Universiteit Leiden
The Netherlands

Opleiding Informatica

Predicting the Discharge Date

of Patients using Classification Techniques

Valérie Paul

Supervisors:
Matthijs van Leeuwen
& Bram Wesselo

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Abstract

The expenses of hospitals are increasing due to medical and technological development. The hospital's objective is to improve the quality of health care while reducing costs. The length of stay is one of the performance indicators that is used as a measure of efficiency of a hospital. It is essential to know the discharge date in advance to optimize the use of human resources and facilities. This thesis aims to research how the discharge date can be predicted more accurately using various classification machine learning techniques.

Several classification models have been used to predict the discharge date. In particularly machine classification and regression tree, random forest, naive bayes, and support vector machine have been applied and compared using the F1-score. Based on the results, the model classification and regression tree appears to be the most insightful model to predict the discharge date. This model has been tuned and visualized.

Moreover, this research shows how challenging it is to work with raw medical data of patients. It gives an insight into how the data is extracted and represented. It also shows how missing values are imputed using different methods.

Contents

Chapter 1

Introduction

Expenditures in health care and the number of patients are increasing due to medical and technological developments [7]. To manage the growing health care expenditures, the cost-effectiveness of hospitals should improve. The ambitions of hospitals are to establish an adequate health care planning and organization while reducing the cost of health care, maximizing the use of resources (material and human) and improving the quality of health care.

One of the performance indicators in a hospital is the average length of stay (ALOS). It is acknowledged as the primary indicator used to measure efficiency [18]. The prediction of the length of stay (LOS) for individual patients is essential to optimize the planning and resources of a hospital while minimizing the effect of uncertainties. Knowing the date of discharge of a hospitalized patient in advance will enable the more efficient use of human resources and facilities in a hospital [18].

Much research is done on the cost efficiency of medical protocols, treatments, and medical operations. However, the discharge date is still manually predicted based on the expertise of the nurses and specialist.

At Leiden University Medical Centre (LUMC), two departments where patients are taken care of after surgery are interested in predicting the discharge date more accurately. At one department the predictions are made relatively well, but in the other department, there are more possible influences that might affect the length of stay of the patient which makes it challenging to estimate the discharge date.

One solution is to predict the discharge date of each patient using machine learning. In this research anonymized labeled attributes are available, consisting of the measurements taken when the patient was hospitalized and a patient's history. For instance, we know how many times the patient had an appointment at the hospital, their age, and we know what their heart rate was when hospitalized.

The increasing cost pressure and the accountability for the given specialist's care and length of stay within health care make it essential to create a model that can predict the discharge date of a patient more accurately. The presented predictive classification model in this thesis is constructed using machine learning. There are two main types of machine learning:

- Supervised learning: Making predictions about the future
- Unsupervised learning: Discovering hidden structures

The objective of the machine learning model that will be created is to make predictions, therefore we will concentrate on supervised learning. Supervised learning is conducted with labeled data. Labeled data is a group of samples that have been tagged with one or more labels. These tags indicate to which group the samples belong. Regression and classification are two subcategories of supervised learning. Regression is used to predict a continuous variable, while classification is used to predict discrete values. Predicting the exact day of discharge is more difficult than predicting the answer to the question of whether someone will be discharged within 48 hours.

The objective of this research is to develop a classification model that predicts whether a patient is being discharged within 48 hours using machine learning techniques.

1.1 Approach & Contributions

To be able to make a reliable classification model, data has to be collected and pre-processed. Missing data is challenging when working with data from Electronic Health Records (EHRs). EHRs were designed to record and improve patient care and streamline billing, and not as resources for research, which leads to complications when attempting to gain information about someone's health. This research will give insight into how medical data is stored and what necessary steps have to be applied before using it for machine learning.

Learning from imbalanced datasets comes with complicated problems which several learning algorithms do not take into account, which causes poor performance. The larger, less relevant class will be preferred by the objective functions that are used for learning the classifiers. We therefore have selected machine learning algorithms that perform well on unbalanced data such as classification and regression tree and random forest. We also apply the algorithms naive bayes and support vector machine. The algorithms will be evaluated using precision, recall, and F1-score.

The departments that are interested in predicting the discharge date of a patient have a variety of specialisms. Very few studies attempt to predict LOS across all conditions using EHRs. It is the first time a prediction model for the discharge date will be developed for the two departments using machine learning within LUMC. By comparing the different algorithms, classification and regression tree resulted as the best performing model. The model has been tuned and visualized which is shown in Chapter 6.

This research will present relevant work that has been done on classification models using medical data. It will show the pre-processing techniques that have been used and present the best performing machine learning model for predicting the discharge date of a patient.

1.2 Overview

Chapter 2 of this thesis describes the aim of the research and the approach of the problem. In Chapter 3 insight is given into related research. Chapter 4 focuses on the data that has been selected and filtered for this research. This chapter will also explain how missing data is managed. In the following chapter, Chapter 5, different machine learning classification techniques are discussed to find a suitable technique for our problem. In the next chapter, Chapter 6, experiments and analyses are performed, where classification models are compared in order to find the best model. It also shows that the parameters of the best model are optimized, and that the final model is being visualized. In Chapter 7 a discussion is opened about this research, and the thesis ends with conclusions in Chapter 8, with suggestions for possible future work and recommendations.

Chapter 2

Problem and Approach

This chapter introduces the necessary background information, and explains the approach, materials and tools used for this thesis. The aim is to provide an understanding of the problem and its scope.

2.1 Background

The length of stay (LOS) of a patient in a hospital has been widely used as an indicator of hospital performance. The average LOS (ALOS) has decreased for decades and proceeds to drop in the industrialized world [23]. Research has shown, that there is still room for improvement. Decreasing the LOS could reduce the costs per patient and ultimately may raise the number of patients that can be treated [23].

The departments VCH1 (Verpleegafdeling Chirurgie 1) and VCH2 (Verpleegafdeling Chirurgie 2) of the Leiden University Medical Centre (LUMC) take care of patients after they had surgery. Department VCH1 specializes in Oncological surgery, Gastrointestinal surgery, ENT, Skin diseases/Dermatology, Ophthalmology and Oral diseases, and Jaw and Facial surgery. Department VCH2 specializes in Orthopaedics, Plastic surgery, Trauma surgery, Urology, and Vascular surgery. Both departments take care of patients that are expected to stay longer than five days. When the KVVL (Kortverblijf, i.e., short stay) is closed during the weekend, patients are moved to VCH1 or VCH2. These patients stay at most two to three days.

Patients are taken in at different times in these departments. A patient can come in with an empty stomach at seven o'clock in the morning and is taken into surgery the same day. Elective patients that are hospitalized later might have their surgery the next day.

There are different ways to be admitted to departments VCH1 and VCH2. One way is that patients come from a different department in the hospital, like the IC (Intensive Care) or the PACU (Post Anaesthesia Care Unit). Other ways patients are taken in are: from abroad or from another hospital. Patients can be admitted via an appointment or acute. The numbers of acute and elective patients are shown in Chapter 6.

Nurses and specialists try to predict the discharge date based on their expertise and experience. At department VCH2 the prediction is made relatively well this way, but at department VCH1 it is challenging to estimate the correct discharge date.

After a patient has finished their treatment in the hospital, they are ready to be discharged. However, this is not always easy because some patients have difficulty taking care for themselves and might be placed in a care home. Due to cutbacks in medical care, it is challenging to find a suitable care home. This situation causes crowded hospitals with people who do not need treatment by specialists anymore [26].

The main goal of this research is to make the prediction of the discharge date easier and more accurate, the secondary goal is to eventually make the methodology available to the clinicians as a helpful tool.

2.2 Formalizing the Problem

Let the training dataset $D = \{x_i, y_i\}_{i=1}^{n}$ consist of n objects in a d-dimensional space, with y_i being the class label for object x_i. We assume that there are k distinct classes so that $y_i \in \{c_1, c_2, ..., c_k\}$. It can be formalized as:

- x_i = is an object, in our case a time slot that contains the corresponding values for all the attributes
- y_i = the class label of object x_i
- $\hat{y}_i$ = the predicted class of object x_i

The object x_i contains the corresponding values for all the attributes. Possible attributes for this problem can be age, pain score, or the number of appointments. Each object is unique due to its hospitalization number and the time of the timeslot. This will be explained more in detail in Chapter 4. Our problem consists of inputs that have discrete and continuous values, and an output that has precisely two possible values; each example input is either classified as true (patient is being discharged within 48 hours) or as false (patient will not be discharged within 48 hours).

