Automating protocol selection using a data-driven approach an Azurion 7 case study

Schoeman, Auke J.

Award date:
2020

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
Architecture of Information Systems Research Group

Automating protocol selection using a data-driven approach: an Azurion 7 case study

Master Thesis

Auke Schoeman

Supervisors:
dr. N. (Natalia) Sidorova (TU/e)
ir. A. (Angelique) Brosens-Kessels, PDEng (Philips Healthcare)
prof. dr. A. (Alexander) Serebrenik (TU/e)

Eindhoven, December 2020

Abstract

In the era of big data, large and complex datasets become more common. With the growing amount of data, increasingly complex patterns can be discovered. The aim of this research project is, given an event log, to asses the feasibility of improving the user experience through prediction of user actions in systems with complex human-machine interaction. In a setting where the user is performing a complicated action where complex interactions between the system and user are required, automating a subset of the actions is beneficial. In the different interactions there may be an interaction which is repeated and relatively simple. Being able to automate such an interaction will allow the user to spend the attention required for the interaction to different activities.

In this thesis a case study is performed at *Philips* where the Azurion 7 platform is the system of interest. The interaction of switching protocol is assessed. Automating X-ray protocol selection on the Azurion 7 platform allows the user to focus on the core work of treating the patient. The Azurion 7 is an interventional X-ray (iXr) solution used for minimally invasive interventions. This research project follows the CRISP-DM methodology. The first phase is business and data understanding, in this phase descriptive analytics is performed to identify the available data sources. Several data challenges are identified in the main data source. Furthermore, this phase limits the study to four protocols. During the second phase, data preparation, the data challenges are discussed and solved. One of the challenges consists of finding a suitable case identifier. This identifier is obtained through testing several scenario's on the Azurion platform in a test setting. Furthermore, the data is prepared for modeling through several preprocessing steps. Modeling is the third phase, multiple machine learning algorithms are explored and optimized to predict the protocol during this phase. The algorithms are step-wise improved and at each step the models are compared to understand in which step the improvements are found. The predictions are evaluated using the F1 score and show that it is not feasible to automate the selection of X-ray protocols based on the available data. The results are not adequate for implementation because of the limited performance of the models. One of the factors which limits the performance is the lack of context information. During the different phases of the CRISP-DM cycle, it is observed that demographic information about the patient is hardly available. Another limiting factor is observed during the analysis of exam trajectories and the results of the model, these showed large differences between exams.

The results show that in the case of automating the X-ray protocol of the Azurion 7 platform, it is not feasible to predict the user action based on the available data. However, in settings where more extensive event logs or additional contextual information is available, the method can help to identify and predict user actions which offer potential for automation.

Management summary

This research project is conducted at *Philips Healthcare BV* at the Innovations department of the image guided therapy business unit. The focus of the project is on the Azurion 7 platform, which is an interventional X-ray (iXr) system used to perform minimally invasive interventions. During these exams, the clinician uses the control module to operate the Azurion platform. For each acquisition performed during the exam, an X-ray protocol is selected. Each area of the patient's body has a separate X-ray protocol which is selected and changed by the clinician throughout the exam.

Problem statement

The Azurion 7 platform can be seen as a platform which requires skilled and experienced users. The users of the platform perform complicated tasks where interactions between the user and the platform are complex. Within the complex interactions there are user actions which are repeated in many exams and have a certain degree of predictability. Automating such user actions enables the user to spend more attention to the core task. Therefore, the following research question has been formulated:

> *"Given a system event log, is it feasible to improve user experience through prediction of user actions in systems with complex human-machine interaction?"*

The *Philips* Azurion 7 platform is used as case study to answer the research question. Currently, the X-ray protocol is manually selected by the clinician and is changed when a different body part is examined. This is an example of a user action which offers potential for automation. The user action of switching protocol is the action of interest in the case study.

Multiple sub-questions are addressed to answer the main research question. During the research project the CRoss Industry Standard Process for Data Mining (CRISP-DM) method is applied. This method consists of six phases, some have been merged for this project. First, the business and data understanding phases have been merged and herein descriptive analysis is performed. The third phase, data preparation, transforms the available data to a dataset which is suitable for machine learning algorithms. The next two phases are modeling and evaluation. These are merged, here predictive analysis is performed and the results are analyzed. The last phases is deployment, in this phase is explored how the model can support he selection of an X-ray protocol.

Descriptive analysis and data preparation

During descriptive analysis, the business and data understanding phases are discussed. The data source is the event log of all Azurion 7 platforms placed in hospitals. The logs save a wide variety of information about the Azurion 7 platform. The research project focuses on the event logs which contain information about the geometry settings of the system or information about the acquisitions during an exam. Further filtering is performed on the X-ray protocols, since this is an exploratory study the number of protocols is limited. The following four protocols are taken in account: 'Upper Legs', 'Lower Legs', 'Abdomen' and 'Iliac/Pelvis'. These protocols are chosen due to the relatively big differences between protocol settings and expected geometry settings.

During descriptive analysis, three challenges in the data are found, these are addressed in the data preparation. The first challenge is the fine-grained logging. There are numerous distinct activities with different levels of granularity. This challenge is overcome by filtering low-level activities and activities which are not related to the geometry. The second challenge revolves around case distinction. The event logs start a case, also referred to as exam, at the first acquisition after a patient is loaded in the system. However, movements recorded before this acquisition are important for this research project. Through a specific event related to starting the procedure, movements up to 60 minutes before the first acquisition in an exam are incorporated. The last challenge are the labels which may include noise. The labels are the selected X-ray protocol for each acquisition. During exams it occurs that the clinician is using the incorrect X-ray protocol which introduces label noise in the dataset. Through a set of assumptions, the exams are filtered to limit the label noise.

The remaining event logs have to be transformed to a dataset which allows a machine learning algorithm to learn. This step is taken during the data preparation phase. The final dataset has an entry for every acquisition during exams of interest. For each acquisition a set of features is collected. The main features are the previous movements of the system, 30 movements preceding the acquisition are added as well as the first movement of the exam.

Predictive analysis

The X-ray protocols of interested have been grouped in two classes, both leg protocols are merged and the protocols 'Iliac/Pelvis' and 'Abdomen' are merged as well, making the problem binary. The following machine learning techniques are used and compared: Random Forest classification, XGBoost, Logisitic regression and a neural network. Every technique is taken through a number of steps to improve the performance. The steps consist of data scaling, feature selection, tuning of hyperparameters and cost-sensitive learning. The models are evaluated on several criteria, most important is the F1 score due to the imbalanced dataset.

Conclusions

The descriptive analysis showed several challenges of which label noise is the hardest to tackle. The predictive analysis steps showed that the base models performed acceptable, but there is significant room for improvements. Going through the steps minimal improvements are observed. Cost-sensitive learning provided the most improvements. The best model is a XGBoost model with a F1 score of 0.727 which is not sufficient to be deployed in a real-life scenario. Analysis of the results show that there are large differences between exams and most incorrect predictions are related to the switch between protocols. Therefore, it is not

feasible to automate the selection of X-ray protocols solely based on the event logs.

Limitations and future work

The main limitation of this project is the available data. The event logs provide extensive amounts of information about the machine but hardly any context information. There is hardly any information about the patient or goal of the exam, which is crucial information. Future work can revolve around incorporating additional data sources. These data sources should mainly focus around the patient and can for example be demographics from medical records or using image recognition to supplement the geometry information. Lastly, in this project the label noise is limited through assumptions. Obtaining a dataset with labels which have less noise should be a priority for future work.

Preface

This master thesis presents my graduation research project conducted at Philips Healthcare in Best. It marks the end of my student life which I gratefully enjoyed. Hereby, I would like to express my gratitude towards everybody who helped me during my thesis and student life in general.

First, I would like to thank Natalia Sidorova, my first supervisor, for all the constructive feedback and guidance during the project. Second, I am grateful for the opportunity to execute my master thesis project at Philips Healthcare. I would like to thank my company supervisor Angelique Brosens-Kessels for her continued support and feedback during the project. Especially during the pandemic which changed to way of working for everyone.

Furthermore, I would like to thank my friends and family who have always supported me during my master thesis and my academic life. The pandemic proved to be a challenging period that changed the way or working and studying. A special thanks to my parents and girlfriend for offering a listening ear to my thesis updates, struggles and thoughts with great patience.

Auke Schoeman, December 2020

Contents

List of Figures

Chapter 1

Introduction

In this chapter the topic of this master project, part of the *Data Science in Engineering* Masters degree at Eindhoven University of Technology, is presented. It starts with introducing the context and the broad aims of the project in section 1.1. In section 1.3 the research goals of this project are discussed. Third, the method used is explained and in the last section the outline of the project is discussed.

1.1 Research context

In the era of big data, datasets are becoming larger and more complex. Big data is expanding in all engineering and science domains (Wu et al., 2013). This is visible in the healthcare domain as well. Each human displays patters of actions every day without thinking about it. All the patterns are opportunities for automation. With the growing amount of data, increasingly complex patterns can be discovered.

This research is performed at the Innovations department of Philips Image Guided Therapy (IGT) systems which is a business unit of Philips Healthcare. This business unit is responsible for developing integrated IGT solutions that advance minimally invasive procedures. With the help of these solutions clinicians are able to guide specific tools through the patient via small incisions instead of open surgery. The specific interventional X-ray (iXr) platform that is of interest of this study is the Philips Azurion 7 platform. This does not only include the physical machine, but also the supporting software applications and flex work-spot in the control room. The aim of the project can in broad terms be described as *understanding and exploring opportunities to improve user experience through data driven methods.* User experience is meant in the broadest sense of the words. Both the experience of the clinician and lab staff interacting with the platform, as well as the patient's experience.

In order to be able to design and manufacture machines which support the clinicians during the treatment of a patient, it is necessary to be aware of the workflow. The studying of user behavior is an important step in product design and development. In a process in which complex user interactions are required, studying user behavior is more complex. *Philips* works with actual users to gain insights in the way of working with the Azurion 7 platform. By observing procedures and collaborating with end-users, the platform is improved. However, the observations and collaborations are with a small subset of the platforms and users.

A method to include more observations is to look at the data which is generated by the Azurion 7 platforms deployed. Historic data about the individual systems are available for many hospitals all around the world. Using system data could provide insights of ways of working of which a usability engineer did not think of. The data which *Philips* stores for each platform has different levels of granularity and is not easy to interpret. An Azurion 7 platform can be seen as a tool which the clinician uses in the process of treating a patient. This tool provides data which can be seen as the state of all resources (e.g. system components, positional data and status of the procedure). The state of the system is logged at a certain point in time rather than a proper sequence of activities. A change in state is logged, but the reason for the state change is not. In order to use machine learning algorithms, this data has to be transformed.

Previous work based on the event log data at *Philips* include research on patterns and anti-patterns (Pietraru, 2018), generation of field-based usability testing scenarios (Hoornaar, 2017) and discovering use patterns through the use of the contextual information by using the notion of high-level activities (Uku, 2019). The previous work focuses on improving the design process by identifying user behaviour that has potential for automation through the use of process mining. Nokelainen et al. (2018) show automation on different levels of granularity. These levels are: augmenting and enhancing, assisting and guiding, and automating. Where the last two levels are most interesting. Previous work at *Philips* focused on finding automation on the level of assisting and guiding. Improvements are introduced which assist the clinician, *Philips* would like to improve the Azurion 7 platform to the next level, automating. Automating core features of the Azurion platform allows to free human capacity, mainly of the clinician. If we draw an analogy to cars, lane assist and cruise control are features which assist the user in driving from A to B. Automating core features like recognizing situations and controlling the car is on the level of automating the car, self-driving capabilities are on this level of granularity. In order to achieve this next level, core features of the platform should be automated. If we take this in context of *Philips IGT*, the treatment of a patient is the core process. The Azurion 7 is an platform which supports the clinician in his work during the treatment process. Currently, there are many supporting and assisting features of the Azurion platform which help the clinician but a core feature should be automated to achieve or start an effort towards automation.

Achieving automation of user actions which are complex is a hard task but can be very rewarding. In a setting where the user is performing a complicated task which requires multiple interactions with the system, the user is concentrated and performing the interactions requires a high amount of human capacity. A subset of the interactions with the system may be suitable for automation, allowing some of the capacity of the user to focus on his core work. Janssen et al. mention this as an advantage in their review on the history of human-automation interaction research; "as automation continues to improve, automated tasks might require less human attention and intervention. This allows humans to focus on other activities" (2019). In the context of the Azurion 7 platform, the user of the system is a clinician and the process is the treatment of a patient. Automating small tasks will allow the clinician to focus more on the core work of treating the patient instead of interacting with the system. Even in a complex environment as the Azurion 7 platform, there are user actions which are not very complicated but require user attention. Small tasks that require attention distract the user from the main process, being able to automate such tasks will benefit the process. All the Azurion 7 platforms deployed provide large amounts of data, by

using machine learning techniques opportunities to predict the next user actions are explored.

1.2 Relevant work

The use of artificial intelligence (AI) is beginning to have an impact in medicine at multiple levels according to Topol (2019). One of these levels is the clinician, by accurate image recognition and interpretation they are supported. Furthermore, Topol (2019) makes an analogy with the car industry, this industry has identified 5 levels of autonomy where level 0 is no automation and level 5 is full automation. Level 3 is defined as conditional automation in which automated systems operate and monitor the environment but rely on a human for backup. Topol (2019) expects that level 3 is the highest level of automation which can be achieved in medicine. Currently efforts are made to support the clinician, for example in clinician decision support systems with cognitive computing (Sardar et al., 2019). These support systems mainly focus on lesion detection and treatment strategies and less on supporting the clinician during the exam through automation of actions (Sardar et al., 2019). Lundberg et al. (2018) show that it is possible to support the clinician with predictions to prevent low blood oxygen. Extensive data sources cover sequential minute-by-minute information and are extended with patient medical records available in the hospital. They applied several machine learning algorithms on the data and compared the results of the models.

Limitations on AI in medicine revolve around several topics. Machine learning algorithms train on data collected in the past and can not reason about unseen conditions as a human can (Stead, 2018). The performance of a model is highly dependant on the available training data which generally is difficult to obtain in medicine due to the privacy regulations. Ozmen et al. (2020) mention: "the predictions AI can make is susceptible to the systematic biases in clinical data collection". Being able to validate the data and model is important, especially in the healthcare domain. Furthermore, several algorithms are so called 'black box' models, exact reasoning behind the model is unknown. The potential of harming the patient due to a flawed algorithm is highly unwanted (Topol, 2019).

This thesis focuses on predicting the next user action to support the user of a system. Predicting user action or intention is a widely studied topic. An interesting application is the work of Sarker et al. (2019) who use smartphone event logs to predict smartphone usage and patterns. Several machine learning algorithms are compared to see the effectiveness of the classifiers. Other applications which predict user actions or user intentions use for example: markov models (Baldominos Gómez et al., 2016), naive bayes and a support vector machine (Shen et al., 2006), or deep learning (Tan et al., 2018).

The aim of this thesis is to bring methods to predict the next user action to the healthcare domain. These methods includes the comparison of machine learning algorithms, like Lundberg et al. (2018) or Sarker et al. (2019) apply. In the healthcare context of interest to this thesis, the human-machine interactions are complex and achieving a higher level of automation would support the clinician.

1.3 Research Goals

As mentioned before, the broad aim of this thesis is to *understand and explore opportunities to improve user experience through data driven methods.* To narrow down this broad aim, a research question is formulated as:

> *"Given a system event log, is it feasible to improve user experience through prediction of user actions in systems with complex human-machine interaction?"*

To answer the research question, several sub-goals are defined. For each sub-goal a short motivation about why this sub-goal is important is included.

Research goal 1: *Identify the available data sources, the limitations and create a relevant dataset in context of the Azurion 7 platform*

In order to make accurate predictions, a dataset is needed. This research goal focuses on finding such a dataset at *Philips IGT*. First, the available data sources are identified and the limitations of each source is discussed. Data exploration is used to find a relevant subset of the available data. Furthermore, challenges within the data are identified and solved.

Research goal 2: *Identify a model which is able to accurately predict the next user action while using the Azurion 7 platform*

When a suitable subset of the data is found, it is important to find a predictive model. In order to answer this question several models are explored, implemented and optimized. These models are compared to each other and the base models before optimization.

Research goal 3: *Explore how the model can be deployed and in what way the prediction can be presented to the user*

A prediction model is not the last step, the prediction has to be brought to the users of the platform. In what way can the information be presented such that the user can act upon the information? Multiple suggestions are made to show in what way the information can be presented to the user.

1.4 Methodology

In this section the overall methodology of this project is discussed. A structured research approach helps to identify, analyze and solve the problem. The presented research goals and the nature of the project are suitable for a data analysis approach. The research questions are answered by using a standardized process introduced by Wirth and Hipp (2000); CRoss Industry Standard Process for Data Mining (CRISP-DM). In CRISP-DM six iterative phases are described, see fig. 1.1.

1.4.1 CRISP-DM Cycle

CRISP-DM is a standard process model that describes the common approaches used by data mining experts and is a widely used analytic model. The CRISP-DM methodology divides a data mining cycle into six distinct phases: *business understanding, data understanding, data preparation, modelling, evaluation, and deployment.*

Each of the phases has three levels of abstraction. At the top level the *generic tasks* are placed, these tasks should cover all possible data mining situations. The second level of abstraction are the *specialized tasks*, here it is described how the generic tasks should be performed with the goals and business domain in mind. The lowest level is the *process instance* level, this represents what actually happened in terms of actions and decisions. In each phase there are several standardized tasks, these are further introduced in the corresponding section. Wirth and Hipp (2000) encourage selecting additional tasks or leaving out tasks based on the added value or irrelevance. The full process is a iterative sequence of sub-processes. During the data understanding there may be problems which were not accounted for during the business understanding. Subsequently, during the modeling phase new modeling techniques can be found which require a different data structure. In both of these cases a phase is revisited to make improvements or change certain aspects.

