataViewSchema modelSchema;
81              ITransformer trainedModel = mlContext.Model.Load(
                    model_path, out modelSchema);
82
83              // Load data (nxm matrix, n rows, m predictors)
84              AppointmentData prediction_vector = new AppointmentData
85              {
86                  utilization_w_current_session = current_util,
87                  ratio_afspraak_zonder_patient =
                        ratio_no_patient_aptmnt,
88                  medewerker_id = medewerker_id,
89                  vestiging_id = vestiging_id,
90                  util_added = 0D,
91              };
92
93              // Predict added utilizations
94              // Create PredictionEngines
95              PredictionEngine<AppointmentData, UtilizationPrediction>
                    predictionEngine = mlContext.Model.
                    CreatePredictionEngine<AppointmentData,
                    UtilizationPrediction>(trainedModel);
96              UtilizationPrediction prediction = predictionEngine.
                    Predict(prediction_vector);
97              float predicted = prediction.utilizationPrediction +
                    current_util;
98
99              if (predicted < 0)
100             {
101                 predicted = 0;
102             } else if (predicted > 1)
```

```
103                 {
104                     predicted = 1;
105                 };
106 
107                 return predicted;
108             }
109         }
110 }
```

The steps taken in the code are

1. Loading the trained model.
2. Retrieving the data from the database.
3. Predicting added utilizations using the trained model and the predictors.
4. Transforming the added utilization to utilization.

The final step has the following requirement:

$$0 \leq \textit{Predicted utilization} \leq 1$$

The required transformation are, therefore:

$$\textit{Predicted utilization} \leftarrow \textit{Max}(0, \textit{Predicted utilization})$$

$$\textit{Predicted utilization} \leftarrow \textit{Min}(\textit{Predicted utilization}, 1)$$

In which the former transformation ensures that the predicted utilization is larger than or equal to 0 and the latter transformation ensures that the predicted utilization is smaller than or equal to 1. In the code, this is achieved with the clamp function.