In conclusion the problem can be described as: given the input values of the attributes, denoted by x_i, predict the output y_i, denoted by $\hat{y}_i$, by training a classification model on the training dataset D. The predictions $\hat{y}_i$ will lie in $[0, 1]$.

2.3 Approach

When starting with a classification analysis, the data has to be converted in a suitable format before learning a model. The whole procedure from turning raw data into a meaningful model can be represented in six main steps. The steps are inspired by the Knowledge Discovery in Databases process [10].

1. Objective and scope

   Knowledge about the objective and scope brings direction to the research. It can be used as a guidance while investigating the relevant and irrelevant data points for the scope of the problem.

2. Data collection

   When a solid understanding of the objective is founded, the available data can be assembled. The assembled data will be transformed into a table.

3. Data cleaning

   After the data is collected it needs to be cleaned. Data cleaning enhances the quality of the data and can establish a more accurate data model. In this step, irrelevant data points can be suspended.

4. Data modelling

   The available data is correlated with the medical objectives and to make a accurate prediction model using classification.

5. Evaluate

   By evaluating accuracy, the model that performs best can be chosen.

6. Iterate

   Note that the five steps above are not sequential, but can be executed in an iterative and incremental way to ensure the most accurate model when making valuable conclusions.

2.4 Materials and Tools

For this thesis, we used programming language Python 2.7 with Spyder. For pre-processing the data we used Microsoft SQL Server Management Studio 2017. Visualizations of data are done with Matplotlib and H2o-Flow. The machine algorithms are implemented using Sci-kit Learn.

Chapter 3

Related Work

Machine learning is used in many research areas, and its strengths have drawn attention to the use of the machine learning algorithms for various applications. Nowadays a hospital is a rich data environment due to technological improvements and monitoring systems. Meyfroidt et al. [15] present an overview of machine learning techniques that have been used to examine large patient databases.

Predicting the date of discharge is typically addressed as predicting the length of stay (LOS) of a patient. A perfect situation would know the LOS of the patient at the time of admission. Turgeman et al. (2017) [22] concluded that there is not enough knowledge about the patient when being admitted, hence they suggest predicting the LOS after the patient is hospitalized for 24 hours. The study presents a regression model based on random forest for predicting the LOS of congestive heart failure patients. The study by Pendharkar et al. (2014) [17] also uses regression for predicting LOS. They used classification and regression tree (CART), chi-square automatic interaction detection (CHAID), and support vector regression (SVR), and found no significant differences in performances between the three techniques. CART was eventually chosen because it provides a model that is easy to understand and interpret.

In the research of Yakovlev et al. (2018) [27], in which they present machine learning methods to predict in-hospital mortality and length of stay of acute coronary syndrome patients, classification and regression models are used. Regression was used to predict LOS with artificial neural networks (ANN), and classification models such as k-nearest neighbors (KNN), random forest (RF), logistic regression (LR), and naive bayes (NB) were used to predict if the patient would die or be discharged.

Predicting LOS with classification models has been approached in different ways. Barnes et al. (2015) [3] applied tree-based supervised machine learning methods to predict discharge by two p.m. and the end of the day for patients daily. Morton et al. (2014) [16] evaluated the performance of multiple linear regression (MLR), support vector machines (SVM), support vector machines plus (SVM+), multi-task learning (MTL), and random forests (RF) for predicting long versus short-term length of stay of hospitalized diabetic patients. SVM+ performed best on their data. Haya Salah (2017) [20] used 13 different classification models to predict to which of the three categories: short LOS (≤ 3 days), medium ($4 \leq$ days ≤ 7) and long (days > 7), a patient

belongs. In her research deep belief network (DBN) performed best. The study of Chuang et al. (2018) [8] adopted supervised learning techniques to predict prolonged LOS for general surgery patients. They applied CART, RF, and SVM. The RF method was the most accurate and stable prediction model.

Prediction of LOS is addressed in many studies but mostly in the context of specific diseases. Very few studies attempt to predict LOS across all conditions using EHRs.

Verheijen [25] has attempted to predict the LOS at the ICU at LUMC. The paper describes which attributes were chosen and which classification models performed best. She applies SVM, RF, and CART. As a result, SVM was the best performing model.

In this research we will apply CART, SVM and RF like Verheijen and Chung et al. We will add naive bayes (NB), this algorithm assumes that the attributes are independent. This is not true for medical data. Therefore, this algorithm will be used as a baseline. When the other algorithms perform worse than NB, our model is not beneficial. More information about the algorithms can be found in Chapter 5. Our research differs from Verheijens because we utilize other data and focus on presenting the pre-processing steps more extensive. It is the first time a prediction model for the discharge date will be developed for the two wards VCH1 and VCH2 using machine learning within LUMC.

Chapter 4

Data

Electronic Health Records (EHRs) can be divided into two types of data: structured and textual. Structured data primarily consists of numerical data (e.g., blood pressure) and categorical data (e.g., diagnosis). Textual data consists of free text areas of a patient chart. There are no specific guidelines or constraints, when filling in the text areas, which leads to textual data that is difficult to analyze. This research will only focus on the structured data, exploiting the free text is left for future work.

4.1 Representation and Extraction

The patients in our research were hospitalized after 1 January 2014 and before 17 March 2018. The selected dates are chosen because the wards were reorganized in 2014 and this research has selected 17 March 2018 as the limit, because the datasets we are working with are updated continuously. They have a minimum stay of four days and a maximum stay of 29 days. The maximum length of stay is limited in order to make our data less unbalanced. The patients in this research are either from the ward VCH1 or VCH2.

Patients occur more than once in the datasets we are using. To prevent that, in our evaluation, the prediction model uses data from a patient that appeared in the future to predict what happens to the same patient in the past, we only took one hospitalization of every patient.

When a patient gets hospitalized, the hospitalization gets a unique ID. This ID corresponds to a start-date and an end-date. These dates are used to calculate the total length of stay (TLOS). Between those dates, the patient will be taken into surgery and taken care of in the hospital. There will be checks ups every day, and some measurements are taken when needed.

To summarize all the data, we have chosen to divide the TLOS in timeslots of twelve hours. As illustrated in Figure 4.1a, an example patient comes in on the first of February and leaves the hospital on the fifth of February. The hospitalization is divided into timeslots of 12 hours, which results in eight timeslots. Each such timeslot is represented by a data object x_i. This timeline is a simplified representation of how a hospitalization

could look like for a patient. In Table 4.1b the data of the timeline is presented which is suitable for machine learning.

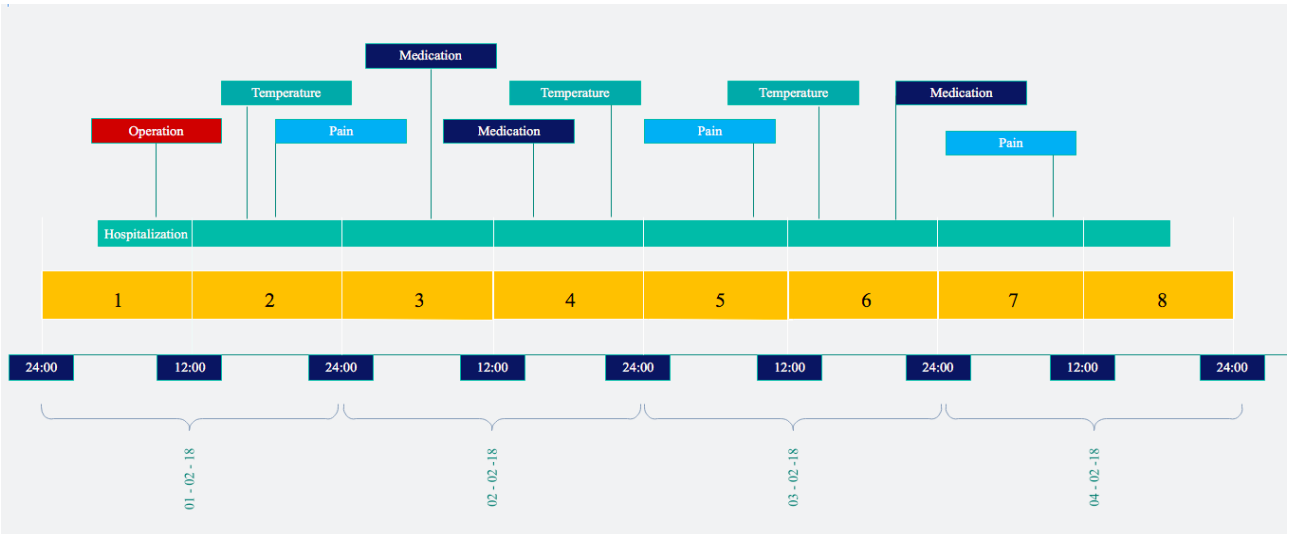

Figure 4.1a: An example of a hospitalization. The patient comes in on the first of February and leaves the hospital on the fifth of February. Their hospitalization, represented by the green line, is divided in eight timeslots, these are represented by the yellow blocks. The timeslots consist of twelve hours, either beginning at 12 o'clock at night (24:00) or at 12 o'clock in the afternoon (12:00).