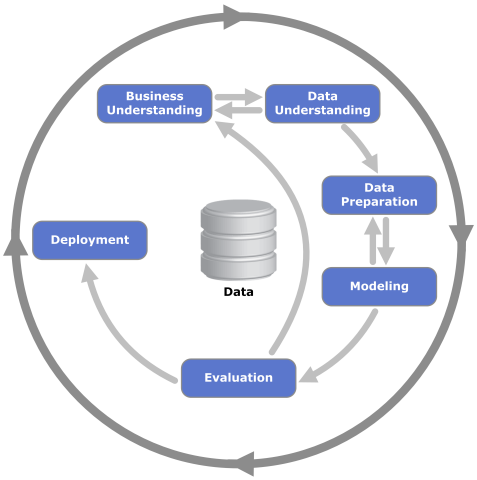


Figure 1.1: CRISP-DM Cycle (Jensen, 2012)

Business understanding

The initial phase focuses on acquiring knowledge about the project objectives and requirements from a business perspective. Thereafter these objectives and requirements should be translated to a data problem.

First, in order to identify and understand the objectives and requirements, one should be familiar with the business operations and product. In this project the focus is on the geometry of the Azurion 7 platform which is a highly complex system. It is required to acquire a certain level of knowledge of the operations around the system to understand the data.

Data understanding

The goal of this phase is to gain insights in the data and detect interesting subsets. Ultimately a good understanding of the available data and its quality is required. This phase has a close link with the business understanding. Formulating a data problem from a business perspective requires knowledge about the available data.

The first step in this phase is the initial data collection and exploration. *Philips Healthcare* uses *Vertica*, a data warehouse solution to store the data. *Vertica* is explored to gain an

understanding of the availability, magnitude and quality of the data. One of the generic tasks is the verification of the data quality.

Data Preparation

The data preparation phase includes all steps which are taken to construct the final dataset. The selected dataset has to be suitable and relevant for the data mining goals. In this project we see that the available data is very extensive and a lot of filtering has to be performed to find this suitable subset. These filtering steps are discussed and justified. Furthermore, in order to gain useful insights, the selected dataset has to represent the correct behaviour in the lab. First, several assumptions are made to what deemed 'correct behaviour'. Due to the nature of the logging and lack of context information this proves to be challenging. With these assumptions we look for a subset of the data that can be used for modeling.

In addition to the selection of the dataset, the data has to be transformed. Due to the structure of the data in *Vertica* several processing steps have to be taken to transform the dataset to a suitable format.

Modeling

Multiple modeling techniques are selected and tested in this phase. In general there are several modeling techniques for the same type of problem. For every of the techniques selected, any assumptions should be defined. Thereafter a procedure to test the models quality and validity should be generated. In this test design the intended plan for the training, testing and evaluation of the models should be described.

After the test design, the model is build. For most models there are several parameters which can be tuned. By using hyper parameter tuning the model is optimized. Furthermore, the model is described and results are discussed.

Evaluation

In the evaluation step the results are discussed with the business objectives in mind. This includes the performance of the models and other results or insights gained. If time and budget permits the model could be tested on a real scenario. For this project, testing any models on a live machine is not feasible due to the nature of the system and limited time.

Deployment

In this phase the evaluation results are used to make a strategy for deployment. Often a deployment plan is written to outline the strategy of deployment. Additionally, monitoring and maintenance is planned. At last a final report is presented. For this project the deployment is out of scope except for exploring some user interface ideas.

1.5 Thesis outline

Chapter 2 discusses the first two phases of the CRISP-DM cycle, business understanding and data understanding. The Azurion platform is discussed and the available data sources are identified and described. Furthermore, the data challenges which are found during the data

understanding phase are mentioned. Chapter 3 discusses every preprocessing step taken to build the dataset. Furthermore, the data challenges described in Chapter 2 are solved. In Chapter 4 the modeling and evaluation phases of the CRISP-DM cycle are discussed. First, the machine learning models are discussed during model design. Second, the models are implemented and taken through a set of steps to optimize the models. After implementation of the models, the results are compared and analyzed. Chapter 5 discusses the validation of the model and the discussion. The disucssion includes the deployment phase of the CRISP-DM model and the limitations and future work. In Chapter 6 the research goals are discussed and the main research question is answered.

Chapter 2

Business and Data Understanding

In this chapter an overview of the platform and available data sources is provided thereafter, the steps for data understanding are discussed. Data understanding is the second step of the CRISP-DM model which contains data gathering, data description and data exploration. During the data understanding step, several data challenges are identified. These challenges are described in section 2.4.

2.1 Azurion platform

The *Azurion* is an interventional X-ray platform (iXR) which is designed to perform a wide range of routine and complex procedures on patients while offering the best support for clinicians. Numerous vascular, cardio and neuro diseases can be treated with the iXR platform, examples of these are coronary artery disease, heart valve disease or vascular disease. The Azurion platform has many features to support the clinicians while performing a procedure (e.g. StentBoost, HeartNavigator). In the following section the most important features of the Azurion platform are discussed.

1. *C-arm:* is used to guide the detector and X-ray tube into place. Different patients and procedures require a different position of the C-arm. Due to the agility and maneuverability of the C-arm, images from all angles can be produced. There are two different systems, the standard monoplane model and a a biplane model for specific use cases.
2. *X-ray Tube:* is what generates the X-ray beam which is 'collected' by the detector.
3. *Detector:* 'catches' the x-ray beam which is generated by the x-ray tube. Using the data which is collected by the detector, an X-ray image or sequence of images is produced and shown on the screens.
4. *Table:* is where the patient is placed. The table is able to freefloat to move to the right position. Depending on the model, this movement can be manual or motorized. Furthermore, the table can move in height.
5. *Control module:* provides a combination of controls which are used to adjust the position of the stand and table, and can also be used to control image functions during acquisition.

6. *Touch Screen Module (TSM):* is used to control the settings of the acquisition, to store and recall projections and to control different applications of the Azurion platform.

7. *Control room:* In here additional parallel work spots are set up, technicians can for example support the doctor in his current exam or set up the platform for the next exam. Additional work spots can be set up if requested.

8. *FlexVision or monitors:* is an optional big screen on which information is displayed to support the clinician in his work. On a standard model there are several monitors instead of one FlexVision screen.

9. *Foot pedal:* has three or four pedals depending on the model, the pedals are used to control the fluroscopy and exposure. The right pedal is customizable and can be assigned to a different function.

One of the features of the Azurion platform is the selection of a procedure card. This is a digital card in which pre-defined settings are saved. These settings are extensive and range from the X-ray protocols and orientation of the patient to presets of the FlexVision and FlexSpot. There is a set of standard procedure cards which are automatically selected. Procedure cards can be customized or new cards can be made by the clinician. If a clinician prefers a certain layout of the FlexVision or performs a certain intervention often, the procedure card can be changed to fit the needs. Procedure cards help to improve the workflow of the exam by standardizing some parts of it.

During an exam, the clinician has many interactions with the system. These interaction range from changing the view on the Flexvision to positioning the C-arm using either the TSM or the control module. The clinician generally moves the detector to several areas on the body during one exam. For each body part there is a X-ray protocol, every area on the body has one or more protocol(s). The selection of protocol is one of the tasks which offers potential for automation. It is a task which occurs in most exams and is performed by the clinician but often is a logical sequence of events. If there is an exam which requires the clinician to move the detector from the abdomen to the heart the difference of position indicates that a switch may be next action of the user. In the case study of the Azurion 7, the switch between protocols will be the action of interest. Automating the switch between protocols will allow the clinician to focus their attention to the patient instead of the interaction with the Azurion 7 platform.

2.2 Data availability

Within *Philips* there is one database which provides an extensive amount of information about the Azurion platforms deployed. This information ranges from the individual components of each system to the activities performed on each system. This research focuses on the Azurion 7 platforms, for these systems event logs are retrieved on a daily basis and stored in this database. These event logs provide information about all aspects of the system, ranging from low-level information about what button is pressed, to all information about the amount of radiation the patient is exposed to. Logging is not performed with data mining or process analysis in mind but for engineers to provide maintenance. Separate parties log activities which are of interest and useful to themselves.

For this project, the main interest in these event logs is the information about the patient and the geometry information of the system. The geometry of the system is expected to be a good predictor of the switch. All aspects of the geometry of the system are recorded, as well as which buttons are used to move the machine in this position. Using existing applications within *Philips*, it is possible to derive the focal point of interest from the geometry settings. In section 2.3 the event logs will be extensively discussed.

Labeling of the data is automated and user-generated. The clinician selects a protocol before every acquisition, this protocol is used as label. Obtaining labels which are manually labeled or confirmed to be correct is not possible because manually labeling the data points is a labour intensive and expensive task, which is not in the scope of this research. User-generated labels may not always be correct and can introduce label noise which means that the ground truth is unknown to some extend.

2.3 Data gathering and description

The data consist of event logs from the Azurion platform, these events can be generated by the system or by user interactions with the system. Every day, the events are downloaded from each individual machine, processed and loaded into the *Vertica SQL Database*, a column based storage solution. For the iXR platforms this results in one large table which is split in several tables by ETL's. A Python application is used to access Vertica. In this application SQL queries are performed to collect the data of interest.

The events table consists of all events from all machines and is extremely large. In order to query properly, these queries have to be well defined and precise. In table 2.1, a sample of a raw event log is shown. It has already been filtered to show several relevant columns. Multiple difficulties in the event log were immediately identified. For example; there are several events at the same timestamp or very close to each other, in these cases the exact order of events is uncertain. This will be discussed in section 2.4 among other challenges.

Eq. Number	**Timestamp**	**Description**	**EventID**
77559984	07:49:02.258	RequestMovement executed	20SSPOS6000025
77559984	07:49:02.531	Viewpad: UiActivity detected	20SSIEC0013000
77559984	07:49:04.023	Command: PivotBeamFrontal.Move	20SSPOS0009921
77559984	07:49:04.023	Command: UIActivity	20SSIGC0009921
77559984	07:49:04.025	RequestMovement executed	20SSPOS6000025
77559984	07:49:04.298	Viewpad: UiActivity detected	20SSIEC0013000
77559984	07:49:05.480	Command stop	20SSPOS6000023
77559984	07:50:15.560	Command: PivotBeamFrontal.Move	20SSPOS0009921
77559984	07:50:15.561	Command: UIActivity	20SSIGC0009921
77559984	07:50:15.563	Command start	20SSPOS6000022

Table 2.1: Sample event log from Vertica

Through the use of several ETL's, the event table is split into sub-tables. These sub-tables all contain information which can be found in the events table. The exams acquisitions table defines every image acquisition and splits information which is stored in the *additional info*

into separate columns. In this way the duration, dose and other attributes are shown as separate columns of the acquisition. This table also defines an ExamID, where the start and end of the exam are defined as the first and last acquisition on the corresponding patient, respectively. Although the ExamID is useful, there are events of importance which occur before the first acquisition thus, only using the ExamID is not sufficient.

Another table of interest is the movement table, in here all the information of the movements is stored. For every movement, the end position of all the geometry is saved. The ETL of the Azurion 7 platform is not yet updated, this means the movement data has to be gathered from the event log. In table 2.1 a *command:stop* movement is shown, included in the logged data is a column named *AdditionalInfo*. In *AdditionalInfo* all movement data is logged, through python code the movement data is extracted from the event logs. The details are discussed in section 3.2.

The geometry consists of a set of features and are listed in table 2.2, from these features all the geometry settings of the system can be calculated. The calculations for the frontal stand of the C-arm, lateral stand and the table are performed separately. Together they depict the geometry settings of the system. The position of the system is combined with several other features to make a prediction of the X-ray protocol. The last known position can provide useful information, but the path which the system has taken to get to this position is insightful as well, especially for acquisitions in a later stage of the exam. To provide a part of this path, the last 30 movements before the acquisition are gathered, together with the time between the movement and the corresponding acquisition. This will be elaborated upon in chapter 3.

Frontal stand	**Lateral stand**	**Table**
BeamLongitudinal	BeamLongitudinal	Height
BeamTransversal	C-arm (Roll)	Lateral
DetectorShift	DetectorShift	Longitudinal
Propeller	Propeller	Tilt
RotateDetector	StandArea	Cradle
StandArea		Pivot
Swing		Swivel
Z-rotation		BaseLongitudinal

Table 2.2: Geometry features

In addition to the numbers, the type of movement is logged, for example *'Change Patient Support Height'* or *'Shift Detector Frontal'*. It is interesting to see what kind of movement occurs right before the acquisition, the top 12 movements are shown in figure 2.1. The top four movements all involve the frontal stand of the system, these are parts related to the C-arm and used to move the detector in place. These kinds of movements are expected to be close before the acquisition.*'MoveBeamLongitudinalFrontal* is used to move the c-arm in a longitudinal motion along the table. *'ShiftDetectorFrontal'* is a movement to extend the detector. In order to get a good x-ray image, the detector is moved close to the body of the patient.

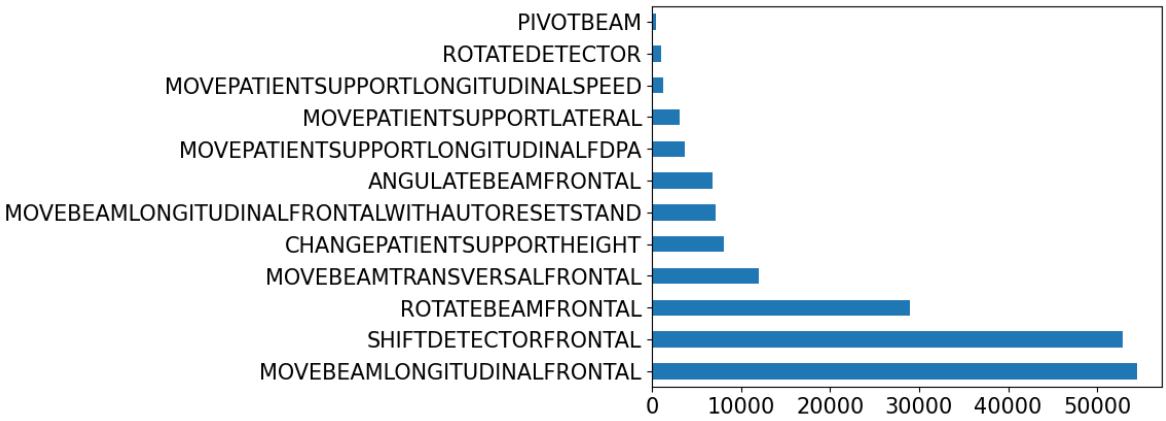


Figure 2.1: Last movement before acquisition

By using the event logs it is possible to create a dataset with a variety of features. Via the equipment number, the system configuration can be extracted from *Vertica*. There is a variety of configurations available, for this dataset several have been chosen which might help predict the X-ray protocol. In table 2.3 all features used in the dataset are listed. The features regarding the patient and table will remain the same over the course of an exam. A high number of movement related features are selected, the last three features range over n where $n \in \{0 \dots 30\}$. This means that the 30 movements preceding an acquisition are used, for every movement all geometry features of table 2.2 are included.

Feature name
Patient weight
Patient age
Table type
Table base
Table top
Position first acquisition
Time to first acquisition
Procedure first acquisition
Position previous acquisition
Time to previous acquisition
Procedure previous acquisition
Patient position movement n
Time to movement n
Geo position movement n

Table 2.3: Features in dataset

Patient age and weight are the same for every acquisition within an exam. In total we have 7415 unique exams in the dataset after selection of section 3.2. In figure 2.2 the distribution of the age of the patients is shown. In 1% of the cases the patient age is unknown. The majority (61%) of the patients is between 60 and 80 years old. A very low percentage of the

patients is younger than 50 or older than 90. The weight of the patient not logged in the majority of the exams, 78% of the patients has an 'unknown' weight.

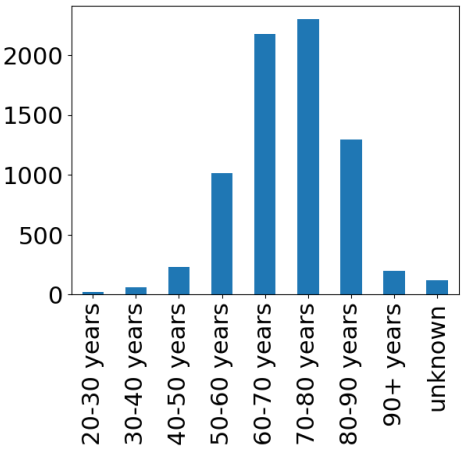


Figure 2.2: Distribution age of patients

The Azurion 7 platform is used for minimally invasive procedures which "are widely regarded as the answer to treating more patients, more quickly and at lower cost" (Tabaksblat, 2019). The Azurion 7 platform has many configurations and is able to perform a wide variety of procedures. This is reflected in the number of X-ray protocols to choose from, the choice of protocol depends on the type of procedure. In this thesis will we focus on a subset of protocols, the reasoning behind this is explained in section 3.2. Four X-ray protocols are taken into account, these are: 'Upper Legs', 'Lower legs', 'Iliac/Pelvis' and 'Abdomen'. Every acquisition has a corresponding protocol, for each protocol the number of acquisitions is shown in figure 2.3. The X-ray protocol which is most used in our dataset is 'Upper Legs' with 42% of the acquisitions. 'Iliac/Pelvis' or 'Abdomen' is selected in 35% of the acquisitions, the remaining 65% is one of the leg protocols.

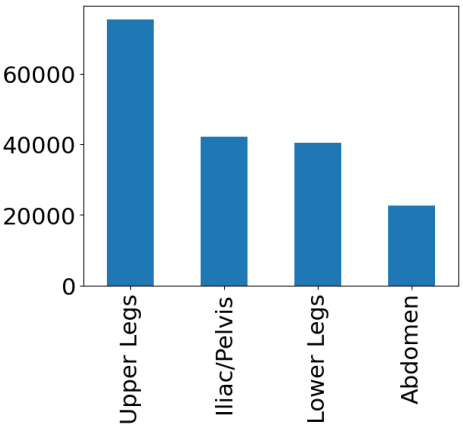


Figure 2.3: X-ray protocol of the acquisitions

2.4 Data challenges

The logging of the Azurion platform has several challenges. Solving these challenges is an important aspect of this project. The event logs are not generated with the purpose of data mining or analysis of user behaviour in mind, but in order to perform maintenance. In the following sections several challenges are addressed, the solutions to these challenges will be provided in section 3. The challenges which are identified and discussed are:

- Fine-grained logging
- Case distinction unclear
- Label noise

Fine-grained logging

Event logs generated by real-world system come from different subsystems. In our case these subsystems can be the *TSM*, *Workspot* or *Flexvision*. All of these can perform the same actions but will be logged differently. In our case study the focus will be on one type of platform however, within this system there are many differences between labs. Layouts differ and the Azurion platform can have different components. Some have multiple parallel working spots, others do not have a Flexvision screen, this adds additional complexity in the logs. Furthermore, as is shown in table 2.1, there are events which happen on the same timestamp. Due to technicalities and different subsystems, the exact order of these events is not clear.

There are 5980 unique event ID's in the event logs, every event ID has a description. There are unique ID's found in which the description differs thus, the event ID's are not a unique identifier. The actual number of unique events is even higher. These events range from low-level events to higher level events. An example of a low-level event would be that a certain *LED light on the UI module is turned on.* A higher level event would be an *Exam started* event. These different levels of granularity add to the complexity of the dataset and will have to be taken into account when selecting a subset of the data.

Case distinction

In order to perform any analysis, it is important that we understand which activities belong to which cases. In our setting we would like a case to involve every activity which belongs to a particular patient. The current approach is that the start of a case is defined by the event *Lab: exam started* and ended when the button *End procedure* is clicked. In many cases there are activities which happen either before *Lab: exam started* or after clicking the button *End procedure* but are relevant to the case. Mainly the activities that occur before the event *Lab: exam started* are of interest. In this period the patient is loaded in the system, procedure cards may be changed and in many cases movements are performed before the event *Lab: exam started* to position the C-arm for the first acquisition.

Label noise

In order to build a model which is able to make correct predictions, the labeling used to train the model should be reliable. Establishing a ground truth is a challenge for the area of interest

where there is a wide variety of factors influencing the procedure and there are no data points manually labeled. The labels used are entered through the system via a procedure card or manually selected by the clinician. Currently *Philips* is not able to supervise this selection or check if the labels are correct.

2.5 Data quality

The study on data quality started in the 1990s with the fast development of information technology. Wang and Strong (1997) defined high quality data as *"data that is fit for use by data consumers"*. Furthermore, Wang and Strong (1997) identify four data quality categories with multiple dimensions per category. Since then, the amount of data has increased tremendously, often referred to as *Big Data*. This introduced new challenges regarding the data quality. Cai and Zhu (2015) propose a two-layer quality standard for assessment. In their study five dimensions of data quality are described, each of these dimensions has between one and five elements as shown in figure 2.4. For every element there are several indicators mentioned as guideline. In this section several elements which are important for this study are discussed.

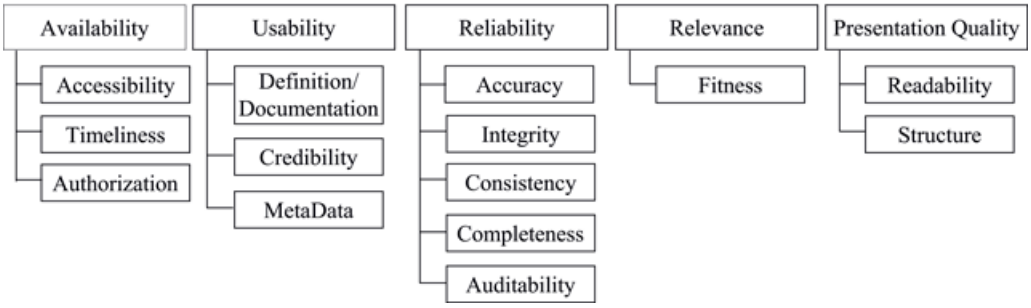


Figure 2.4: Two-layer big data quality standard, (Cai and Zhu, 2015)

Timeliness is an element in the availability dimension. In this element it is important that data arrives on time, data is regularly updated and whether the time interval between collecting and processing the data meets the requirements. The *Vertica* database is updated daily. The logs are downloaded from all individual machines, processed and stored in the data warehouse.

Connected to timeliness are the elements integrity and consistency in the reliability dimension. Due to legacy systems, not all the downloaded data is processed and stored in the data warehouse. For random platforms the process fails and that day of event logs is lost, which results in inconsistency between days.

Furthermore, we have the relevance dimension. The fitness of the data is important in this dimension, fitness has two-level requirements. The amount of data which can be accessed by the user and the degree of relevance of the data; does the data match the need of the user? In *Vertica* there is a large range of data which is accessible by the user. The user has to select the data which fits the needs. Most of the data is not logged with data or process analysis in mind. This leads to a database in which much information is stored but the user has to perform a lot of steps in order to get data that fits his needs.

Chapter 3

Data Preparation

When the data understanding phase is concluded, the next phase of the CRISP-DM cycle starts, which is the data preparation phase. The goals of this phase are to select a subset of the available data and construct a dataset which is used for modeling. The data preparation phase consists of multiple tasks. These tasks are: selecting a dataset, cleaning the dataset, feature engineering, and the integration and transformation. In this chapter we will start with describing how the data challenges of are solved. Thereafter, each of the tasks of the data preparation phase will be discussed.

3.1 Data Challenges

In this section several data challenges will be discussed, these challenges have been identified in the previous chapter.

3.1.1 Case distinction

As mentioned in the data challenges, it is important to understand which activities belong to what case. Translated to this project, it means every activity belonging to the treatment of a patient should be in a case. In this section the current approach to define a case in the data is discussed as well as a new approach.

Current approach

The current approach of defining a case is as follows; there is a separate table with exams named *Exams acquisitions*, this table is generated by an ETL which processes data from the event logs. This ETL distinguishes cases by two events. These events are: *Lab: Exam Started* and *Lab: Exam ended*. The logging of these events is triggered by different activities. *Lab: Exam Started* is logged when the first acquisition occurs after the activity *Start procedure clicked*. *Lab: Exam ended* is triggered when the button *End Procedure* is clicked. All the acquisitions and the corresponding information between *Lab: Exam Started* and *Lab: Exam ended* is stored in the *Exams acquisitions* table with a unique case ID. This table does not contain other activities which are related to the case.

The start and end time defined above can be used to gather all activities per case. After exploration of these cases it was concluded that using these start and end times did not provide all necessary information. One of the variables of interest is the orientation of the patient on the table. Within the start and end time of a case, as defined above, the patient orientation is only logged when it is changed during an exam. There is no information about the patient orientation at the first acquisition. Furthermore, we are interested in all the movements which are involved with the treatment of a patient. The current approach does not capture the movements that occur before the first acquisition, while these movements are of importance.

The identification of a case and the corresponding missing information were object of a detailed study to get more insights in the logs. The objective of the study is to get a definition of a case which includes all the events that are related to the treatment of a patient and to get additional insights in the logging of the patient position through the procedure cards. In this study several scenarios are written and have been performed on a system. Thereafter, the logs are downloaded from the machine and analyzed.

Analysis of the event logs of the scenarios showed that the patient position is not only logged when it is changed during the procedure but also after the event *Start procedure clicked*. The patient position logged after *Start procedure clicked* corresponds with the patient position of the procedure card.

Proposed approach

In order to capture all information belonging to a case, the start of a case is changed. Data exploration shows that there are large differences in the amount of time between *Start procedure clicked* and the first acquisition. A balance has to be found between incorporating as many *Start procedure clicked* events and the degree of certainty that the event is part of the case. The risk of a large gap between *Start procedure clicked* is that the patient on the table differs from the patient information on the machine which introduces noise in the dataset. Such a difference can occur for a number of reasons. A clinician can have the habit of using the machine without patient information, the machine is running and ready or why would I adjust the patient? Or a urgent case can come in between, the patient selected is a scheduled intervention while an urgent case arrives which has priority over the scheduled case. Furthermore, a reason can simply be that the clinician forgets to change the patient information or expects the technician to have changed the patient information already.

Through data exploration an initial time frame of 30 minutes before the acquisition is chosen. For all exams of interest the difference between *Start procedure clicked* and the first acquisition is saved. Results show that the cases are relatively even distributed across the 30 minutes except for the first minute and are illustrated in figure 3.1. In the first minute there is a high frequency which indicates that *Start procedure* is clicked and within seconds to a minute the first acquisition takes place.

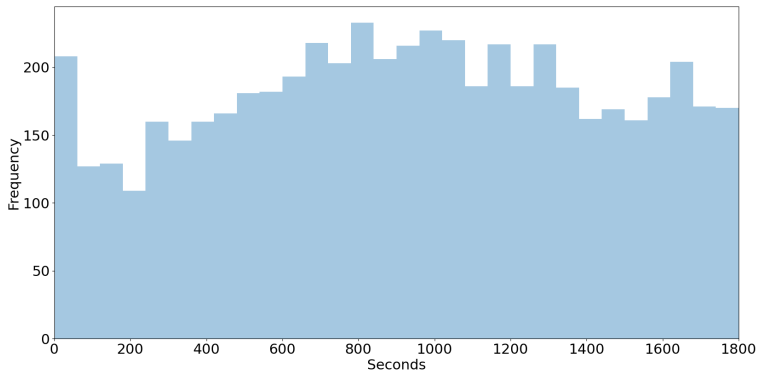


Figure 3.1: Frequency of *start procedure clicked* within 30 minutes

Because the frequency is relatively even distributed the time frame was increased to 60 minutes. Fig 3.2 shows that after 30 minutes the frequency gradually decreases. At 50 minutes before the exam, the decrease stagnates and levels. A time difference of over 60 minutes between *start procedure clicked* and the start of an exam is deemed unlikely, therefore the decision is made to include all *start procedure clicked* events up to 60 minutes before the first acquisition. When taking 60 minutes as cut-off value, a number of exams is lost. The risk of introducing label noise by having a discrepancy between the patient information and the patient on the table is deemed higher than the loss of information by removing a relatively small number of exams.

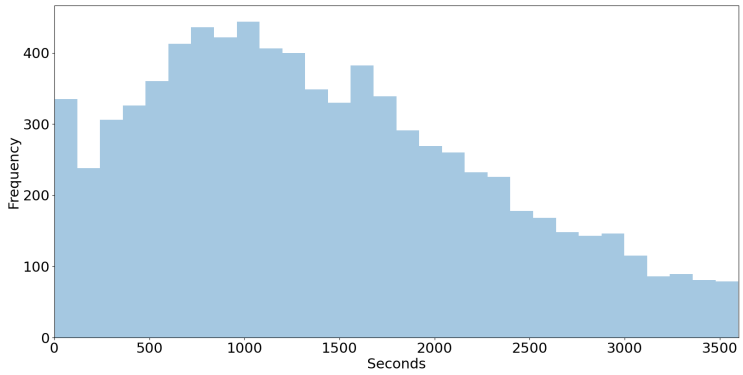


Figure 3.2: Frequency of *start procedure clicked* within 60 minutes

3.1.2 Fine-grained logging

Filtering is an important step in making the event logs more usable. Through exploration of the event logs, all activities which are of interest for our research are identified. A high amount of the activities in event logs are generated by low-level applications or applications which are not related to the geometry settings. Filtering on relevant activities cleared a lot of entries but was not sufficient.

The activities related to the geometry settings are still fine-grained and extensive. For each movement a *Command:start* and *Command:end* are generated with the starting and ending values of the geometry settings for the system. During the movement multiple activities are generated but these proved to have little information and are disregarded. *Command:end* provides the geometry features as described in 2.2. For each movement these features are extracted at the ending position. This provides the geometry settings for the system, through calculations the position of each part of the system can be calculated.

The geometry data itself provides no information about the ongoing exam and corresponding acquisitions. In order to know what X-ray protocol is used, more information is needed. This information is provided in activities related to the acquisitions during an exam and are matched to the geometry information. Through an ETL all acquisition data is stored in the table *Exam acquisitions*. This table provides all information about every acquisition during an exam, most importantly the X-ray protocol which was used for the acquisition. The X-ray protocol will be the label in the dataset, section 3.3.1 explains this in more detail.

3.1.3 Label noise

A reliable ground truth is essential to build a predictive model, the data should represent the real world. The available labels in our dataset contain X-ray protocols which are selected by the clinician or automatically selected through the procedure card. There is a factor of uncertainty whether the correct protocol is selected. There is a trade-off between traditional labeling which is labor intensive and expensive to perform, especially for big data (Frénay and Verleysen, 2013; Zhou et al., 2017) and automatic or user-generated labels. User-generated labels can introduce label noise in a variety of ways. Selecting an inappropriate protocol or switch too late or too early is possible. Minimizing the noise is important because modeling incorrect user-behavior is not desired.

The label noise is limited through a set of assumptions. However, these assumptions will not remove all label noise in the data. This has to be taken in account when machine learning models are fitted. The initial assumptions are:

1. *Looking at a Clea floor mounted system where the patient lies with his feet at the head end of the table, it is likely that a vascular protocol is used.*
2. *If we are in the scenario as described above and we see a switch between X-ray protocols, it is assumed that the switch is intended and appropriate*

By limiting the exams as described in the assumptions, the focuses is on floor mounted models. The second assumption limits the dataset to only include exams in which a switch between protocols occurs. This assumption will not remove all label noise since a number of times the switch will not be intended or appropriate.

3.2 Data selection

The first step of the data preparation phase is data selection, in this section all filtering steps and decisions are discussed. Four vascular protocols are the main topic of this research and used as labels, namely: 'Upper legs', 'Lower legs', 'Abdomen' and 'Iliac/Pelvis'. Due to similarities between the protocols it is decided to make a distinction between both of the legs

protocol and 'Abdomen' or 'Iliac/Pelvis'. The similarities of the procedure cards settings and location on the body within groups are high while the similarities between the two groups are low. This type of exams can be performed on all Azurion 7 platforms however, vascular labs generally have a system with a M20 detector. Therefore, all other systems will not be taken in account. Furthermore, to limit the amount of data only exams performed in 2019 are selected.

Exams are only included in the dataset if there is a switch between an 'Upper leg' or 'Lower leg' protocol and an 'Abdomen' or 'Iliac/Pelvis' protocol. This resulted in 7415 exams of interest. For all exams, the time of acquisition, protocol and several other features are saved. The goal is to predict the protocol based on the previous movements and multiple other features. Therefore, every acquisition has positional data of the last 30 movements added as features. This was no trivial task due to the structure of the Vertica database, this is elaborated upon in section 3.3.1. In 99% of the cases the patient age is logged. The age is logged in bins where the bins before the age of 20 are of different sizes, furthermore before the age of 20 the human body is not fully grown yet. Therefore, for this study every patient with an age below 20 is removed. The remaining bins have a range of ten years each with the exception of '90+'. It is observed that within exams there are multiple acquisitions without movements in between, the last acquisition before the next movement is taken and the others are removed.

3.3 Feature engineering and dimensionality reduction

The following section discusses feature engineering and feature selection. Both are central tasks in the data preparation step of the CRISP-DM model. Feature engineering aims to improve the predictive power of models by constructing and transforming features. Feature selection has been studied for several decades and is proven to be effective. Li et al. (2017) describe the objectives of feature selection as: "building simpler and more comprehensible models, improving data mining performance, and preparing clean, understandable data". In the big data era feature selection has become more important. Datasets are increasing in size and more features are obtained. Having a large amount of features may lead to over-fitting, higher computational costs and higher storage requirements (Li et al., 2017).

3.3.1 Feature engineering

Several feature engineering steps are taken in order to get a dataset which can be used for predictive models. In this section these steps will be discussed. The aim is to predict the protocol based on the features of the data. These features include the geometry position of the Azurion 7 platform. The time of every acquisition is known, based on this time the previous movements are gathered from the data. In order to capture all information, up to 30 previous movements are taken in account. From data exploration we see that in several cases, many small movements are performed which may provide information about the label. During the feature selection stage, the number of movements may be reduced. In addition to the 30 previous movements, the geometry settings and protocol used in the previous acquisition are added. The same features are added for the first acquisition of the exam. The idea behind this feature is that once the first acquisition has taken place, the position and protocol can be used to predict the next protocol.