Timeslot	Hospitalization Number	Patient ID	Date	Total number of Operations	Temperature	Pain Score	Total number of Medications	Discharged within 48 hours
1	987654	1234567	1-2-2018 : 12:00	1	–	–	0	0
2	987654	1234567	1-2-2018 : 24:00	1	38.00	7	0	0
3	987654	1234567	2-2-2018 : 12:00	1	–	–	1	0
4	987654	1234567	2-2-2018 : 24:00	1	37.50	–	2	0
5	987654	1234567	3-2-2018 : 12:00	1	–	5	2	1
6	987654	1234567	3-2-2018 : 24:00	1	37.00	–	3	1
7	987654	1234567	4-2-2018 : 12:00	1	–	2	3	1
8	987654	1234567	4-2-2018 : 24:00	1	–	–	3	1

Figure 4.1b: Corresponding table of the example hospitalization

In Table 4.1 the different approaches in which the chosen attributes were collected are described. As shown the number of medications is counted, but we do not give an indication which type of medication the patient had. Due to the large number of different medications used at these wards in the specified timeframe (there are 2241 different medications) this straightforward approach has been preferred to save time. The attributes that are selected are only a small subset of the available data. For instance, there are 335 different generic measurements to choose from the available data. The chosen attributes are selected based on the advice of clinicians at the two departments and based on what we thought were attributes that would have an influence on the discharge date of a patient.

Attributes	**The way that the data was collected**
PseudoID	Available
Hospitalization Number	Available
Hospitalization Ward	Available
Date	Available
Day of the Week	Based on the date.
Gender	Available
Age	Based on year of birth
Number of Surgeries	The number of surgeries counted between 24 hours before hospitalized and the date of the object. The reason for this is that a patient can have surgery before registered as hospitalized, because urgent care was required.
Number of Lab requests	The number of lab requests counted during 24 hours before the date of the timeslot.
Number of Appointments of type V	Total number of appointments had in LUMC till the date of the object.
Number of Appointments of type T	
Number of Appointments of type E	
Number of Appointments of type Star	
Average Pain Score	The minimum, maximum, and average values that are measured during 24 hours before the date of object.
Minimum Pain Score	
Maximum Pain Score	
Average Temperature	
Minimum Temperature	
Maximum Temperature	
Average Average Blood Pressure	
Minimum Average Blood Pressure	
Maximum Average Blood Pressure	
Average Systolic Blood Pressure	
Minimum Systolic Blood Pressure	
Maximum Systolic Blood Pressure	

Average Diastolic Blood Pressure	The minimum, maximum, and average values that are measured during 24 hours before the date of object.
Minimum Diastolic Blood Pressure	
Maximum Diastolic Blood Pressure	
Average Pulse	
Minimum Pulse	
Maximum Pulse	
Average SpO2	
Average Respiration	
Average FiO2	
Diagnosis	Available
BMI	Available, taken the last measured value.
Length	
Weight	
Medication Taken	The number of medication counted during 24 hours before the date of the object.
Medication Taken Adhoc	
Medication Taken Extra	
Number of Prescriptions	Number of ongoing prescriptions that the patient has on date of the object.
From	Available
Hopitalization Specialism	Available
Current Length of Stay	Total number of days stayed since hospitalization till date of the object.
Urgent/Elective Hospitalization	Available
On the Ward	Comparing date of the object with timeline of patient if patient is on the ward.
Being Dicharged within 48 hours	Binary label; 1 is when discharge date is within 48 from date of object else 0.

Table 4.1: Collected Attributes

4.2 Filtering

Data that has been made available for this thesis is either measured by a machine and automatically saved in the patient's electronic health record or is filled in by a clinician. Machines are not a hundred percent reliable, but because we are working with crucial medical data the assumption of 100 percent reliable output of machines is made. When filling in the information of the patient the system will not give an error when a measurement is filled in incorrectly (this assumption has been made while looking at the data set). For instance, it is not likely that someone is 60 cm long when the patients in the dataset are all above the 16 years

old. Filtering out data that is not correct is a straightforward method. Data that has been typed incorrectly could be corrected. For instance, the value for length is in centimeters, but if someone entered 1.69, this could easily be corrected, but as this method takes a lot of time we have chosen to apply a simple method.

With the help of H2O.ai, we could easily find the outliers and filter them out. Filtering is done by deleting the value of an attribute that is considered an outlier. We used common sense and a clinician's advise to determine the constraints as shown in table 4.2.

Attribute	**Constraint**
Average Blood Pressure	0 <value <500
Systolic Blood Pressure	0 <value <500
Diastolic Blood Pressure	0 <value <500
Pulse	0 <value <500
BMI	11 <value <80
Length	100 <value
Weight	30 <value <200

Table 4.2: Constraints for the attributes it was necessary for

4.3 Missing Data

Missing data is challenging for all studies, and becomes more challenging as the size and scope of data expand. It is particularly problematic for Electronic Health Records (EHRs). EHRs were designed to record and improve patient care and streamline billing, and not as a resource for research, causing complications when attempting to gain information about the human health.

There are three types of missing data:

- Missing completely at random (MCAR)
- Missing at random (MAR)
- Not missing at random (NMAR)

When data is missing completely at random (MCAR), there is no explanation found correlating with other attributes of the dataset on why it should be missing. Missing at random (MAR) data is missing within subgroups of other observed data. It depends on another attribute whether the data is missing or not, for instance, without knowing the weight of the patient, the Body Mass Index (BMI) cannot be known. When the data is not missing at random (NMAR) it depends on the missing values themselves. For example, there is no value for the attribute SpO2 (an estimate of the amount of oxygen in the blood) because the clinician did not need that information about the patient.

In the data representation that is chosen for this research missing data arises in different ways:

1. Filtered out - NMAR
As described in the previous section we have filtered the data. This will also cause missing data that has to be dealt with.

2. Not known yet - NMAR
When a patient comes in at seven o'clock in the morning it could be that there were no measures done till 12 o'clock, this results in no information about some variables in that specific timeslot.

3. Not filled in (actual missing values) - MCAR and MAR
Some information is evident for the clinicians and will not be reported in the EHR, what results in missing information for the data analyst.

4.4 Imputation

The simplest method to deal with missing data is to delete every row that contains missing values. Deleting all objects for which data are missing on at least one variable, is called listwise deletion [1]. This method causes a lot of data loss which is not beneficial when the aim is to maintain as much information as possible. If every object (timeslot) for which data are missing on at least one attribute would be deleted, crucial information about the patient could be deleted. Missing data, therefore, has to be filled in in a way such that the validity of conclusions drawn by this research is not reduced.

In this thesis, various techniques are used to fill in the missing values. These techniques are only applied on the necessary attributes. The used techniques are: filling in the healthy norm, the mean of the attribute, and making the attribute binary.

Filling in the healthy norm

The missing values of the selected attributes in Table 4.3 have been filled in with their healthy norm. When filling in the missing values with healthy norms, we assume that the patient has healthy values when not measured, which will not always be the case. Healthy norms differentiate, i.e., someone that is seventy years old has other healthy norms (e.g., blood pressure [11]) than a professional athlete that is twenty-five years old. To fill in the correct healthy norm for every patient is therefore very challenging. We have chosen to take one healthy norm to fill in for every patient.

Attribute	Healthy value
Average Blood Pressure	83.0
Systolic Blood Pressure	70.0
Diastolic Blood Pressure	110.0
Pulse	72.0
Temperature	37.0

Table 4.3: Healthy norms chosen

The *pain score* is filled in with zero, assuming that when pain is not measured the patient has no pain, although we know that this is not always the case.

Filling in the mean

The missing values can also be filled in with the mean of the attribute. That is done for the following attributes:

- Length
- Weight
- BMI

When a patient is hospitalized for the first time in critical state and has to be taken into surgery immediately, it could be that the length and weight of the patient have not been filled in yet. In this case, the information can be filled in looking at the mean of the length of all the patients that have been hospitalized in the department.

Making the attribute binary

The following attributes were replaced by a binary attribute because they are not needed for all the patients all the time, which results in a lot of ”missing” values. These attributes now indicate if the measurement was taken or not:

- SpO2
- Respiration
- FiO2

Chapter 5

Method

In this chapter the classification algorithms which are used to train our model are explained. We use four different machine learning algorithms: classification and regression trees (CART), random forest (RF), naive bayes (NB), and support vector machine (SVM). These algorithms have been used in previous research, as previously stated in Chapter 3.