The geometry of the system consists of a sequence of numbers which describe the overall orientation. Through all these numbers the position of the detector and source can be calculated, as well as the position of the table. These three positions are used to calculate the approximate focus point of the acquisition. The focus point is approximated by drawing a line between the detector and the source, thereafter calculating the intersection between this line and a xy-plane. This xy-plane depends on the height of the table and accounts for patient thickness by adding 10 cm. This results in the approximate (x, y) coordinates of the focal point on the table.

The time of each movement is recorded, it is expected that movements which occur most recent are most important for the prediction. In order to include the time aspect of each movement, the time between the acquisition and each of the 30 movements are added as features.

3.4 Data transformation

In order to transform the data in a format which is suitable for machine learning, several processing steps have to be taken. This section starts with explaining the scaling and dimensionality reduction methods. Thereafter, several transformation steps are discussed.

3.4.1 Scaling

Scaling can be important to achieve good results. Features are expressed in different units and ranges. These differences in ranges are visibile in our dataset, some features range from [0,1] while other features have a range of [0, 16200]. Machine learning algorithms may give some features a greater weight (Han et al., 2011) due to distance based optimizations. This effect can be avoided by normalizing or standardizing the dataset. Furthermore, this can help to speed up the learning phase and converge faster (Grus, 2019).

Normalization

The aim of normalization is to scale and shift the features to the interval [0,1] without impacting the data distribution. The normalization method used is known as MinMax Scaling. The general formula is:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

x' is the new feature value and depends on the minimum and maximum value of that feature. By applying this formula the data will be scaled to the interval 0 to 1.

Standardization

Opposed to normalization, standardization does not limit the range. The aim of standardization is to scale the features such to unit variance, so it has a mean of zero and a standard deviation of 1. As mentioned before, features which have different scales are not equally contributing and some features may be given more 'weight' than others only based on their range. The standard score is calculated as follows:

$$x' = \frac{(x - u)}{s}$$

Where u is the mean and s is the standard deviation.

Three datasets are constructed, each of the datasets representing a scaling technique. Unscaled, normalized and standardized datasets are constructed. The unscaled dataset has values which remain unchanged. Furthermore, in the normalized and standardized dataset, the respective technique is used to scale the data. During the modeling stage of the CRISP-DM models, several machine learning models are fitted to the individual datasets. This allows to see what the impact of the scaling is on the models.

3.4.2 Dimensionality reduction

Feature selection methods are used to reduce the number of features. When the dimensions of the data are high, reducing the number of features can have several advantages, namely:

- Lower training time
- Reduced complexity
- Improved results

There are two techniques which are often used, these are feature selection and feature extraction. Feature extraction transforms data into a new set of features, which is often smaller than the original feature set. Whereas, feature selection selects relevant features. The result of feature selection is a subset of all features. Both techniques can be used in combination with each other. Feature selection can be divided in three techniques which are discussed next.

Filter

Filter techniques select features without evaluating the performance of the classifier model and measurements are based on data distribution (Rong et al., 2019). Filter methods generally are less computationally expensive than other methods and therefore a good option for high dimensional datasets (Yu and Liu, 2003). These methods can be divided in several sub-groups; distance-based, probability-based, mutual information-based, consistency based, correlation based and neighborhood-based methods (Freeman et al., 2015).

Wrapper

A wrapper is used to find the optimal subset of the features using a performance metric, this metric is calculated by the classification algorithm (Rong et al., 2019). For every subset of features, the performance metric is evaluated and features are added or removed and the model is reevaluated with the new subset of features. This is an iterative process, is computationally expensive and has a higher risk of over-fitting than a filter method (Saeys et al., 2007).

There are many wrapper methods but those can be divided into several sub-groups; forward selection, backward elimination and nested methods. In forward selection, a feature is added every iteration. In the first iteration all features are individually fitted to the data and evaluated using the performance metric. The best is chosen and then a new feature is added and all combinations are tried. In backwards elimination this process is reversed, so it is starting with all features and eliminates them one by one. Nested methods have the option to add or remove features. From this introduction on wrapper methods it is clear that due

to the iterative nature of the selection, wrapper methods are computationally expensive and therefore not the best solution for high dimensional data.

Embedded

In embedded methods, the learning progress of the classifier is incorporated in the feature selection (Rong et al., 2019). The classifier decides which features are deleted, the deleted features have a small influence on the results of the classifier. Embedded techniques communicate with the classifier. The computational complexity is lower than a wrapper method since the classifier is not reevaluated for every subset of features. The most important factors of the computational complexity and accuracy of the embedded method are the type of classifier, its settings and parameters.

Feature extraction

In feature extraction the number of dimensions is reduced by creating a reduced feature set from the original feature set. In these new features, transformations on the original data are applied and redundant data is removed. A well known feature extraction method is Principal Component Analysis (PCA). PCA is a linear dimensionality reduction method to summarize the original data distribution with a reduced number of features. A disadvantage of feature extraction compared to feature selection is that the features are transformed in such a way that the original features are often not recognizable anymore. The data is transformed which will make individual features more difficult to interpret.

A combination of techniques is used in section 4.2.3. This section applies feature selection to see how the models are affected. As mentioned before, the goal of feature selection is to reduce the training time, complexity and increase results.

3.4.3 Transformation steps

Several other minor transformation steps are taken as well. Features which are categorical will be 'one-hot encoded'. So for every category of that feature a dummy variable is added. If the category of a data point matches the category of the dummy variable, the value of this dummy is set to 1 and all other dummy variables are set to 0. By performing this step, the machine learning algorithms can distinguish between classes without there being an ordering. As mentioned in section 3.2 the protocols 'Upper legs' and 'Lower Legs' are grouped together and assigned label 0, 'Iliac/Plevis' and 'Abdomen' are assigned label 1.

Some features do not have a proper type for machine learning algorithms. Several features are of a timedelta type, these are converted to total seconds. Furthermore, some features have numbers in string format, these are changed to floats.

Missing data strategy

The missing data is limited to missing movements. For every acquisition it is tried to record the 30 preceding movements. It is not possible to record all 30 movements for every acquisition, this can have two reasons. First, there are less than 30 movements used to get the system in position before the first acquisition. Second, there are exams in where movements occur which are relevant for the exam but are not taken into account because these movements

happen before *Exam started* or *Start procedure clicked*. In figure 3.3 we see that the number of movements which are missing are increasing when looking at movements further in the past. So the recent movements are available but after 10 previous movements we see an increase in missing movements. For 33212 acquisitions we are missing the 30^th previous movement which is a significant fraction of the total dataset. In order to account for these missing values a missing data strategy called 'Last observation carried forward' (LOCF) is used. "LOCF is a common statistical approach to the analysis of longitudinal repeated measures data where some follow-up observations may be missing" (Lachin, 2016). Longitudinal repeated measures are often used in medical studies with patients, where data is missing due to the patient not being able to attend a hospital visit. In these cases the last known observation is carried forward, in our dataset we will use this technique to carry the last known movement back. So if the movement is missing, the last known movement will replace the missing movement. It is expected that the movements close to the acquisition are most important and up to 10 movements the number of missing ones is relatively small.

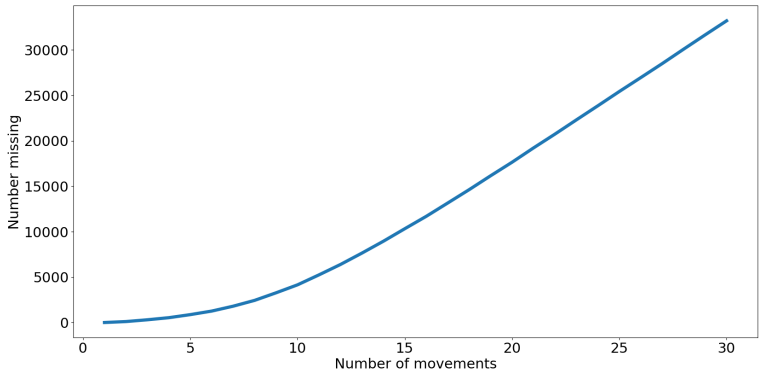


Figure 3.3: Frequency of missing movements

3.5 Iterative approach

The CRISP-DM cycle is an iterative process over the data preparation and modeling phase. During these stages the data set is processed to prepare for modeling. If during the modeling phase issues arise or it is deemed not productive enough, CRISP-DM allows to take a step back and reiterate the data preparation phase. During this project there are two main iterations, since the first iteration provided interesting results it will be discussed.

3.5.1 Floor models

During the first iteration in the data preparation phase the focus is on the Clea floor systems with a M20 detector, in 2019 a total of 25 systems of this type are deployed. These systems have a fixed C-arm which is mounted to the floor, limiting the movements of the C-arm. The goal of this iteration is to, within our assumptions, find a subset of the data which is suitable for modeling. In line with the assumptions, data for Azurion 7 systems with a M20 detector is retrieved. 429 exams of interest are found, these exams have a switch between a leg protocol and either 'iliac/pelvis' or 'abdomen' protocol. In the exams of interest a total

of 4450 acquisitions with a unique position are found where 9.05% of the acquisitions have a patient in 'legs up' position. This position has the patient with his legs at the head end of the table. Only 403 acquisitions are within the limits of our assumptions.

Heavy filtering was applied on the system type and exam requirements, resulting in a small subset of the data. Furthermore, data exploration within this subset shows that 63% of the data is generated by one system. If this subset is used to build a model the results will be biased to the particular machine and is expected to not be able to generalize to all floor systems or to ceiling models. However, the subset is able to provide insights. In figure 3.4 the table movements have been plotted, these movements give an indication of the location of the detector compared to the table. Differences between the leg protocols and the iliac/pelvis and abdomen are seen, as well as a difference between the left and right leg. Due to the limited available data and inability to generalize a step back is taken to the data preparation phase in order to find a subset of the data which includes additional exams.

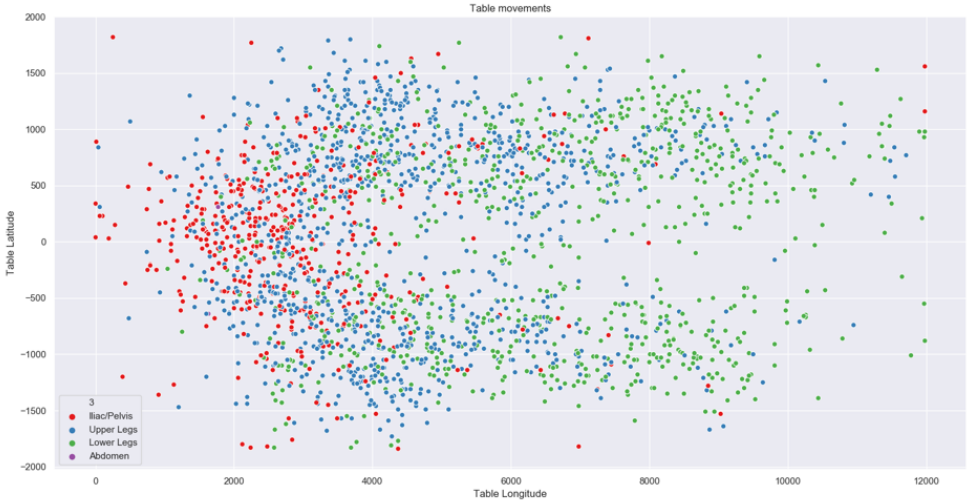


Figure 3.4: Table movements

Figure 3.4 shows an outline of the lower body of the patients. There is a clear distinction between the left and right leg in the figure. Another difference is that the lower leg protocol (green dots) are visible more towards the end of the table whereas the upper leg protocols (blue dots) are positioned towards the middle of the table. Furthermore, it is visible that the illiac/pelvis and abdomen protocols are more centered than the leg protocols. This is all expected behavior e.g., the lower legs are beneath the upper legs of a patient. However, the figure shows outliers. For example, the two red data points at the upper right of the figure are unexpected. This is not the place where iliac/pelvis acquisitions are expected.

3.5.2 Ceiling models

In the second iteration we are interested in a subset of the data which is extensive enough for modeling and is able to generalize to a broader set of systems. In order to meet these

requirements the assumptions have to be reevaluated. Within the early assumptions only floor models are taken into account. This has proven to provide a subset of the data which is not suitable for modeling.

A ceiling model is the standard choice due to the advantages this type of system has compared to floor models. Floor models have specific use cases and are generally installed when the building is unable to support a ceiling model. This results in a small number of floor models compared to the ceiling models. Furthermore, the ceiling model is able to reach every body part of the patient regardless of the patient orientation.

In order to incorporate the ceiling models we have to take a look at assumption one: '*When having a Clea floor mounted system and the patient lies with his feet at the head end of the table, it is likely that a vascular protocol is used.*'. Looking at the patient position in exams with a ceiling model, it is observed that this patient orientation is hardly present whereas the 'regular' patient position is present in most exams. In order to find a dataset which is more extensive, assumption one is disregarded. In the second iteration, all exams which have a switch between one of the legs protocols and either the 'iliac/pelvis' protocol or 'abdomen' protocol are taken in account. Incorporating the ceiling models results in a subset of data in which the number of unique exams is 7415, which is an significant increase from the 429 exams of interest of the floor models.

3.6 Data limitations

In this section two aspects of the limitations are discussed. As mentioned in section 2.2 the main data sources of this project are the event logs which are collected by Philips, where the focus is on the geometry settings of the system. In section 2.4 several data challenges are mentioned which have been partially overcome as described in this chapter.

The limiting factor in the available data is that little is known about the patient. The patient is an important factor, every patient is different in virtually every aspect. In our research the patient demographics are of great importance, especially the height and weight. The X-ray protocols are specific to certain locations on the patient and listen closely. In the event logs age is recorded consequently but the weight of the patient is recorded in only 22% of the cases. Additional information about the patient is not known to *Philips* through the event logs.

Furthermore, the label noise as described in 3.1.3 is one of the limitations of the data. Through assumptions the label noisy is minimized but not ruled out. The percentage of label noisy is unknown, which adds to the challenge of validating the models.

Chapter 4

Modeling

The next phase in the CRISP-DM cycle is the modeling phase. Several modeling techniques are applied and their parameters are tuned. The problem is a supervised classification problem where we distinguish between two classes. Multiple machine learning techniques can be used for this kind of problem and this section explores several of these techniques. No dataset is the same and there is no strict rule of thumb of which model should be applied, "if one model better than another in some domains, then there are necessarily other domains in which this relationship is reversed." (Rokach and Maimon, 2008).

4.1 Model design

The design and configuration of the models is discussed in this section. The goal of the models is to make an accurate prediction with high precision and recall. Several machine learning techniques are explored and the best one is selected and optimized.

Four machine learning techniques are selected, these are: random forest classification, XGBoost classification, logistic regression and a neural network. Due to the exploratory nature of this thesis, multiple type of techniques are selected. With random forest classification and XGBoost classification, two decision tree based models are included. A neural network is included due to the complex relationships which are expected within the dataset. Last, logistic regression is included since it it simpler than the techniques mentioned before. If a relatively simple model is capable of good performance, the complex models might not be needed. The individual techniques will be elaborated upon in the next section. These models are optimized by using hyperparameter tuning. Furthermore, the impact of data scaling and feature reduction is discussed and compared.

4.1.1 Random forest

A random forest (RF) can be used for regression and classification problems. It is an ensemble learning technique this can be described as running a 'base' learning algorithm several times and the majority vote will decide the final result (Dietterich et al., 2002). The 'base' algorithm is a decision tree in the case of a random forest. A random forest classifier will consist of many individual decision trees and decide, by majority vote, on the classification. In this way

it will overcome that one decision tree may not be optimal. By incorporating multiple trees, a global optimum should be found.

A decision tree consists of nodes and leafs. In each of the nodes a test on an attribute is performed and the data splits in two or more subsets depending on the outcome of this test. At the end of the tree the data reaches a leaf which is an end node. In the leaf of a tree the data is not split further but a classification is provided, along with a probability. The decision tree will continue to grow until one of the stopping criteria is met, examples of stopping criteria are: the number of nodes, the number of leafs and the depth of the tree. By tuning these parameters over-fitting of the decision tree can be prevented (Rokach and Maimon, 2008). In figure 4.1 a simple decision tree is illustrated. Here the root node has the decision whether *BeamLongitudinal* is smaller than 5000 and the internal node has decision *PatientPosition=Up*. The decision tree illustrated is a simplification of the actual decision trees obtained.

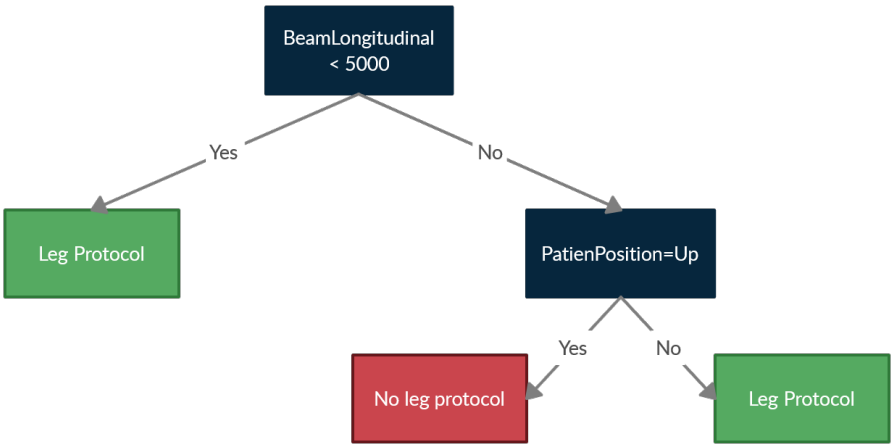


Figure 4.1: Example of a decision tree

An advantage of a random forest is that it can provide the importance of each feature. The model will tell what the most important features are to predict the target label. Using the feature importance it is possible to make a selection of the features which are the most important in our dataset which can be useful when we try to decrease the number of features as discussed in section 3.3. Furthermore, the models are based on a set of decision trees. These trees can be viewed when the model is trained, all the decisions can be tracked. This can be an advantage when the model has to be explained. On the other hand this can be a disadvantage when the model allows for a large number of trees with a high depth.

4.1.2 Logistic Regression

Logistic regression is a classification algorithm often used when the target is binary. It uses a logistic regression function which has a S-shaped curve which takes a value between 0 and 1. The input features are linearly combined using weights to predict the target. Logistic regression predicts a probability between 0 and 1, this outcome is used to determine the class

where the default cut-off score is 0.5. Logistic regression is defined as:

$$log\frac{P}{1-P} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

Where P is the probability of an event and β represent the coefficients. Every β corresponds with a feature, when logistic regression is used in a binary setting, a positive β contributes towards a prediction of 1 and a vice versa.

In this use case, the main disadvantage of using a logistic regression algorithm is the expectation that the algorithm will struggle to find the optimal solution when there are complex relationships within the dataset. Furthermore, the training time can be extensive on a large, high-dimensional dataset. An advantage of logistic regression is the easiness of implementation and simplicity of the model.

4.1.3 XGBoost

XGBoost is an ensemble algorithm which uses a gradient boosting framework in combination with decision trees, introduced by Chen and Guestrin (2016). Boosting models are trained in succession where every new model will be trained to correct the errors made by the previous models (Friedman, 2002). Estimators are added until the stopping criteria is met or no improvements are made. Chen and Guestrin (2016) have proposed a scalable method which makes accessible for large high dimensional datasets by several optimizations.

Over-fitting is prevented by a regularized learning objective, shrinkage and subsampling of features. Shrinkage was introduced by Friedman (2002), in this step newly added weights are scaled by a hyperparameter. By applying shrinkage the influence of individual trees is reduced in order to leave space for future trees to improve the model. Feature subsampling reduces the correlation between trees similar to the random forest.

The advantages and disadvantages of the XGboost classifier are comparable to the random forest. In addition to the advantages mentioned in section 4.1.1, the XGBoost uses a gradient boosting framework. Through this framework the advantages of gradient boosting are combined with the random forest technique.

4.1.4 Artificial Neural Network

An artificial neural network is explored to look for non-linear relationships between the features. The specific type of neural network that will be used is a multilayer perceptron (MLP). This algorithm is based on the perceptron algorithm of Rosenblatt (1958). an MLP is a feed-forward neural network consisting of an input layer, one or more hidden layers and an output layer. The number of neurons in the input layer corresponds to the number of features in the dataset and number of neurons in the output layer equals the number of classes. Every neuron in a hidden layer is interconnected with all neurons of the adjacent layers with different weights. Due to the architecture consisting of multiple layers, non-linear relations can be found by the MLP. In figure 4.2 a schematic representation of a multilayer perceptron is illustrated.

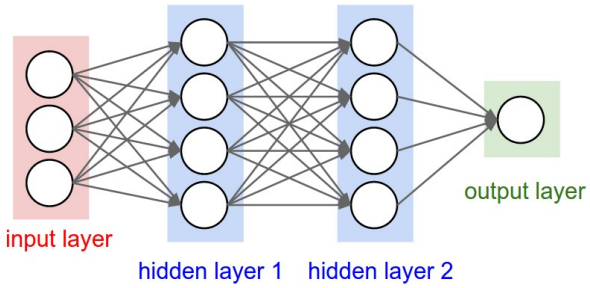


Figure 4.2: Example of an MLP architecture, adapted from 'Convolutional Neural Networks for Visual Recognition' (2020)

an MLP is a feedforward network and learns through backpropogation. The learning consists of two stages, the forward pass and the backwards pass. In the forward pass the data flows from the input layer through the hidden layer to the output node. In the output layer a decision on the label is made which is measured against the ground truth label. During the backwards pass backpropogation is used to change the weights and bias of all neurons. Through this process the MLP attempts to obtain the minimal difference between the model classification and the ground truth label.

An advantage of an MLP over the previously mentioned algorithms is the ability to derive complex relations within our dataset. On the other hand, this model is much harder to interpret and more of a black box model than other algorithms.

4.1.5 Cross-validation

The dataset is split in a training and validation set. Training will be performed on the training set using cross-validation to find a model which performs best. By using cross-validation we can test the model without using the validation set. One of the most used technique is *k-fold* classification. Here the training data is divided in k folds. The classifier is learned using $k-1$ folds and tested on the remaining fold. The *k-fold* estimation error is the average error of all folds. When the best model is selected it will be trained on the training data and evaluated on the validation data.

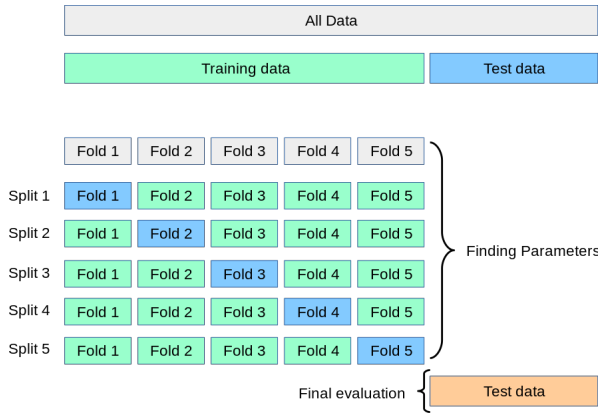


Figure 4.3: K-fold cross validation structure, adapted from SKlearn (2020)

The dataset is split into a validation set and training set. This split is based on the case identifier, every exam has an identification number which is used. By making a split on the identification number, it is guaranteed that data points of an exam are either in the training or in the validation set. This method of splitting the data is also applied when cross-validation is used. The validation set is used to get baseline models and when comparing the final models. The training set is used for training of the models. Due to the high number of data points combined with the high number of features, in the default and scaled models cross-validation is not used. Instead the training data is split in one training and one test set opposed to multiple folds. This is due to the extensive time it takes to train the models. Cross-validation is used in the other steps.

4.1.6 Hyperparameter optimization

All the models have its own hyperparameters which can be tuned to optimize the model. Grid search and manual search are the strategies which are used most (Bergstra and Bengio, 2012). A new method called *Random Search* is proposed by Bergstra and Bengio (2012). With grid search, a set of values for each hyperparameter is set by the user. Gridsearch will evaluate all the Cartesian products of the set (Feurer and Hutter, 2019), whereas Random Grid will evaluate random searches of the grid. This approach works best when some hyperparameters are unimportant for the performance. In figure 4.4 the differences are illustrated, if there is a fixed budget (B) of evaluations, grid search can afford to only evaluate $B^{\frac{1}{N}}$ for each of the N hyperparameters. Whereas B unique values are evaluated by using random search (Feurer and Hutter, 2019).

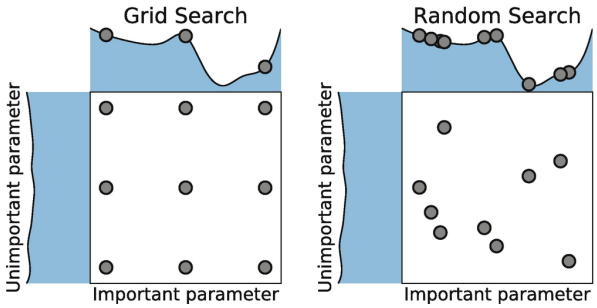


Figure 4.4: Differences between Grid Search and Random Search , adapted from Feurer and Hutter (2019)

Every algorithm has different hyperparameter to optimize, for each algorithm several important parameters will be explained. The RF has five important parameters; *n_estimators, max_depth, min_samples_split, min_samples_leaf and max_features*. These are all straightforward definitions, the number of estimators is the number of trees in the random forest, where every tree has a max depth. The minimal number of samples for a split is the required number to split an internal node in the individual decision tree and every leaf has a minimal number of samples. XGBoost has four important parameters; *eta, min_split, max_depth and min_child_weight*. Eta can be seen as the learning rate, smaller steps are used when it is decreased. The minimum child weight is a value to control the shrinkage, explained in section 4.1.3. The last two parameters correspond with the parameters of the RF.

4.1.7 Cost-sensitive learning

Cost-sensitive learning introduces a cost to a misclassified label and will try to minimize these costs. This method is often used when there is a high cost if a case is misclassified. For example, when a doctor wants to determine if a patient has cancer or not, missing cancer (the patient is classified as negative but is positive) has a larger influence than vice versa. Furthermore, unbalanced data can favor the majority class (Longadge et al., 2013), within the cost-sensitive learning there are methods to account for this. Two methods of cost-sensitive learning are explored: sampling and probability thresholds. Each of the methods will be discussed in the next section and applied in section 4.2.5.

Sampling

Sampling is used to change the class distribution of the training data. If there is an imbalanced dataset, techniques which are centered around accuracy tend to models which favour the majority class. This class usually has a lower cost of misclassification (Seiffert et al., 2008). Two techniques to change the distribution of the data are oversampling and undersampling and are illustrated in figure 4.5.

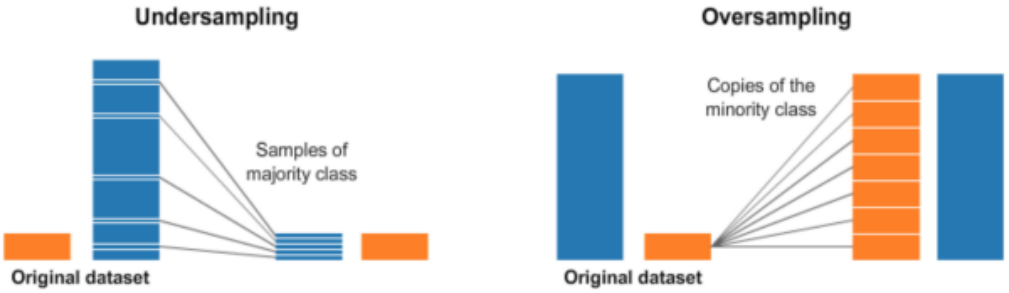


Figure 4.5: Undersampling and Oversampling, adapted from Badr (2019)

Figure 4.5 illustrates the differences between undersampling and oversampling. Undersampling removes data from the majority set. By removing these data points useful information may not be taken in account. Oversampling increases the minority sample, when these are randomly created data points they may cause over-fitting. Chawla et al. (2002) proposed Synthetic Minority Over-sampling Technique (SMOTE). This is an over-sampling approach where the minority class is over-sampled by creating 'synthetic' examples. It uses a k nearest neighbors algorithm to generate synthetic samples. These samples are generated as follows: the difference between the sample and its nearest neighbor is taken and multiplied with a random value between 0 and 1. This value is then added to the sample, giving a point on the line between the sample and its nearest neighbor. Furthermore, Chawla et al. (2002) introduce SMOTE-NC where it takes in account the categorical variables.

Probability thresholds

Making use of the probability thresholds is sometimes referred to as thresholding, it is a simple and effective method to make classifiers which produce probability estimates cost-sensitive (Sheng and Ling, 2006). As the name implies, it uses the probability estimate generated by

the model. This probability is used to evaluate the model and in a standard evaluation the threshold is 0.5. If a data point has a prediction probability larger than or equal to 0.5 it is classified as 1. This threshold can be shifted based on the total costs of misclassifications to improve the model.

4.1.8 Evaluation

There are several measures to evaluate the performance of the model. Since we have a binary classification problem, we will make use of the confusion matrix which allows for visualization of the performance of the models. In table 4.1 a confusion matrix is illustrated.

		True label	
		1	0
Predicted label	1	TP	FP
	0	FN	TN

Table 4.1: Basic confusion matrix

The predictions are split in four categories in a confusion matrix: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). As can be seen in the table, when a model predicts a 1 and the true label is 1, this sample corresponds to the TP cell. As we have a binary classification problem, each of the cases will be in one of the four cells. From the confusion matrix several evaluation measures can be derived: Accuracy, Precision, Recall and the F1 score. Every measurement will be discussed next.

Accuracy is an evaluation metric which is well known and tells the number of labels which are classified correctly. The accuracy in a binary classification problem can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

If the data is imbalanced, accuracy is not a good measurement. By predicting all cases to the majority class a good accuracy can be achieved. Since our data is slightly imbalanced, the accuracy is not very important during evaluation.

Precision, also known as the predictive positive value, is calculated by dividing the number of correctly classified positive cases (TP) by the total number of cases labeled by the model as positive, $precision = \frac{TP}{TP+FP}$. Precision is hardly used as single evaluation metric and often coupled with recall, also called sensitivity. Recall is defined as $recall = \frac{TP}{TP+FN}$. And is the percentage of true positive labels we have identified as such.

The F1 score combines the precision and recall in one evaluation measurement. It can be calculated as follows:

$$F1 = 2 \times \frac{(Precision \times Recall)}{Precision + Recall}$$

The F1 score is used to seek a balance between the precision and recall when there is an imbalanced dataset.

4.2 Model implementation

In this section the previously mentioned models are implemented and results on the dataset are presented. First, the basic models are implemented in which all algorithms will have basic settings which are not tuned. Thereafter, step by step, the models will be extended through scaling, feature selection and hyperparameter tuning. After these models have been implemented cost-sensitive learning will be applied.

The data will be split in a training and validation set, the validation set will not be touched until the model is evaluated and thus will all be unseen data. When applying feature selection and hyperparameter tuning, k-fold cross validation is used on the training set to create a split between test and training data. When the optimal values are found the model is trained on the full training data and evaluated on the validation set.

4.2.1 Basic model

The machine learning algorithms in the default settings is applied to the training set and are evaluated using the validation set. For the basic model no scaling is applied, all available features are included, no hyperparameter tuning is performed and cost-sensitive learning is not applied.

The four different machine learning algorithms all have their own default settings, these remain unchanged for the basic models and are changed in section 4.2.4. LR has one setting which is not the default setting, it uses the 'saga' solver which is used in larger datasets. The default settings of the models are:

RF: n_estimators=100, max_depth=None, min_samples_split=2, min_samples_leaf=1, max_features=auto
XGBoost: eta=0.3, min_split_loss=0, max_depth=6, min_child_weight=1
MLP: hidden_layers=[128, 1], activation=['relu', 'sigmoid'], optimizer='Adam'
LR: solver= 'saga'

This are not all parameters for every model but only these will be optimized in section 4.2.4 and therefore only these are mentioned. The MLP can consist of a number of hidden layers and the default model used in this research consists of two layers, one with 128 neurons and one with 1 neuron and activation functions 'relu' and 'sigmoid', respectively. In table 4.2 the evaluating metrics on the validation set are illustrated. As explained in 4.1.5, due to the training time cross-validation is not applied. Looking at the individual models we notice large differences between the models. The RF and XGBoost models already perform relatively well with an F1 score of 0.723 for the RF and 0.716 for XGBoost. The MLP performs not as good with an F1 score of 0.643. LR is unable to make accurate predictions this could be an early indication that the problem is too difficult to learn for the algorithm. It is expected that the MLP and LR will perform better when scaling is applied to the data.

	Accuracy	Precision	Recall	F1 score
RF	0.822	0.782	0.674	0.723
XGBoost	0.817	0.770	0.669	0.716
MLP	0.760	0.661	0.625	0.643
LR	0.323	0.219	0.373	0.276

Table 4.2: Results of the basic model on the validation data

The evaluation metrics of the models on the validation set are calculated in order to be able to see the improvement of results on the validation set with the fully tuned model. In table 4.2 these result are illustrated. In the following steps the models will be improved and better performing models will be found. In the end the improvements of the models will be compared to these basic models. It is shown that LR, RF and XGB score worse on the validation data whereas the MLP performs better. Table 4.3 shows the scores on the test data.

	Accuracy	Precision	Recall	F1 score
RF	0.799	0.751	0.648	0.696
XGBoost	0.800	0.743	0.662	0.700
MLP	0.788	0.725	0.646	0.683
LR	0.376	0.174	0.204	0.188

Table 4.3: Results of the basic model on the test data

4.2.2 Scaled model

The first step is to apply a transformation to the data which is scaling. The two scaling methods tested are normalization and standardization. Where normalization changes the range of the data, in standardization the distribution is changed as well. It is expected that the MLP will benefit the most from scaling the data while the RF and XGBoost will hardly be influenced. Both decision tree models are based on 'IF' conditions which will remain almost the same except the numbers are scaled.

For the following results, scaling is applied to the data, all features are used, the default hyperparameters are used and cost-sensitive learning is not applied. The results on the validation set for RF, XGBoost, MLP and LR can be found in tables 4.4, 4.5, 4.6 and 4.7, respectively.

	Accuracy	Precision	Recall	F1 Score
RF	0.822	0.782	0.674	0.723
RF MinMax	0.813	0.770	0.655	0.708
RF StandardScaler	0.813	0.772	0.649	0.705

Table 4.4: RF scaling results on test data

	Accuracy	Precision	Recall	F1 Score
XGBoost	0.817	0.770	0.669	0.716
XGBoost MinMax	0.808	0.773	0.629	0.693
XGBoost StandardScaler	0.801	0.778	0.629	0.695

Table 4.5: XGBoost scaling results on test data

As mentioned before, the results for the decision tree based models are not expected to change significantly when the data is scaled. The RF and XGB models perform slightly worse when scaling is applied. Therefore, for RF and XGB the unscaled data will be used.