5.1 Classification and Regression Trees

The algorithm classification and regression trees (CART) is a nonparametric decision tree learning technique that generates classification and regression trees. It is a modern name for the classical decision tree algorithm.

Decision trees often perform well on imbalanced datasets and are frequently used to examine large patient databases [15]. The tree model that is presented by the decision tree algorithm is easy to interpret. Interpretability is important when explaining the machine learning model to someone that does not have a computer science background. In our case it is useful when presenting the model to the client. The algorithm has been used to predict if a patient had a chance of getting a heart disease [6].

A decision tree is a recursive, partition-based tree model that predicts the class $\hat{y}_i$ for each object x_i. [28] Let R denote the data space that contains the set of input objects D. An axis-parallel hyperplane is used to split the data space R into two, resulting half-spaces or regions, R_1 and R_2. It also leads to a partition of the input objects into D_1 and D_2. As a result, the computed hierarchy created by split decisions forms the decision tree model, with the leaf nodes labeled with the majority class among the objects in those regions. To classify a new test object, we have to recursively evaluate which half-space it belongs to until we reach a leaf node in the decision tree, at which point we predict its class as the label of the leaf.

The CART algorithm uses the Gini index function, which provides an impurity-based criterion that measures the differences between the probability distributions of the dependent variable's classes. We define the Gini index as: $I_g = \sum_{i=1}^{C} 1 - f_i^2$, where f_i is the frequency of label i at a node and C is the number of unique labels.

The downside of decision trees is that they often overfit. It has to know when to stop adding more detailed leaves. There are various approaches to avoid a decision tree from overfitting, such as pre-pruning and post-pruning. Pre-pruning will stop the tree from growing before it has classified the training set perfectly. Post-pruning will prune the tree after it has perfectly classified the training set.

Figure 5.1 shows a table with indicators of the weather which is made into a decision tree to illustrate when there are good circumstances to play golf. The blue nodes indicate the decision nodes and the orange ones indicate the leaf nodes that represent the classification.

Figure 5.1: Representation of a decision tree that is made based on data of the table [21]

5.2 Random Forest

Definition of random forests given by Breiman [4]: "A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \theta_k\}$ where the $\{\theta_k\}$ are independent identically distributed random vectors, and each tree casts a unit vote for the most popular class at input x."

In other words, a random forest consists of an ensemble of decision trees. The idea of ensemble learning methods is to select a collection, or ensemble, of hypotheses from the hypothesis space and combine their predictions [19]. For example, we might generate fifty different decision trees, and have them vote on the best classification for a new example.

The possible improvement in performance of the random forest algorithm over decision tree has a negative aspect; it loses the interpretability of the model. A set of many trees is harder to comprehend than a single tree. A random forest algorithm has been applied to distinguish patients with first-episode schizophrenia from healthy individuals [9].

5.3 Naive Bayes

The naive bayes algorithm owes its name to its naive approach of the Bayes theorem by assuming that all the attributes are conditionally independent (even when they are not). Bayes' theorem defines the probability of an occurrence, based on prior knowledge of conditions that might affect the occurrence. Naive bayes systems can work surprisingly well, even when the conditional independence assumption does not hold [19]. Although the algorithm is known as a decent classifier, it is known to be a bad estimator [13]. We therefore use the naive bayes machine learning model as a baseline. Our goal is to have a better model than the naive bayes model.

5.4 Support Vector Machine

Support Vector Machine (SVM) is a classification method based on maximum margin linear discriminants. The fundamental factor of SVMs is the hyperplane. In a binary classification task (like this research), the hyperplane is the geometrical separation between the two outputs. The algorithm aims to find the maximum margin hyperplane that maximizes the gap or margin between classes [28]. SVMs can accurately achieve nonlinear classification by utilizing kernels that can transform the hyperplane into a nonlinear input separator. The strength of the algorithm depends on the following parameters: the selection of kernel, the kernel's parameters, and the cost parameter C. SVM requires C for misclassification tolerance. SVMs are known to have good accuracy, but take much time to compute when using large datasets, and do not perform well on noisy data [2]. SVMs have been applied for classification in medical domains [15]. For instance, it has been used to predict stroke mortality at discharge [12]. The selected SVM that is used for this research is the linear support vector machine due to its fast computation time. It owes its name to its linear kernel.

Chapter 6

Experiments

After having pre-processed and cleaned the data, different machine learning models can be developed for predictive purpose. Since the response variable is binary, we use binary classification. Particularly, decision trees, random forest, naive bayes and support vector machines, as described in the previous chapter. To investigate which model suits the data best, $F1$-score is used to evaluate the models.

This chapter will present the evaluation techniques that are used, an exploratory data analysis and the results of comparing the different models. The best model is then being tuned and visualized.

6.1 Evaluation

To train and evaluate the performance of the machine learning methods, we use 10-fold cross-validation and compare the goodness of the fit of the methods. Several metrics can be used to evaluate the classification. In some cases, accuracy is a useful metric (percentage of right decisions), but in most cases, it is not suitable since classes are often unbalanced: high accuracy in one class might mean low accuracy in another class. We therefore use precision and recall and F-score as a metric.

6.1.1 10-fold cross-validation

The most straightforward evaluation technique that can be used to build a machine learning model is train/test split. This technique splits the data in a training and test set. The model will learn from the training set, make a prediction on the test set, and is evaluated by comparing the predictions against the expected results. This algorithm evaluation technique is very fast. It is optimal when using an extensive dataset (millions of records) that has been split in a way that is representative of the underlying problem [5]. The downside of using train/test split is that that a single performance indicator does not present information about the actual trustworthiness of the results [24]. To have a more reliable performance indicator a confidence interval can

be calculated. However, confidence intervals can only be computed when using multiple measurements; this can be done by using k-fold cross-validation. This approach is often used as a performance indicator [24]. The dataset is divided into k equal subsets; each split is called a fold. The algorithm is trained on $k-1$ folds, with one fold kept out that is used as the test set. This is repeated k times, which will result in k different performance scores that can then be summarized using a mean and a standard deviation. The selected k has to be large enough to split the data into representative samples, but should also allow enough repetitions to represent a reliable evaluation method. Values 3, 5, and 10 are the most commonly used [5]. In this research, a k value of 10 is used. Having k test results instead of one can give more information about the trustworthiness of the model. The results are however not entirely independent since the k test sets overlap within the k-folds as well. In our case data of the same hospitalization could be divided into different k-folds. To avoid this, we group objects of the same hospitalization in the same fold.

6.1.2 F_1-score

In machine learning, a confusion matrix is an interpretable table that depicts the performance of a machine learning algorithm. The matrix owes its name to the fact that it gives insight in how the algorithm "confuses" multiple classes. In this research, there are only two classes: the positive class (discharged within 48 hours) and the negative class (will not be discharged within 48 hours).

	True Class	
Predicted Class	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Table 6.1: Confusion matrix for two classes

The entries that are given in the confusion matrix shown in Table 6.1, are given the following definitions:

- True Positive (TP): The number if objects that the classifier correctly predicts as positive.
- False Positive (FP): The number of objects the classifier predicts to be positive, which in fact belong to the negative class.
- False Negative (FN): The number of objects the classifier predicts to be in the negative class, which in fact belong to the positive class.
- True negative (TN): The number of points that the classifier correctly predicts as negative.

We define precision as a proportion of correct assigned labels and recall as the proportion of true labels that was assigned. The F_1-score is the harmonic mean of precision and recall:

$$Precsion = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F_1 = 2 * \frac{precision * recall}{precision + recall}$$

6.2 Exploratory Data Analysis

Before starting the evaluations of the models it is beneficial to have more knowledge about the data that is used for this research. This is done by giving different visualizations of certain attributes.

From the first figure, Figure 6.1, is clear that most patients that have been taken care of at these particular wards are elderly. In Figure 6.2 it is shown that there is a small increase in the average length of stay compared to the age. Figure 6.3 presents the average length of stay per specialism.