	Accuracy	Precision	Recall	F1 Score
MLP	0.760	0.661	0.625	0.643
MLP MinMax	0.784	0.716	0.621	0.665
MLP StandardScaler	0.797	0.713	0.693	0.703

Table 4.6: MLP scaling results on test data

The results of MLP are shown in table 4.6. A slight improvement is seen in several of the evaluation criteria when MinMax scaling is applied. Differences between the scaling methods are clear, the F1 score is better when the StandardScaler opposed to the MinMax scaler. For the next sections standard scaled data is used when applying MLP models.

	Accuracy	Precision	Recall	F1 Score
LR	0.323	0.219	0.373	0.276
LR MinMax	0.776	0.719	0.577	0.640
LR StandardScaler	0.774	0.715	0.575	0.638

Table 4.7: LR scaling results on test data

In table 4.7 the results for LR are shown. The LR model has a significant improvement in the F1 score. When comparing the scores of the LR to the other models we see that the LR score is lower which further indicates that the problem might be to difficult to solve for a LR model. The scores for the MinMax scaler and StandardScaler are very close, the standard scaled data is used to evaluate LR in the following sections.

4.2.3 Feature selection

Feature selection helps to rank and select features based on an algorithm. Different models may require different subsets of features to reach the optimal performance, therefore the features selection algorithms are performed for each model. The method used for feature selection will consist of two steps.

1. Reduce the number of movements
2. Apply a feature selection algorithm

The first reduction step is to see what number of movements is optimal for the model. To determine this, the models are fitted to a number of movements from 1 to 30 using a 5-fold cross validation. It is expected that the movements closest to the acquisition are most important, so from 1 to 5 every movement is tested. Thereafter, the step size is gradually increased. The second step is to use a feature reduction algorithm on the remaining features to determine which are important. The algorithm used is Recursive Feature Elimination with cross-validation (RFECV).

Random Forest

The first step is to fit the RF to the number of movements. The results are illustrated in figure 4.6 and show that the RF prefers to have less movements selected. If we select one movement the mean cross validation score is lowest, increasing the number of movements does increase the F1 score. The next step is to use a feature selection algorithm to the remaining features. When 15 movements are included we have a total of 867 features. Recursive feature elimination with 5-fold cross-validation and a step size of 50 is applied. The algorithm starts with 867 features, every step the 50 least important features are eliminated. This process is repeated until only one feature remains.

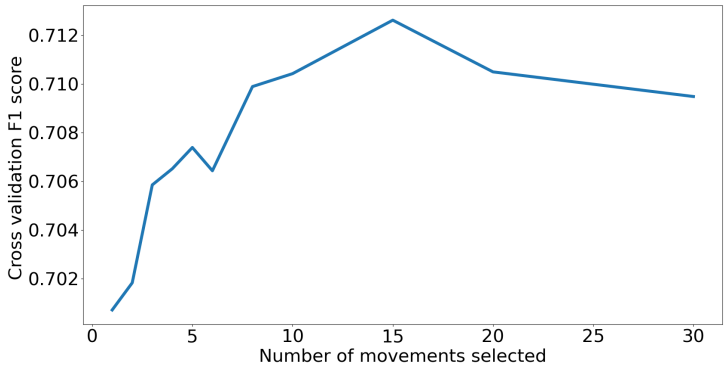


Figure 4.6: Number of movements selected RF

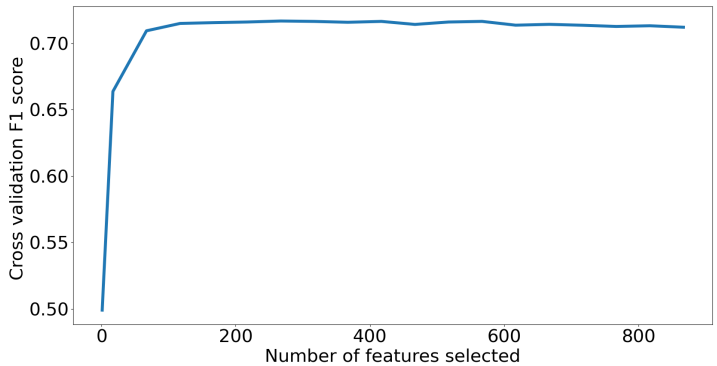


Figure 4.7: Results of RFECV

Figure 4.7 illustrates the F1 score for the number of features used. The F1 score gradually increases up to 0.709 when 67 features are selected. Thereafter there are small increases and the highest F1 score is 0.717 when 267 features are selected. The F1 score keeps on this level for the remaining number of features but never exceeds it. The feature set with 267 features is used in the following steps.

XGBoost

XGBoost is trained on the same number of movements in the first step as the RF. The F1 score is increasing up to 20 movements included, as is illustrated in figure 4.8. RFECV is applied starting at 20 movements. For this classifier the step size is increased to 100 due to the high number of features included. After running RFECV with 5-fold cross-validation the highest score F1 score obtained is 0.706 when 112 features are included. This feature set is used in the following steps.

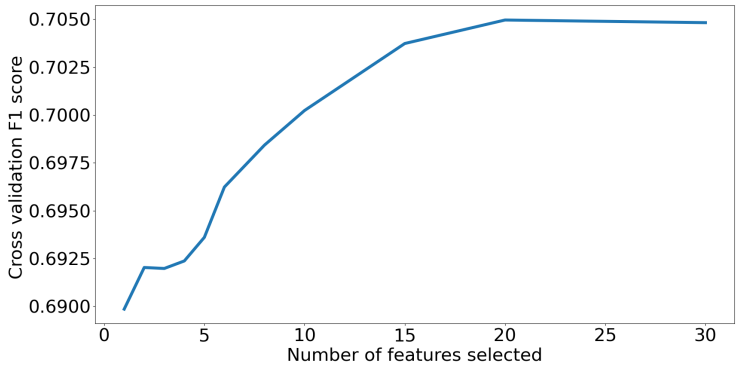


Figure 4.8: Number of movements selected XGB

Logistic Regression

The F1 score of LR keeps increasing as the number of movements selected increase. Therefore, feature reduction is applied with a step size of 100. RFECV with 5-fold cross-validation was not able to fit all models due to time constraints. The reduced feature set of RF and XGB are applied in order to reduce the feature set but both provide a F1 score of 0.62. Since there are no results for RFECV, the model has been under-performing compared to the alternative models and the expectation that LR would be unable to find complex relationships in the data as mentioned in section 4.1.2, the decision is made to not take LR in account in the next sections.

MLP

Feature selection for the MLP will differ from the method above since the MLP has no feature importance, performing recursive feature elimination using the sklearn algorithms is not possible. In order to test if the MLP benefits from a reduced amount of features 5-fold cross-validation is applied for the number of movements. Cross-validation was applied but the results were inconclusive. The mean cross-validation scores of the movements are all

between 0.700 and 0.730 with the highest scores at 20 selected movements. What makes the results inconclusive is the standard deviation, which ranges from 0.017 to 0.034. Increasing the number of folds in the cross-validation and increasing the patience of the neural network would likely lower the standard deviation and provide a more reliable number. The number of folds is not increased due to the training time of cross-validated neural networks. As mentioned before the highest scores of the 5-fold cross validation are at 20 movements included. Together with the expectation that the number of movements closest to the acquisition are important, the choice is made to include 20 movements. To do further comparison, an MLP with the optimal feature set of the RF and XGB is trained.

5-fold cross-validation is used to compare the three configurations. In table 4.8 the results are shown for all three configurations. It is clear that the MLP prefers to have all 20 movements as input and benefits from more available information.

Included features	Mean F1 score	Standard deviation
20 movements	0.739	0.030
RF features	0.710	0.014
XGB features	0.712	0.017

Table 4.8: Feature selection results of MLP

4.2.4 Hyperparameter tuning

Tuning the hyperparameters is an important step to improve the results of the models. In order to tune the hyperparameters sklearns RandomizedSearchCV is used. For each of the models a parameter grid is constructed. Random search selects the parameters randomly and will use 5-fold cross validation. 30 different sets of parameters are chosen and evaluated using the F1 score resulting in a total of 150 fits to be performed. The parameters with the highest F1 score are used in cost-sensitive learning.

Random Forest

We see that tuning the hyperparameters for the RF model increases the 5-fold cross-validation F1 score to 0.723. The final parameters are: *number of estimators: 638, minimal sample split: 4, minimal sample leaf: 3, maximum features: 'auto', max depth: 19, bootstrap: False*. The parameters which are the most interesting are the sample split and sample leaf, combined with the max depth. This model can have deep decision trees which are based on four samples in a split and three at the leaf. This allows for over-fitting.

From the hyperparameter tuning results we see that a 19 of the 25 combinations have a mean F1 score between 0.710 and 0.723, all with a standard deviation between 0.011 and 0.013. This indicates that the model performance is close together for different combinations. Since there is a indication of noisy labels it might be beneficial to use a model which is more robust and is on the conservative side regarding over-fitting.

XGBoost classifier

The hyperparameters of the XGBoost model are tuned using 5-fold cross validation. The best mean F1 score is 0.715 with a standard deviation of 0.010. The final parameters are: *eta:*

0.286, gamma: 5, max depth: 17 and minimal child weight: 0. The low eta indicates that the step shrinkage is low and thus makes the estimator less conservative. On the other hand, the max depth and minimal child weight allow for deep individual decision trees with a low number of data points at each leaf which allows for over-fitting.

MLP

The MLP itself does not have parameters which can be tuned. Instead the structure of the MLP is changed, the MLP used so far has two hidden layers with 128 and 1 neuron(s). A simpler model of two hidden layers with 64 and 1 neuron(s) is fitted using cross-validation to see if this is sufficient to learn the problem. Furthermore, a more complex model is fitted, this model consists of three hidden layers with 128, 64 and 1 neuron(s). In addition two dropout layers are added between the hidden layers in which 20% of the neurons are randomly set to 0 during training, which helps prevent over-fitting. In table 4.9 the results are shown. We see that the results are close but the MLP structure with 128 and 1 neuron(s) has the highest F1 score. This model will thus be used for cost-sensitive learning.

Model structure	Mean F1 score	Standard deviation
[128, 1]	0.739	0.030
[64, 1]	0.712	0.022
[128, 64, 1]	0.722	0.019

Table 4.9: MLP structures

4.2.5 Cost-sensitive learning

The last step in the model implementation is cost-sensitive learning. Two cost-sensitive approaches are explored, sampling and probability thresholds. Before applying cost-sensitive learning we will train the final models and evaluate using the validation set. In figure 4.10 the scores of the final models are illustrated. If we compare these scores to the initial models we see that the RF and XGB improve minimally with an increase of 0.012 for the RF and 0.013 for the XGB. The F1 score of the MLP decreases with 0.002.

	Accuracy	Precision	Recall	F1 score
RF	0.806	0.758	0.664	0.708
XGBoost	0.812	0.775	0.661	0.713
MLP	0.788	0.731	0.637	0.681

Table 4.10: Results on validation set

Sampling

Sampling is used to balance the distribution within the training set. Due to the structure of the data the sampling is difficult. Group based cross-validation is used which requires every data point to be in a group. In our setting this group is an exam. When oversampling is used, every new point should be included in an exam. The structure of exams makes it hard to recreate data points which represent the exams in the training data. Whereas undersampling

only removes data points from the exam. A random undersampler is used which selects a random sample of the data points and removes these.

In table 4.11 the results are shown with random undersampling applied. All three models benefit from a balanced data set and increase in F1 score. Other sampling methods are not used due to the group structure.

	Accuracy	Precision	Recall	F1 score
RF	0.793	0.686	0.724	0.724
XGBoost	0.802	0.712	0.740	0.726
MLP	0.779	0.674	0.727	0.700

Table 4.11: Random undersampling on validation data

Probability threshold

The probability estimates can be calculated for each of the three remaining models. The evaluation criteria are checked for all threshold scores between 0.1 and 0.9 with an interval of 0.05. If the prediction probability is above the threshold, it assigned a 0 otherwise it is labeled as 1. If the threshold is 0.9 the model must have at least a certainty of 90% to assign a 0.

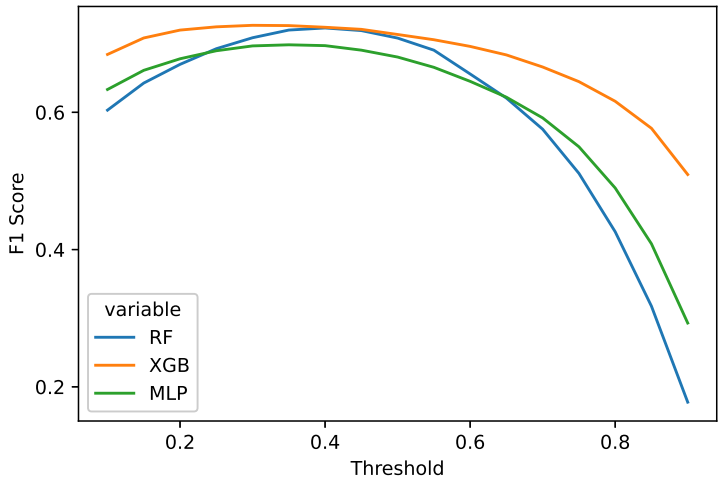


Figure 4.9: Threshold scores

In figure 4.9 the threshold scores are illustrated. We see that when the threshold is increased the F1 score increases for each model. The best scores are shown in table 4.12. The best scores for the models are at threshold 0.4 for the RF, 0.3 for XGB and 0.35 for the MLP. If we compare the threshold scores we see the F1 score increase marginally.

	Accuracy	Precision	Recall	F1 score
RF	0.794	0.691	0.758	0.723
XGBoost	0.798	0.697	0.760	0.727
MLP	0.771	0.656	0.747	0.698

Table 4.12: Threshold scores on validation data

Hybrid

The last step of cost-sensitive learning is to combine both methods. The models will use random undersampling combined with the best found threshold. The results are shown in table 4.13. The results show that combining both methods does improve he RF model marginally, XGBoost has the same F1 score as thresholding and the MLP performs best when sampling is applied.

	Accuracy	Precision	Recall	F1 score
RF	0.794	0.687	0.767	0.725
XGBoost	0.798	0.697	0.760	0.727
MLP	0.773	0.660	0.742	0.699

Table 4.13: Hybrid scores on validation data

4.2.6 Results

In the first step all models are applied with the basic settings. RF and XGBoost have similar scores where RF is the best model with a F1 score of 0.723. In the next step scaling is applied to the dataset, this resulted in improvements for the MLP and LR models. These improvements are significant, the F1 score of LR and MLP increases from 0.276 to 0.640 and from 0.643 to 0.704, respectively. For both models standard scaling is used for the remainder of the models. The RF and XGBoost are affected only slightly due to these models being decision-tree based models. The best model remains the unscaled RF model with a F1 score of 0.723.

The following stage is to apply feature elimination, this stage is performed for each model separately and consists of two steps. The number of movement is reduced, thereafter recursive feature elimination with cross-validation is applied. RF includes 15 movements and 267 features in the final set with an F1 score of 0.717. XGBoost includes 20 movements in the first step and selects 112 features with an F1 score of 0.706 in the second step. For MLP a different approach is taken since this model can not provide feature importances. The first step is applied and results in 20 selected movements with a F1 score of 0.739. In addition to this step the feature set of RF and XGB are applied, resulting in a F1 score of 0.710 and 0.712, respectively. The best MLP model uses 20 movements. LR performs best with 30 movements included, RFECV is not applied due to time constraints and LR is not used in further analysis.

Hyperparameter tuning is performed using a random search over a parameter grid. 30 sets of parameters are randomly chosen and evaluated using 5-fold cross-validation. The F1 score of RF increased to 0.723 after tuning the parameters which is an increase of 0.006 compared to the model after feature selection. XGBoost has an F1 score of 0.715 after tuning, an increase of 0.009. MLP has no parameters to be tuned, instead the structure of the MLP is changed. Two variations are implemented but both do not improve the model.

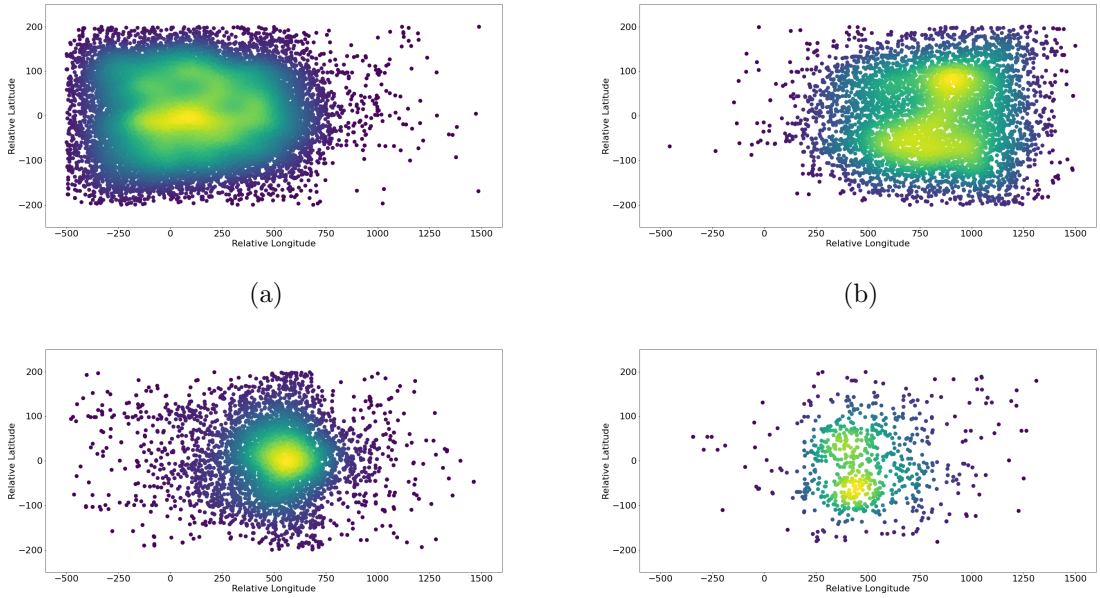
The last step is cost-sensitive learning, before this is applied the best models found thus far are applied on the validation set. When comparing the validation scores to the validation scores of the basic model improvements can be seen but they are marginal. Two techniques are used in cost-sensitive learning, namely sampling and thresholding. Furthermore, a hybrid solution of both is applied. Sampling and thresholding improve the F1 scores. The best RF model is the hybrid model with a F1 score on the validation set of 0.725 which is an increase of 0.029 compared to the basic model. The F1 scores of the thresholding and hybrid models are exactly even with an F1 score of 0.727, an increase of 0.027 compared to the basic model. The MLP performs best when sampling is applied, the F1 score is 0.700, increasing by 0.019.