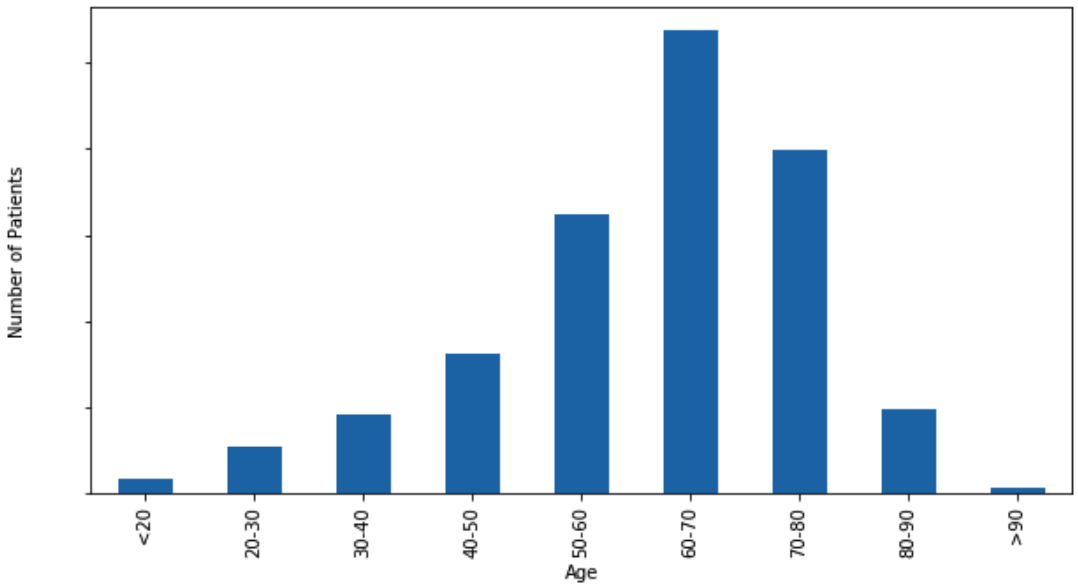

Figure 6.1: The number of patients per age group

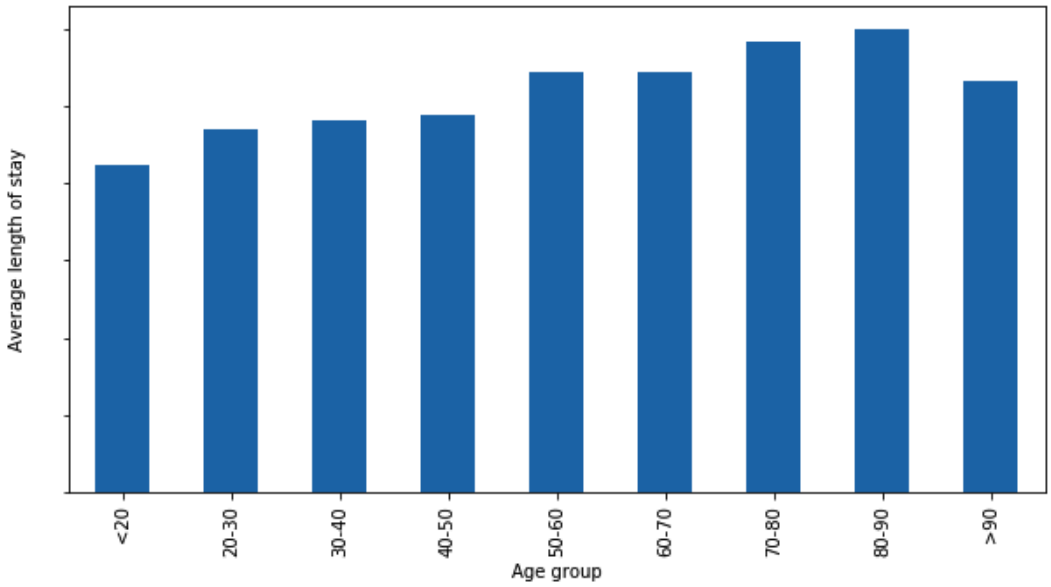

Figure 6.2: The average length of stay per age group

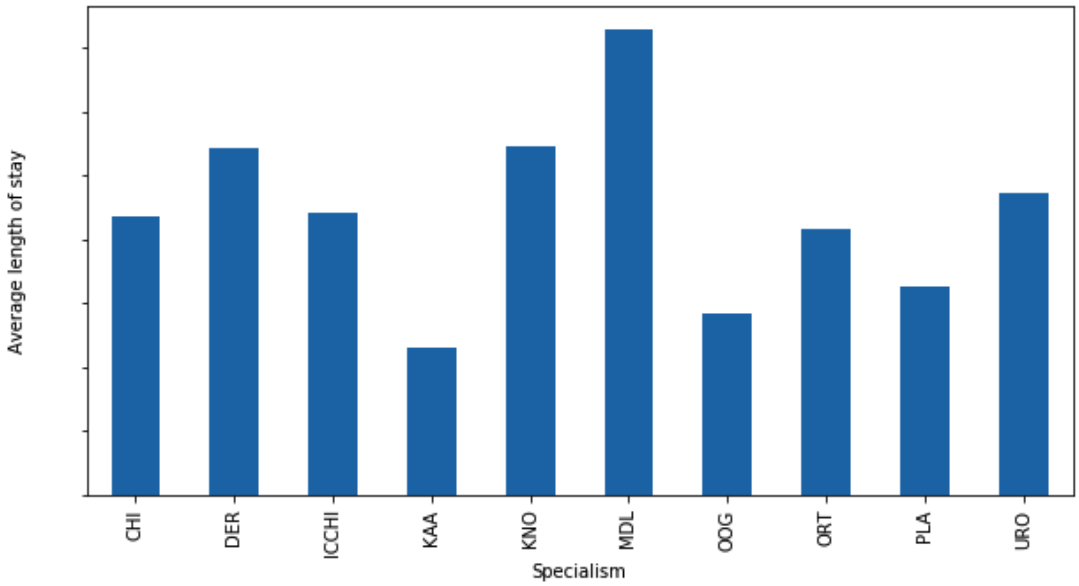

Figure 6.3: The average length of stay per specialism

To give insight in how many elective and urgent patients are being hospitalized throughout the year, a visualization has been made. In Figure 6.4 the different colors represent different years and the different kind of patients (urgent or elective). In the tag "Spoed_0_14", zero indicates an elective patient and 14 indicates the year 2014. This figure makes clear that there is a slight resemblance between the number of elective patients each year. It also shows that in 2017 a lot urgent patient came in between June and July.

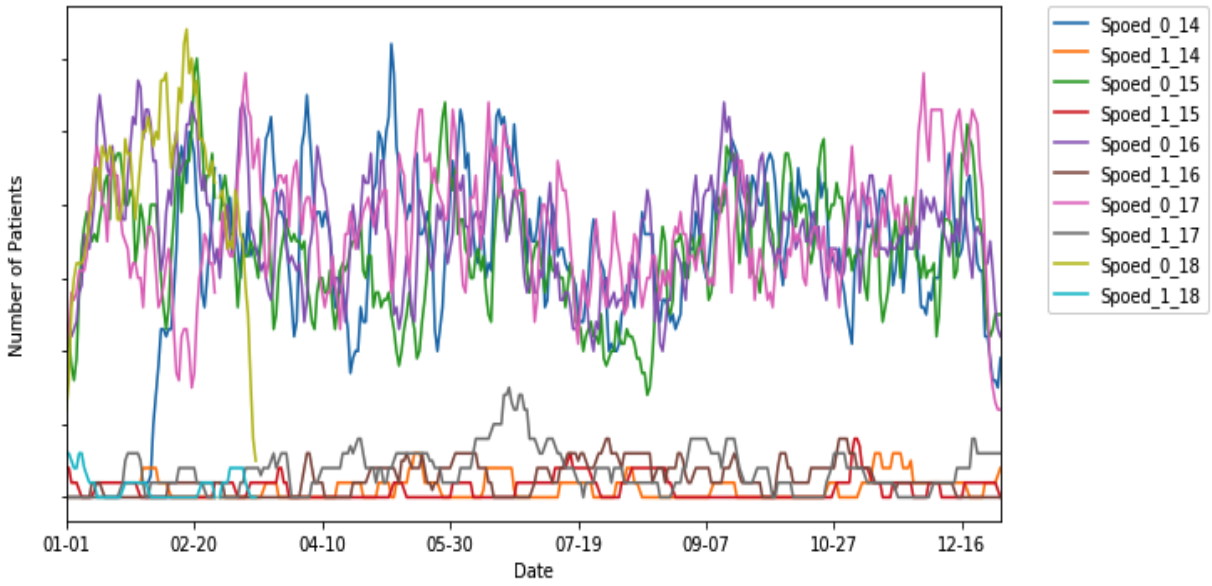

Figure 6.4: The average length of stay per age group. In the tag "Spoed_0_14", zero indicates an elective patient and 14 indicates the year 2014.

6.3 Comparing Machine Learning Models

Machine learning algorithms can be compared by looking at the performance measures. In this research, we also want a machine learning model that is easy to interpret and does not take much time to compute. Interpretability is essential when explaining our model to the client, a machine model that is easy to visualize is preferred. The aim is to make a model that can be used every moment of the day as a helpful tool for clinicians. We therefore want a model that does not take much time to compute.

6.3.1 Results

We compared the following machine learning models: classification and regression tree (CART), random forest (RF), naive bayes (NB) and support vector machine (SVM). Table 6.2 presents the results of the confusion matrices computed for every machine learning algorithm. The results of every fold are summed up. Computing the average score does not provide additional information from the cumulative matrix and could be biased if the folds are not all the same size. The presented scores have been normalized.

Method	TP	FP	FN	TN	Recall	Precision	F1-Score
CART	**0.53**	0.23	**0.47**	0.77	**0.53**	0.43	**0.47**
RF	0.35	**0.09**	0.65	**0.91**	0.35	**0.58**	0.44
NB	0.29	0.15	0.71	0.85	0.29	0.39	0.33
SVM	0.30	0.30	0.70	0.70	0.30	0.25	0.27

Table 6.2: Performance scores of four classification models by looking at confusion matrix and recall, precision and F1-score. The bold numbers indicate the best scores.

The most important measure to look at is the value of False Negative (FN). These are the objects that the model predicts to be in the negative class which in fact belongs to the positive class. This research aims to create a model that predicts if a patient can be discharged within 48 hours more accurately than the clinicians. It is therefore essential to predict the objects that belong to the positive class correctly. The model aims to have an FN score near 0 and a TP score near 1.

6.3.2 Findings

From the results, it is clear that CART and RF are better models than NB. They both have better values for FN, TN, and F1-score. These scores are satisfying as we aimed to create a model that would perform better than NB. SVM, however, does not perform better, this was not expected. It could be explained because we used a linear kernel which computes fast but has resulted in a not well-performing model. Comparing CART and RF, we can see that RF is more accurate in predicting the negative class and CART predicts the positive class more accurately. We aim to create a model that predicts the positive class the best, we therefore have chosen CART as the best model, based on the performance scores.

When considering the time needed to compute (not shown), we compared CART and RF. CART is faster than RF, an explanation for this is that RF has to compute more decision trees, thus requiring longer computation time. As far as visualization is concerned, in Chapter 5 it is stated that CART is the easiest to visualize. In conclusion, CART is the best machine learning model for the problem of this research.

6.4 Hyperparameter Optimization

In machine learning, hyperparameter optimization is the problem of selecting the correct setting per parameter for the machine learning algorithm [24]. The aim is to tune the algorithm so that it can optimally solve the machine learning problem. The following parameters have been selected with the corresponding options of settings.

- Criterion = ['gini', 'entropy']
- Splitter = ['random', 'best']
- MaxDepth = [15,20,25]
- MinSampleSplit = [2,3,4]
- MinSampleLeaf = [1,2,3]

The developers of sci-kit-learn have described the parameters as following [14]:

Criterion: With this parameter the function to measure the quality of a split can be selected. The supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. More information about the Gini impurity can be found in Chapter 5.
Splitter: The strategy used to choose the split at each node. The supported strategies are best to choose the best split and random to choose the best random split.
MaxDepth: With this parameter the maximum depth of a tree can be chosen. If None, then the tree will be expanded until all leaves contin less than MinSampleSplit or until all leaves are pure, this however causes the tree to overfit. To avoid this we pre-prune the tree by stopping it from growing at a certain depth.
MinSampleSplit: The minimum number of samples required to split an internal node.
MinSampleLeaf: The minimum number of samples required to be at a leaf node.

The default setting for the selected parameters is [gini, best, None, 2, 1]. Computing all the different settings resulted in 182 different models with their corresponding F1-scores. The best performing model had the parameters: ['gini', 'best', 15, 4, 3]. From Table 6.3 it is clear that predicting the positive class has not improved. However, predictive accuracy of the negative class has slightly improved.

Method	TP	FP	FN	TN	Recall	Precision	F1-Score
Default	0.53	0.23	0.47	0.77	0.53	0.43	0.47
Best	0.53	0.19	0.47	0.81	0.53	0.47	0.50

Table 6.3: Performance scores of two CART models presenting measures of confusion matrix and recall, precision and F1-score

Figure 6.5 shows a comparison of F1-scores of the two models. As shown the Default model has lower average F1-score than the Best model. The box plot of the Best model is comparatively tall, which indicates that the model computes more various F1-scores than the Default model.

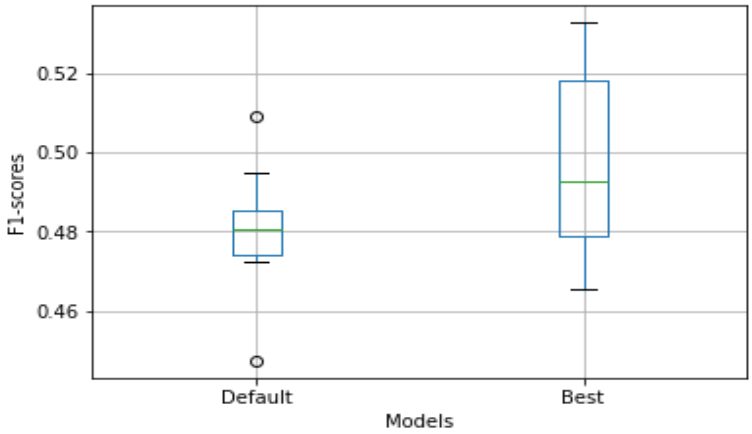

Figure 6.5: A comparison of F1-scores of the two models

6.5 Visualization

One of the great features of CART models is that they are easy to visualize. By visualizing the model, we get insight in how the model works. This is beneficial when explaining the machine learning model to the client. The tree computed for this research will either classify an object as true (being discharged within 48 hours) or false (not being discharged within 48 hours).

Figure 6.6 shows the computed CART model until depth 4 for an arbitrary chosen fold. The 'value' row in each node indicates how many objects that are sorted into that node are classified in each of the two categories. The colors of the tree indicate how much of the objects are classified in each of the two categories. The more orange means that there are more object classified as false and the more blue nodes indicate that more objects are classified as true. As shown in the figure the first criteria to split the data on is the current length of stay (CurrLOS). Patients in our dataset stay longer than three days, so it is expected that in the first two timeslots (first 24 hours) the patient is classified as false. The first node thus splits the data into two samples, in one sample objects have a CurrLOS smaller than 2.5 days and one sample that has objects that have a CurrLOS larger than 2.5 days.

The Gini index gives insight in how balanced the data of a node is. A value of zero indicates perfect equality, implying that the data only consists of one class.