The model which performs best is the hybrid XGBoost model with a F1 score of 0.727, an accuracy of 0.798, a precision of 0.697 and a recall of 0.760. When comparing these results to the basic model an improvement is observed however, this improvement is marginal and is mostly due to cost-sensitive learning. In order to be deployed in a real-life scenario the results should be improved.

Analysis of the model results

In this section the results are analyzed with the goal of finding an indication where the model is under performing. The results of the XGBoost model are used for analysis of the results. First, we will look at the position of correct and incorrectly labeled data points. Thereafter, the trajectories are discussed.

In figure 4.10 the correctly and incorrectly labeled data points have been plotted per patient position. The patient position can change in two ways, the standard patient position has the patient lying on his back with his head at the side of the C-arm. This position is called head up and legs down. In all the exams of interest the patient lies on his back but the orientation of the legs differ. In the figures the relative positions have been plotted, this is the position of the focal point adjusted for the movements of the patient table, so if the table would have been fixed this is the position of the focal point. The x-axis is the relative longitude in millimeters compared to the center point of the table. The y-axis is the relative latitude in millimeters.



(a) (b)

(c) (d)

Figure 4.10: Heatmaps of correctly (a, b) and incorrectly (c, d) labeled leg protocols per patient position (legs down (a, c), legs up (b, d))

In figure 4.10 (a) the position of correctly labeled leg down protocols are plotted. The heatmap illustrates that a significant amount of the acquisitions which are correctly labeled have a relative longitude less than 750 millimeters. In figure 4.10 (c) the position of incorrectly labeled leg down protocols are plotted which range mostly between a longitude of 250 and 750. The differences between legs down and legs up protocols are visible, in figure 4.10 (b) the correctly labeled leg down protocols mainly have a relative longitude higher than 500. Looking closely at figure 4.10 (a) and (b), the difference between the left and right leg faintly visible. However, the distinction is not as clear as in figure 3.4 in section 3.5.1.

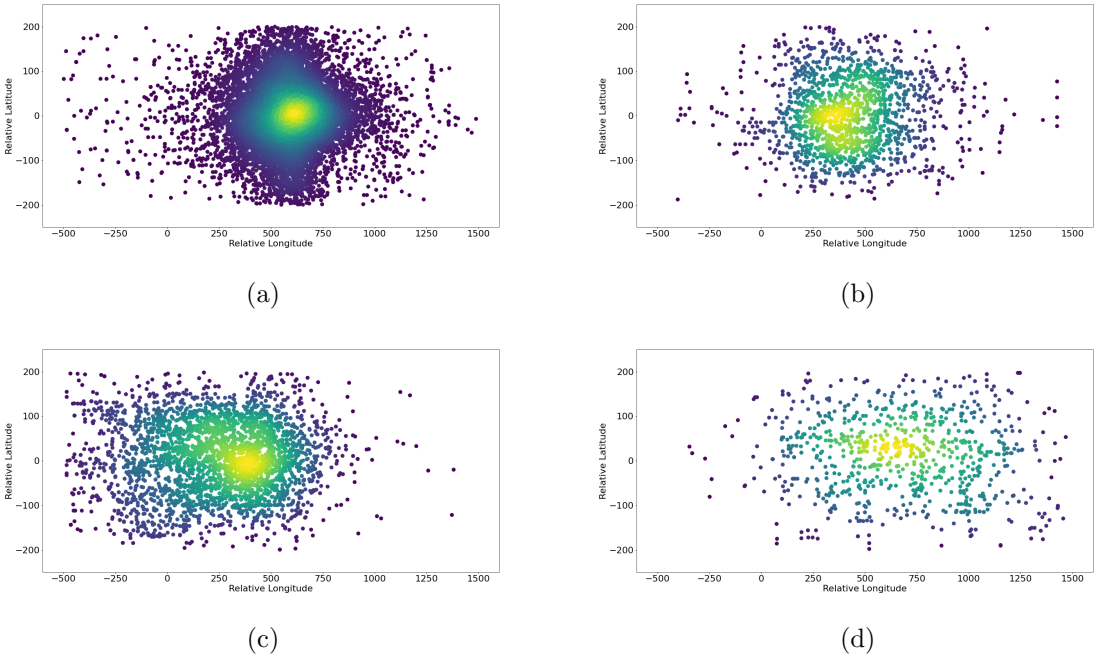


Figure 4.11: Heatmaps of correctly (a, b) and incorrectly (c, d) labeled iliac/pelvis or abdomen protocols per patient position (legs down (a, c), legs up (b, d))

Figure 4.11 illustrates the heatmaps for the correctly (a) and incorrectly (b) labeled acquisitions of iliac/pelvis and abdomen protocols. The correctly labeled acquisitions center around a relative latitude of 0 for both patient positions, whereas the relative longitude is differs per patient position. Note that the incorrectly labeled iliac/pelvis or abdomen protocols are spread out more across the table than incorrectly labeled leg protocols, both in the latitude as well as the longitude direction.

When the correct and incorrect labels are compared, multiple observations are made. Through the observations we identify where the model struggles to make the correct decision. It is observed that the correctly labeled leg acquisitions on average has a lower relative longitude than iliac/pelvis or abdomen acquisitions. When observing the leg down protocols, the opposite is true for the leg up protocol.

Furthermore, leg protocols are performed more to the sides of the table which is illustrated on the right-hand side in figure 4.12 (a) and (b), especially (b) shows a higher density just off the centre of the table which corresponds with the right and left leg. The differences between the relative longitude of patient positions is visible. In figure 4.12 (a) a leg down position is used, here the acquisitions with a leg protocol have a relative longitude between -500 up to 1000 whereas a leg up position has a relative longitude between 0 and 1500.

An interesting observation is that the correctly and incorrectly labeled iliac/pelvis or abdomen acquisitions have a comparable distribution in the longitudinal direction, in the latitudinal direction the distribution are not as similar (see fig. 4.12). Here the incorrectly labeled acquisitions deviate from the center more than in the correctly labeled data.

The findings above indicate that the incorrectly labeled legs acquisitions are close to the correctly labeled iliac/pelvis or abdomen acquisitions whereas the incorrectly labeled iliac/pelivis or abdomen acquisitions are spread more around the table. It seems that the model is not able to predict in this 'edge' area properly. This area is located around the upper thighs and lower pelvis of the patients, here small differences of the focal area can mean that the wrong protocol is used.

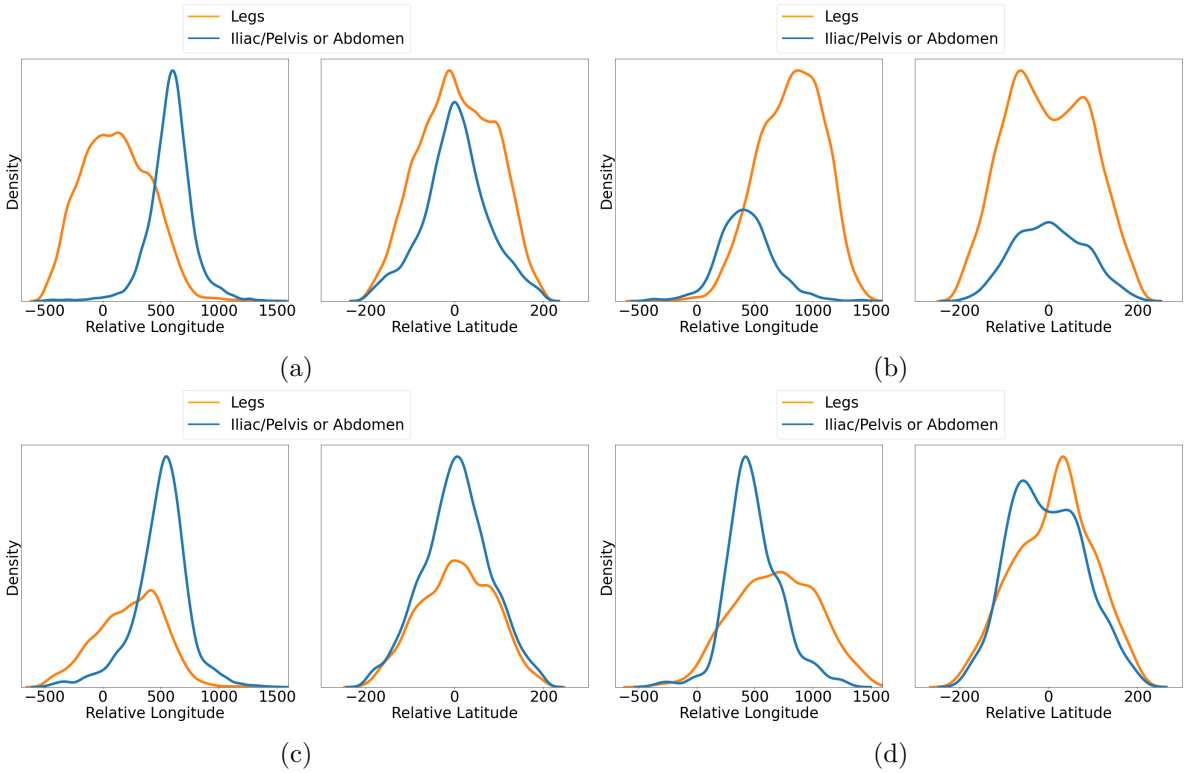


Figure 4.12: Density plots of correctly (a, b) and incorrectly (c, d) labeled protocols per patient position (legs down (a, c), legs up (b, d))

Two moments during the exam are of special interest namely: the start of the exam and the switches which occur during the exam. From the business understanding, it is known that the first acquisition is expected to be of the iliac/pelvis or abdomen protocol. The validation set consists of 2225 unique exams, in 10% of the exams the model does not predict the correct starting procedure. The first acquisition is in 84% of the exams an iliac/pelvis or abdomen protocol. When an exam starts with an iliac/pelvis or abdomen protocol one percent is predicted incorrectly whereas an exam started with a leg procedure is predicted incorrectly in 58,3% of the exams.

The second moment of interest is a switch in during the exam. The 2225 exams in the validation set have 3064 switches in total, of which 47,7% are predicted incorrect. On average it takes 2.82 acquisitions to predict the correct protocol. Of the 10676 incorrect predictions, 8624 are associated with a switch. Thus, showing that identifying the exact moment of the switch is important.

Lastly, the trajectories of the exams will be discussed. In the positional data outliers have been found, these data points have values which are outside the dimensions of the table. Exams are removed when one or more outlier is present in the exam, 1467 exams remain in the validation set. When analyzing the trajectories we see a large variety between exams. As mentioned before, 84% of the exams start with an iliac/pelvis or abdomen protocol, this is visible in the trajectories. Exams start relatively centered, ranging from a relative longitude between 250 and 750. After the first acquisition(s), trajectories vary widely. In figure 4.13 5 randomly selected trajectories are illustrated to show the differences. The trajectories do not provide additional insights without additional information of the patient or the goal of the exam.

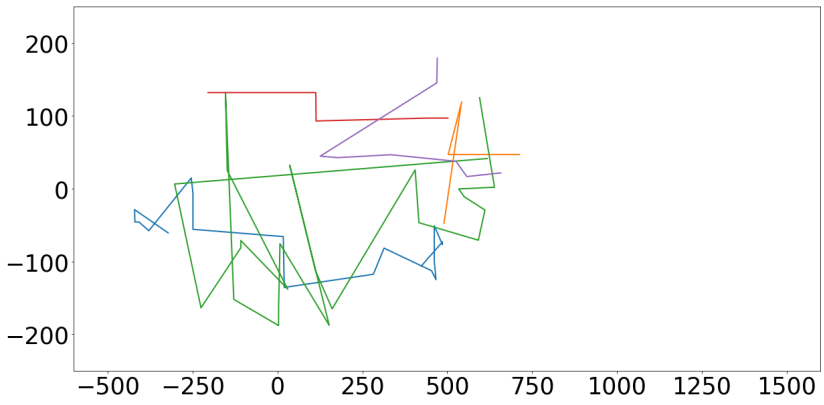


Figure 4.13: 5 trajectories of random exams

Chapter 5

Validation and Discussion

In this chapter the validation and results of the models are discussed. In the discussion, the last step of the CRISP-DM model, deployment, is described. Furthermore, the limitations are mentioned and future work is outlined.

5.1 Validation

Validation of the models is difficult due to the unknown extend of label noise. Through a set of assumptions, the label noise is limited. However, in the data there are still data points which seem to have an incorrect label.

As mentioned in section 4.1.5, the data is split in a training and a validation set. The validation set is solely used for evaluating the model. The best performing XGBoost model has an accuracy of 0.798 and a F1 score of 0.727 on the validation data. From the analysis of the results it is observed that a high amount of the incorrectly labeled data points are in the 'edge' area around the upper thighs and lower pelvis, based on the relative position of the table. This indicates that for these data points there is not enough information to make a clear distinction. A number of these points may have an incorrect label but as the analysis showed, large differences between the exams exist. This makes identifying incorrectly labeled data points in this edge area difficult. On the other hand, outliers are visible in the validation set. Data points which, based on the location should have a different label exist.

Before cost-sensitive learning, the tuned models are compared to the basic models and showing minimal improvements. The RF and XGB improve with an increase of 0.012 and 0.013, respectively. The F1 score of the MLP decreases with 0.002. The tuned models add complexity. Taking the XGBoost model as example, the default parameters for max depth is 6 and a minimal child weight of 1. This are relatively simple decision trees which are not deep. Looking at the tuned models we see that the max depth increases to 17 and the minimal child weight decreases to 0. Both allow for additional complexity within the models. Deeper trees increase complexity by increasing the number of nodes and the lower minimal child weight delays partitioning. Simple models are preferred when taking in account the explain-ability of the models, "shallow decision trees are intrinsically interpretable, so that explaining its predictions becomes a trivial task" (Samek and Müller, 2019).

Furthermore, during the analysis it is observed that 80.8% of the incorrect predictions are related to a switch, showing that the switch is an important event in the exam. Only a switch between two protocols has to be predicted in this study due to the choices which are made. When looking at more protocols it is expected that the differences between the protocols are smaller. The four protocols have been divided in two groups with similar protocols. Predicting the four protocols individually is expected to be harder, which would result in a decrease in performance. Taking in account that this is only a small selection of all available protocols, including additional protocols will increase the complexity and is likely to lower the performance.

5.2 Discussion

The discussion is split in several subsections, starting with a general discussion. Thereafter, the last phase of the CRISP-DM model, deployment, is discussed. Last, the limitations and future work are discussed.

As the results indicate, there is value in the available data at *Philips* but context information is missing. The main available data source is the *Vertica* database which provides an extensive amount of information however, extensive information about the patient is missing. There is information available about the age and to a small extend about the weight of the patient. Arguably the most important patient demographic for this study, length, is not included. Additionally, due to the structure of the data, user behavior is included. This can affect the validity of the model. Using user-generated labels can not only introduce label noise by switching at the incorrect moment but also introduce user behavior in the model. There may be clinicians who prefer to switch later than others or use the machine in a slightly different way. The way of working between hospitals may differ and while they can both be correct, this introduces differences based on user behavior in the dataset.

5.2.1 Deployment

Deployment is the last phase of the CRISP-DM cycle, in this section a deployment strategy for the model is discussed. A full deployment strategy and deployment itself is not the aim of this research. Although results are not sufficient to be used in a real-life scenario, several ideas on the design of the implementation will be discussed. These options revolve around three topics, namely: should the switch be automated or manually selected and how is this information presented to the clinician? Furthermore, the time complexity of the model is discussed, is the model able to run in real-time?

The switch between protocols would be evaluated after each movement. In practice this means that after each *command:stop* event, the position is retrieved and the model predicts a protocol. As mentioned in section 2.1 the Azurion platform includes a touch screen module and a Flexvision or a set of monitors. These are the main sources of information for the clinician. Presenting information about the switch can be done in multiple ways, e.g. a notification on the TSM or Flexvision or integration in the default layout of the TSM and Flexvision.

The choice of an automated switch versus a manual switch or confirmation of the switch will depend on several factors. The main factor will be the performance of the model. As

mentioned before, the performance of the model is currently not sufficient. Depending on the performance, the switch could be fully automated if the model performs outstanding however, given the environment in which a mistake can have large impact on the patient an automated switch may not be desirable. Manual switches could be implemented as a notification with the suggestion for a switch, which can include an option to switch immediately. An approach which minimizes risks, is easy to use and does not impact the workflow is the goal.