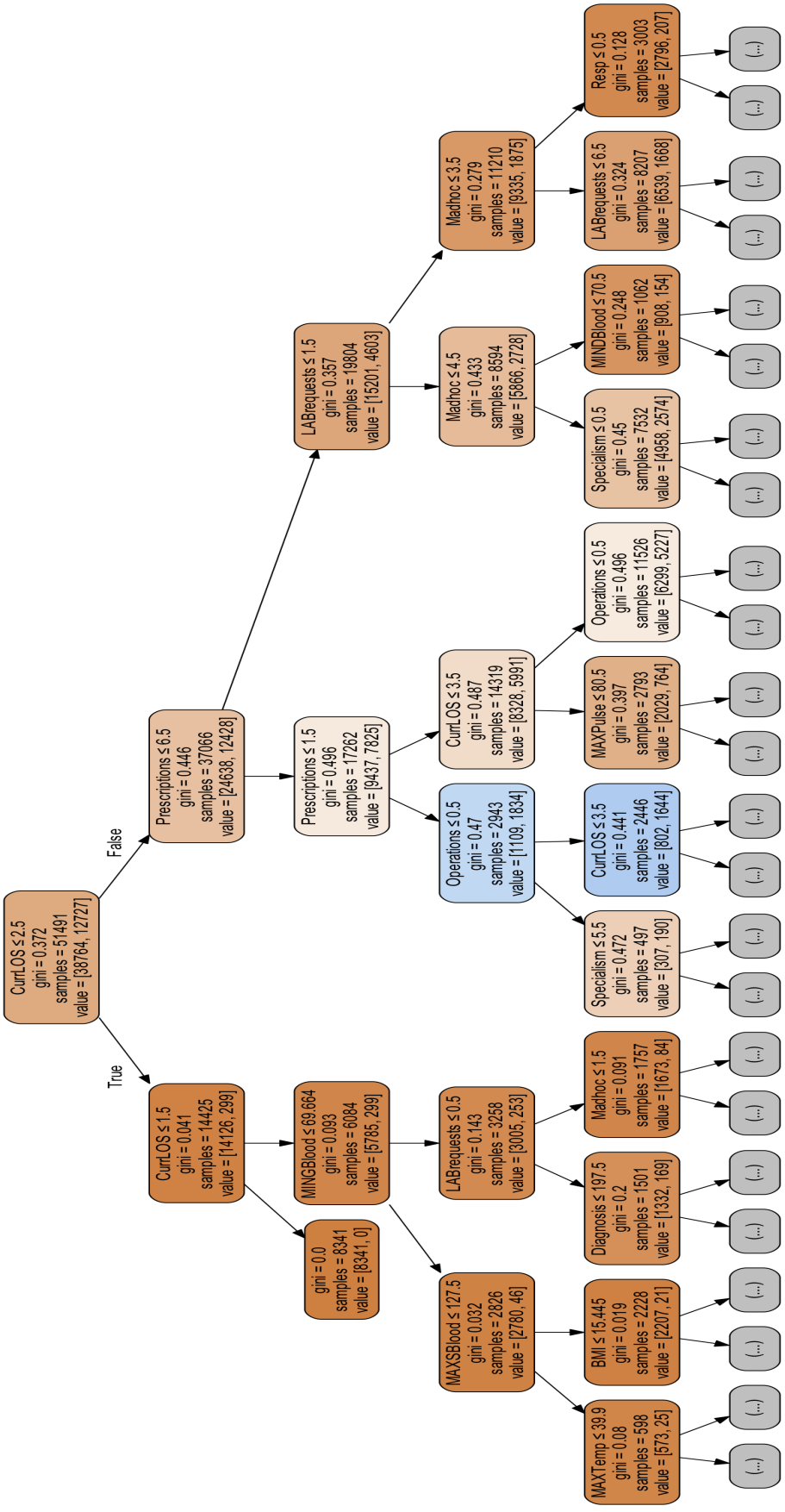

Figure 6.6: CART visualized till depth 4

In Figures 6.7 till 6.9 some attributes are displayed with the number of times a specific threshold was used for splitting the data. Figure 6.7 shows that the attribute lab requests is used several times with various values. It is not shown on which depth of the tree the split is made. In Figure 6.8 it is shown that the attribute maximum pain score is used several times but not on all the possible values a split is made. It interesting to see in Figure 6.9 that the split values are very specific. It suggests that there is a difference made somewhere in the tree if an object has a maximum temperature above 37.25 or under that specific value.

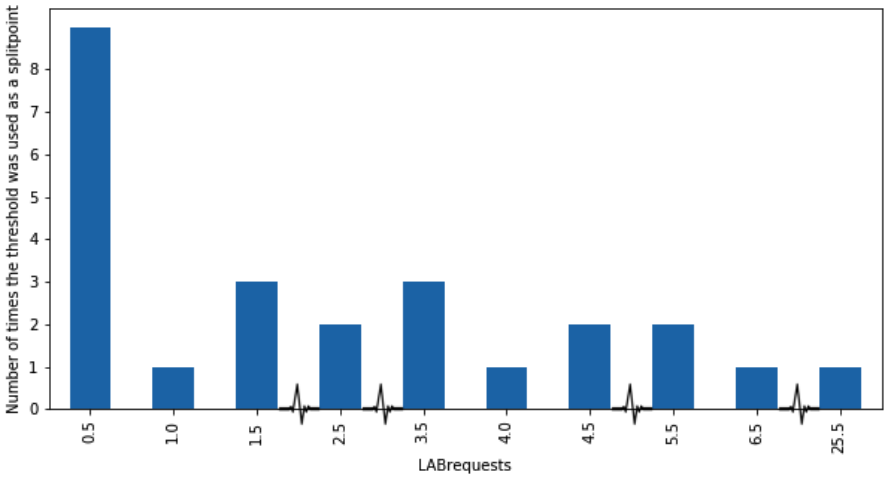

Figure 6.7: The number of times a specific threshold was used for splitting the data. Total of lab requests since hospitalized

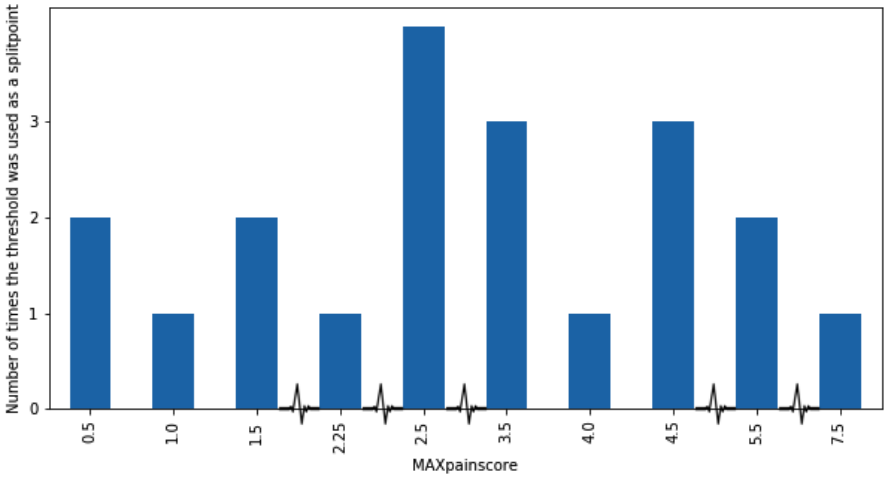

Figure 6.8: The number of times a specific threshold was used for splitting the data. Maximum Pain Score measured in the past 24 hours

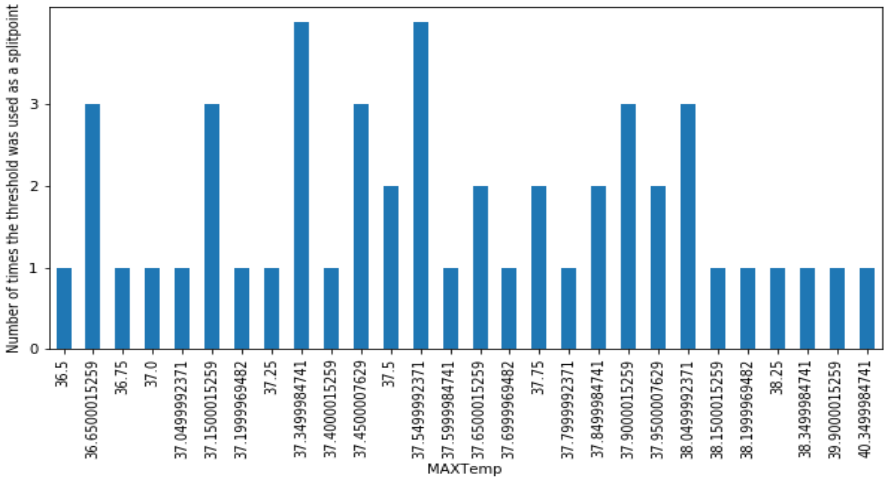

Figure 6.9: The number of times a specific threshold was used for splitting the data. Maximum Temperature measured in the past 24 hours

Table 6.4 presents the first ten most predictive attributes, starting with the most important one. The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature [14]. It is also known as the Gini importance. An interesting observation is that number of prescriptions plays a influential role when determining whether a patient is being discharged within 48 hours.

Attribute	**Gini Importance**
Current Length of Stay	0.22784617
Number of prescriptions	0.10872678
Diagnosis	0.0431056
Number of Lab requests	0.03125004
Date	0.03007951
Age	0.02816389
Maximum Pulse	0.02651
Medication taken adhoc	0.02632737
Appointment type V	0.02600289
Hospitalization Number	0.02439653

Table 6.4: Feature importance ranking for the tree that has been visualized for an arbitrary chosen fold.

Chapter 7

Discussion

If a classification model has an accuracy above 50%, it is suggested that it performs better than random. For recall, precision, and F1-score a score above 0.5 indicates that a classification model performs better than random. In the previous chapter (Table 6.3), it is shown that the best performing model in this research has an F1-score of 0.5 which suggest that our model does not perform better than random. The aim was to have a model with an FP (False Negative) score near 0 and a TP (True Positive) score near 1. Both these scores of the best model are around 0.5 implying that our model does not perform as well as we wanted to.

The parameters that have been tuned are only a sample of all the possible parameters of the model, and only a few settings have been tested. For future work, this can be extended by examining more parameters and more settings. Not only can this be done for the best performing model but also for the other classification models.