As mentioned above, the goal is to present a prediction after each movement. This requires a low time complexity to perform the prediction, the prediction has to be presented as soon as the movement is over or even during the movement. Ideally the model runs in real-time and at each moment in time a prediction is made. When a difference between the current and predicted protocol is observed, a switch is recommended.

5.2.2 Limitations and future work

As mentioned before, the current performance does not meet the requirements for deployment. In this section several limitations will be discussed and how future work can overcome these limitations.

The main limitation on the current model is the input data, there is hardly context information available. It is unknown what the patient is treated for and what the demographics of the patient are. There are multiple data sources which can help to improve the model but are unavailable for *Philips* due to privacy regulations, e.g. medical records and medical imaging. However, these data sources are locally available in the hospital. Furthermore, currently the label noise limits the study, the labels are user-generated and introduce noise and user behavior in the dataset. Both limitations are discussed in the following subsections, along with directions in which future work can be pointed.

Medical image recognition

Medical image recognition is used in a wide variety of ways. An example is lesion detection, which has been around for decades. Drukker et al. (2002) investigated the detection lesions on breast ultrasounds. In recent years research has shifted to deep learning models. Much research has been done on image recognition in the medical domain. The domain of interest which can help *Philips* is recognition and classification of parts of the body.

Zare et al. (2013) present a combination of discriminative and generative approach on classification of medical X-ray images. The ImageCLEF 2007 medical dataset (IC dataset) is used, this consists of 11,000 X-ray images. Labels for this database consist out of eight main human body regions which are subdivided in sub-regions, in total 116 sub-regions are present in the dataset. The best accuracy rate Zare et al. (2013) achieve is 92.5% on the sub-regions. These regions include: ‘upper legs’, ‘lower legs’, ‘knee’, ‘lower abdomen’, ‘middle abdomen’, ‘iliac bone’ (Lehmann et al., 2003). Being able to classify medical images in such specific regions will help to understand what the focal point of the acquisition is. The abdomen protocol of *Philips* encompasses multiple sub-regions in the IC dataset.

A second use case of image recognition could be the estimation of stature. Stature estimation has been studied extensively (Sarajlić et al., 2006; Ismail et al., 2018; Ramezani et al., 2019) and a variety of bones are used to estimate the stature. Literature shows that ”long bones

are popular and provide accurate results" (Ismail et al., 2018). Ismail et al. (2011) show that regression equations can be derived based on the length of the femur, tibia and fibula. These are three bones located in the leg, estimating the stature based on the length of either one of these bones or a combination of the bones showed a correlation. In a scenario where the length of the patient is unknown, stature estimation can be used.

If the focal point of interest is defined in more detail, the geometry data will be put in a different perspective. Imagine a scenario in which the geometry data shows a downward movement and the last acquisition is in the 'upper abdomen' region it is likely that the following X-ray protocol will still be in the abdomen and shifted to the region 'middle abdomen'. Using image recognition to understand the focal point better can result in a different decision. If the last acquisition would have been in the 'lower abdomen', the downward movement is more likely to shift to a leg protocol. The same applies to stature estimation, if there length of the patient can be estimated, it can help the classifier predict more accurate.

The difficulty for *Philips* in obtaining training data for such a model are privacy regulations, medical images made during an exam are not available for *Philips* in general. However, the hospitals have access to these images locally. An opportunity for *Philips* is to deploy a trained model on the Azurion platform which uses the medical images locally to support the prediction of X-ray protocol. Combining image recognition with the available geometry data of *Philips* could improve the performance of the models.

Medical records

The second data source which is available in the hospital is the medical record of the patient. It encompasses all medically relevant information about the patient, e.g. surgical history, medication as well as demographics. The medical record provides a lot of information about the specific patient which can provide features for a model. Evident pieces of data which are relevant are the patient demographics. The height and weight of the patient are important factors. In this project we have seen that the model is under performing in areas where a small difference means a switch of protocol. Adding key patient specific information can increase the performance.

Directly adding the height and weight of the patient to the dataset can be seen as low hanging fruit. If there is access to this data, it is a relative straightforward step to add these variables. A second step could be to use the patient demographics to build a estimation of the body proportions. This is expected to be more precise than solely adding the weight and height of the patient to the model. An additional benefit of a 'body model' is that the dose area product (DAP) can be customized per patient. The DAP is the amount of radiation required for a clear image. An advanced step could be to use additional information about the medical record. Examples are: matching images of the current exam with images in the patient's medical records or use natural language processing to look for keywords in the medical records which can indicate what sort of intervention is performed during an exam. Again, the privacy regulations will have to be taken in account.

Trajectory of exam

During the analysis of the results we have seen that the trajectories of exams vary widely. Some similarities have been identified but the trajectories have not been explored extensively.

Future work could focus on this subject. The purpose of the exams is unknown while this is an important factor in the analysis of trajectories. The purpose can be to perform an intervention or an exploratory exam to have a look and diagnose a patient. Information about why certain movements are performed helps to understand behaviour of the clinician.

Label Noise

At this moment, the event logs obtained from the Azurion platforms are extensive and not designed for data mining purposes. The labels chosen for this thesis are the X-ray protocols selected by the clinician who performs the exam. This is where this project is limited, it is currently impossible to verify whether the selected protocol is correct, which introduces label noise. In this thesis we have made a set of assumptions in order to minimize the label noise within our use case. During the first iteration it is observed that the floor models have data points which are likely to be labeled incorrect. It would be interesting to individually the trajectories of such exams, as well as exams which have outliers.

A reliable, extensive dataset where the labels are verified would improve the models. Constructing such a dataset should be a goal for *Philips*, there are several methods to obtain this dataset. However, most methods are labour intensive and require a hospital to cooperate. Future work should focus on the question if it would be feasible to build a dataset which has less label noise or build a model which is more robust and can perform when label noise is present.

The directions of the future work mentioned in this section can provide valuable information for *Philips* to improve the Azurion platforms. Being able to add and combine several of these data sources could provide substantial value for all parties involved. Being able to accurately predict the X-ray protocol will help the clinician choose the right protocol. Which will also benefit the patient by having a DAP that is better suited for the acquisition. Reducing the radiation in the room is beneficial for every person in the room, clinician, lab staff and patient.

The idea of the current model is to run in real-time. Adding one of the data sources mentioned before will affect the computational complexity. Depending on the implementation, image recognition will increase the time complexity of the model to a certain degree. During the discussion on the implementation it is mentioned that after or during each movement a prediction is presented. In order to be useful the time between the end of the movement and the prediction should be limited. Otherwise, the clinician may continue with the incorrect protocol.

Chapter 6

Conclusions

6.1 Research goals

In this section the conclusions are discussed. First the research goals are discussed and thereafter the research question is answered.

Research goal 1: *Identify the available data sources, the limitations and create a relevant dataset in context of the Azurion 7 platform*

The main data source is the event log of the Azurion platforms. These provide extensive information about all aspects of the system. The event logs regarding the geometric position and the acquisitions during exams are used to create a dataset. This dataset consists of exams in which a switch between the protocols 'Upper Legs' or 'Lower Legs' and 'Iliac/Pelivs' or 'Abdomen' occurred. Whereas the dataset provides extensive information about the geometric position, the context information is limited. Through extensive preprocessing the dataset has been cleaned and transformed to make it suitable for modeling. The main limitation of the dataset is the minimal context information available. In most exams there is hardly information about the patient except for the age, other demographics are not included or not precise. Another limitation is the presence of label noise in the dataset. Labels are user-generated which allows for incorrect user behavior in the dataset.

Research goal 2: *Identify a model which is able to accurately predict the next user action while using the Azurion 7 platform*

Several machine learning models have been selected and the performance of these models have been evaluated at each step. A logistic regression, random forest classifier, XGBoost classifier and multilayer perceptron have been selected. First, a base result was obtained for comparison between the models. Thereafter a number of steps were taken in order to improve the results. Scaling is applied, relevant features are selected, hyperparameters are tuned and cost-sensitive learning is applied. The combination of the steps resulted in marginal improvements for the models. The best performing model is the hybrid XGBoost classifier with a F1 score of 0.727, an accuracy of 0.798, a precision of 0.697 and a recall of 0.760. The improvements are small when comparing the tuned model to the basic model and an accuracy of 79.8% is insufficient. The fact that performances only slightly increase from the

basic model to the final model indicates that this is the best the models can perform given the dataset.

Research goal 3: *Explore how the model can be deployed and in what way the prediction can be presented to the user*

The model can support the selection of the X-ray protocol by continuously predicting the appropriate protocol. In our case this means that after each *command:stop* event the model predicts a protocol and present this information to the clinician. There are several ways in which the information can be presented which depend on the model performance, the aim is to minimize the risk and have an easy-to-use solution which does not impact the workflow during the exam.

Main research question:

> *"Given a system event log, is it feasible to improve user experience through prediction of user actions in systems with complex human-machine interaction?"*

Based on the research goals, it is concluded that it is not feasible to improve the user experience through prediction of user action in our case study. The prediction of X-ray protocols using the available data at *Philips* is not accurate enough. Providing a solution which meets the performance requirements is challenging. Data exploration showed that the event logs provide a lot of information however, only a small subset of the data is suitable for achieving the research goals. During the data preparation phase, many steps are taken to transform the event logs in a usable and relevant dataset. The modeling phase showed that multiple models have comparable results with small differences. However, none of the models are able to make predictions with a performance that is sufficient to implement the model. The small improvement between the basic and final models indicate that given the current data, the models are performing as best as they can and there are significant differences between exams. The analysis of the results shows that the model struggles in the upper leg and lower pelvis area. Small differences can change the X-ray protocol in this area. Given that there is hardly information available about the patient, this does not come as a surprise. Additional data is required in order to provide a solution which meets the performance requirements.

In the case study it proved to be difficult to predict the next user action. The Azurion platform is a system in which the interaction between user and machine is complicated. Additionally, the context information is important for the use of the system. If the method is applied to different use cases where the event logs are more extensive or additional context information is available, it may be feasible to use prediction of user actions to improve the user experience.

Bibliography

Badr, W. (2019). *Oversampling versus undersampling*. Retrieved August 20, 2020, from https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

Baldominos Gómez, A., Albacete Garcıa, E., Marrero, I. & Saez Achaerandio, Y. (2016). Real-time prediction of gamers behavior using variable order markov and big data technology: A case of study.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*(1), 281–305.

Cai, L. & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data science journal*, *14*.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system, In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*.

Cs231n convolutional neural networks for visual recognition. (2020). https://cs231n.github.io/neural-networks-1/ (accessed: 09.08.2020)

Dietterich, T. G. Et al. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110–125.

Drukker, K., Giger, M. L., Horsch, K., Kupinski, M. A., Vyborny, C. J. & Mendelson, E. B. (2002). Computerized lesion detection on breast ultrasound. *Medical physics*, *29*(7), 1438–1446.

Feurer, M. & Hutter, F. (2019). Hyperparameter optimization, In *Automated machine learning*. Springer, Cham.

Freeman, C., Kulić, D. & Basir, O. (2015). An evaluation of classifier-specific filter measure performance for feature selection. *Pattern Recognition*, *48*(5), 1812–1826.

Frénay, B. & Verleysen, M. (2013). Classification in the presence of label noise: A survey. *IEEE transactions on neural networks and learning systems*, *25*(5), 845–869.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367–378.

Grus, J. (2019). *Data science from scratch: First principles with python*. O'Reilly Media.

Han, J., Pei, J. & Kamber, M. (2011). *Data mining: Concepts and techniques*. Elsevier.

Hoornaar, T. (2017). *Extracting real-life workflow models from relational data and using these to generate field-based usability testing scenarios at philips healthcare* (Master's thesis). Einhoven University of Technology. the Netherlands.

Ismail, N. A., Abd Khupur, N. H., Osman, K., Shafie, M. S., Nor, F. M. Et al. (2018). Stature estimation in malaysian population from radiographic measurements of upper limbs. *Egyptian Journal of Forensic Sciences*, *8*(1), 22.

Ismail, N. A., Abidin, A. H. Z., Hamzah, S. P. A. A., Osman, K., Hamzah, N. H. Et al. (2011). Stature approximation of malays, chinese and indian in malaysia using radiographs of femur, tibia and fibula. *Jurnal Sains Kesihatan Malaysia (Malaysian Journal of Health Sciences)*, *9*(1).

Janssen, C. P., Donker, S. F., Brumby, D. P. & Kun, A. L. (2019). History and future of human-automation interaction. *International journal of human-computer studies*, *131*, 99–107.

Jensen, K. (2012). *A diagram showing the relationship between the different phases of crisp-dm and illustrates the recursive nature of a data mining project.*

Lachin, J. M. (2016). Fallacies of last observation carried forward analyses. *Clinical trials*, *13*(2), 161–168.

Lehmann, T., Güld, M., Thies, C., Fischer, B., Spitzer, K., Keysers, D., Ney, H., Kohnen, M., Schubert, H. & Wein, B. (2003). The irma project: A state of the art report on content-based image retrieval in medical applications, In *Korea-germany workshop on advanced medical image*.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J. & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, *50*(6), 1–45.

Longadge, M. R., Dongre, M. S. S. & Malik, L. (2013). Multi-cluster based approach for skewed data in data mining. *Journal of Computer Engineering (IOSR-JCE)*, *12*(6), 66–73.

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Low, D. K.-W., Newman, S.-F., Kim, J. Et al. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, *2*(10), 749–760.

Nokelainen, P., Nevalainen, T. & Niemi, K. (2018). Mind or machine? opportunities and limits of automation, In *The impact of digitalization in the workplace*. Springer.

Ozmen, M. M., Ozmen, A. & Koç, Ç. K. (2020). Artificial intelligence for next-generation medical robotics, In *Digital surgery*. Springer.

Pietraru, A. (2018). *Anti-pattern detection and analysis in data-driven product design* (Master's thesis). Einhoven University of Technology. the Netherlands.

Ramezani, M., Shokri, V., Ghanbari, A., Salehi, Z. & Niknami, K. A. (2019). Stature estimation in iranian population from x-ray measurements of femur and tibia bones. *Journal of Forensic Radiology and Imaging*, *19*, 100343.

Rokach, L. & Maimon, O. Z. (2008). *Data mining with decision trees: Theory and applications* (Vol. 69). World scientific.

Rong, M., Gong, D. & Gao, X. (2019). Feature selection and its use in big data: Challenges, methods, and trends. *IEEE Access*, *7*, 19709–19725.

Rosenblatt, F. (1958). *The perceptron: A theory of statistical separability in cognitive systems*. United States Department of Commerce.

Saeys, Y., Inza, I. & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, *23*(19), 2507–2517.

Samek, W. & Müller, K.-R. (2019). Towards explainable artificial intelligence, In *Explainable ai: Interpreting, explaining and visualizing deep learning*. Springer.

Sarajlić, N., Cihlarž, Z., Klonowski, E. E. & Selak, I. (2006). Stature estimation for bosnian male population. *Bosnian journal of basic medical sciences*, *6*(1), 62.

Sardar, P., Abbott, J. D., Kundu, A., Aronow, H. D., Granada, J. F. & Giri, J. (2019). Impact of artificial intelligence on interventional cardiology: From decision-making aid to advanced interventional procedure assistance. *JACC: Cardiovascular Interventions*, *12*(14), 1293–1303.

Sarker, I. H., Kayes, A. & Watters, P. (2019). Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data*, *6*(1), 57.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J. & Napolitano, A. (2008). A comparative study of data sampling and cost sensitive learning, In *2008 ieee international conference on data mining workshops*. IEEE.

Shen, J., Li, L., Dietterich, T. G. & Herlocker, J. L. (2006). A hybrid learning system for recognizing user tasks from desktop activities and email messages, In *Proceedings of the 11th international conference on intelligent user interfaces*.

Sheng, V. S. & Ling, C. X. (2006). Thresholding for making classifiers cost-sensitive, In *Aaai*.

Sklearn, cross validation. (2020). https://scikit-learn.org/stable/modules/cross_validation.html (accessed: 12.08.2020)

Stead, W. W. (2018). Clinical implications and challenges of artificial intelligence and deep learning. *Jama*, *320*(11), 1107–1108.

Strong, D. M., Lee, Y. W. & Wang, R. Y. (1997). Data quality in context. *Communications of the ACM*, *40*(5), 103–110.

Tabaksblat, R. (2019). Retrieved September 1, 2020, from https://www.philips.com/a-w/about/news/archive/standard/news/press/2019/20191127-one-millionth-procedure-carried-out-on-philips-azurion-advanced-image-guided-therapy-platform.html

Tan, F., Wei, Z., He, J., Wu, X., Peng, B., Liu, H. & Yan, Z. (2018). A blended deep learning approach for predicting user intended actions, In *2018 ieee international conference on data mining (icdm)*. IEEE.

Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature medicine*, *25*(1), 44–56.

Uku, R. (2019). *Data-driven product design : Discovering potential for automation by analyzing user behavior* (Master's thesis). Einhoven University of Technology. the Netherlands.

Wirth, R. & Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining, In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Springer-Verlag London, UK.

Wu, X., Zhu, X., Wu, G.-Q. & Ding, W. (2013). Data mining with big data. *IEEE transactions on knowledge and data engineering*, *26*(1), 97–107.

Yu, L. & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution.

Zare, M. R., Mueen, A., Awedh, M. & Seng, W. C. (2013). Automatic classification of medical x-ray images: Hybrid generative-discriminative approach. *IET Image Processing*, *7*(5), 523–532.

Zhou, L., Pan, S., Wang, J. & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361.