In the previous chapter, the feature importance ranking is shown. This could be used to improve the classification model by selecting the most important features to construct a better model. It could reduce overfitting of the classification and regression tree. In Table 6.4, the hospitalization number is one of the most predictive features. This was not expected because we constructed the folds in a way that the hospitalization number would only occur in one fold.

The data that was made available was structured data from Electronic Health Records (EHRs). The way we filtered the data and imputed the missing values can be done differently. It is challenging to determine the perfect way to present an attribute without losing much information or presenting unreliable data.

It should be noted that we have been focusing on the patients that have been discharged, neglecting the fact that the patient that has been discharged could also have been passed away. This means that we predict if a patient dead or alive will leave the hospital within 48 hours. For a more reliable model the patients that passed away could be excluded from the data or an extra category can be introduced. An extra category would result in three different categories. The patient than could be classified as not leaving the hospital, being discharged, or will pass away.

We have chosen to predict 48 hours in advance because this would be an improvement on the current situation. The ideal situation would be to create a model that can predict even earlier. To do this however the model for 48 hours has to be perfectly accurate.

This research makes clear that working with medical data and finding the right model is a long and challenging process. To be able to create a model that is more accurate than the discharge estimate of clinicians, more medical data has to be used, and it is beneficial to have a medical background to make working with the data less complicated.

Chapter 8

Conclusions

This thesis aims to create a predictive model that is more accurate than the discharge estimate of clinicians. Based on structured data from Electronic Health Records (EHRs), we created a dataset consisting of objects that could be classified as true, meaning the patient being discharged within 48 hours or false, indicating that the patient will stay in the hospital the next 48 hours.

Since the response variable is binary, working with classification techniques was preferred. In particularly we computed four different machine learning classification algorithms: classification and regression tree (CART), random forest (RF), naive bayes (NB), and support vector machine (SVM). Based on the F1-score to evaluate the classification, CART was the best performing model. CART was not only the best performing but also had a fast computation time and is the easiest model to interpret. To make the model even better, we tuned it. This resulted in a slightly better F1-score. To give insight in how the model works we visualized the decision tree.

The objective of this research is to create a classification model that can predict whether a patient will be discharged within 48 hours accurately. The F1-score of the best performing model of this research is 0.5, which indicates that it does not perform better than random. We can conclude that we did not achieve our goal. However, there are still enough things that have not yet been examined which leads to enough room to expand the research further.

We have experienced how challenging it is when working with medical data. It is difficult to determine which attributes to use and how to filter the data without losing information. Missing values have been filled in in different ways. We imputed missing values by filling in the mean, the healthy norm or by making the attribute binary.

8.1 Future Work

The goal of the model was to predict the patients discharge more accurate. However, this goal has not been achieved. We therefore recommend the following approaches for future work.

- **Text mining** - This research only focused on the structured data of the patient. The Electronic Health Record (EHR) also consists of textual data. Mining text data could result in gaining more information about the patient. By doing this it could be interesting to show how important the text data is.
- **More attributes** - We only took a sample of the available attributes for our research. There are a lot more to include. For instance, we only looked at how many medications the patient had but not which types of medicine the patient took.
- **Parameter Tuning** - CART is the only model that has been tuned in this research, for future work, tuning the other models is recommended.
- **Logistics** - The date of discharge is influenced by the destination of the patient. Most of the patients are elderly that cannot care for themselves and have to be placed in a care home. By combining the information of the logistics department it could give an insight on how the discharge date is dependent on how busy the logistics apartment is.
- **Other models** - The information that has been filtered can also be used for other data driven research. It could be possible to group patients with certain characteristics to give an insight on their medical data.

8.2 Recommendations

While working with data from EHRs we experienced the unfiltered information about the patient that is stored in the database. This takes a lot of time when working with the data for data-driven research. We recommend using an EHR system that constrains the possible input. This can be done by using sliders, for instance for pain score where the possible value is between zero and ten. The system could give an error when the value that has been filled in cannot be true, for instance, a recorded minus value for pain cannot be true. This would make working with data of a patient easier.

Bibliography

[1] Agresti Alan and Finlay Barbara. *Statistical methods for the social sciences*. Pearson Education Limited, 2014.

[2] Laura Auria and Rouslan A Moro. Support vector machines (svm) as a technique for solvency analysis. *DIW Berlin Discussion Paper No. 811*, 2008.

[3] Sean Barnes, Eric Hamrock, Matthew Toerper, Sauleh Siddiqui, and Scott Levin. Real-time prediction of inpatient length of stay for discharge prioritization. *Journal of the American Medical Informatics Association*, 23(e1):e2–e10, 2015.

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

[5] Jason Brownlee. Machine learning mastery with python. *Machine Learning Mastery Pty Ltd*, pages 100–120, 2016.

[6] Vikas Chaurasia and Saurabh Pal. Early prediction of heart diseases using data mining techniques. *Caribbean Journal of Science and Technology*, 1:208–217, 2013.

[7] Michael E Chernew and Joseph P Newhouse. Health care spending growth. In *Handbook of health economics*, volume 2, pages 1–43. Elsevier, 2011.

[8] Mao-Te Chuang, Ya-han Hu, and Chia-Lun Lo. Predicting the prolonged length of stay of general surgery patients: a supervised learning approach. *International Transactions in Operational Research*, 25(1):75–90, 2018.

[9] Yi Deng, Karen SY Hung, Simon SY Lui, William WH Chui, Joe CW Lee, Yi Wang, Zhi Li, Henry KF Mak, Pak C Sham, Raymond CK Chan, et al. Tractography-based classification in distinguishing patients with first-episode schizophrenia from healthy individuals. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 2018.

[10] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[11] Jessica Hegg. Understanding blood pressure. `https://www.vivehealth.com/blogs/resources/understanding-blood-pressure`, 2018. [Online; accessed 22-June-2018].

[12] King Chung Ho, William Speier, Suzie El-Saden, David S Liebeskind, Jeffery L Saver, Alex AT Bui, and Corey W Arnold. Predicting discharge mortality after acute ischemic stroke using balanced data. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1787. American Medical Informatics Association, 2014.

[13] Sci kit learn developers. 1.9 naive bayes. `http://scikit-learn.org/stable/modules/naive_bayes.html`, 2017. [Online; accessed 14-August-2018].

[14] Sci kit learn developers. sklearn.tree.decisiontreeclassifier. `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`, 2017. [Online; accessed 18-August-2018].

[15] Geert Meyfroidt, Fabian Güiza, Jan Ramon, and Maurice Bruynooghe. Machine learning techniques to examine large patient databases. *Best Practice & Research Clinical Anaesthesiology*, 23(1):127–143, 2009.

[16] April Morton, Eman Marzban, Georgios Giannoulis, Ayush Patel, Rajender Aparasu, and Ioannis A Kakadiaris. A comparison of supervised machine learning techniques for predicting short-term in-hospital length of stay among diabetic patients. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 428–431. IEEE, 2014.

[17] Parag C Pendharkar and Hitesh Khurana. Machine learning techniques for predicting hospital length of stay in pennsylvania federal and specialty hospitals. *International Journal of Computer Science & Applications*, 11(3), 2014.

[18] OECD. Publishing. *Health at a Glance 2013*. OECD publishing, 2013.

[19] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[20] Haya Salah. *Predicting Inpatient Length of Stay in Western New York Health Service Area Using Machine Learning Algorithms*. PhD thesis, State University of New York at Binghamton, 2017.

[21] Dr. Saed Sayad. Decision tree - classification. `http://www.saedsayad.com/decision_tree.htm`, 2010. [Online; accessed 24-August-2018].

[22] Lior Turgeman, Jerrold H May, and Roberta Sciulli. Insights from a machine learning model for predicting the hospital length of stay (los) at the time of admission. *Expert Systems with Applications*, 78:376–385, 2017.

[23] Aart R van de Vijsel, Richard Heijink, and Maarten Schipper. Has variation in length of stay in acute hospitals decreased? analysing trends in the variation in los between and within dutch hospitals. *BMC health services research*, 15(1):438, 2015.

[24] Wil MP Van der Aalst. *Process mining: data science in action*. Springer, 2016.

[25] Laurien Verheijen. Predicting patient discharge at the intensive care unit, 2016. LUMC.

[26] Saskia Wassenaar and Sjors Molenaar. Zorg in regio slibt dicht: ziekenhuizen kunnen ouderen niet kwijt. `https://www.gelderlander.nl/regio/`

zorg-in-regio-slibt-dicht-ziekenhuizen-kunnen-ouderen-niet-kwijt~a9ee9e39/, 2017. [Online; accessed 28-June-2018].

[27] Alexey Yakovlev, Oleg Metsker, Sergey Kovalchuk, and Ekaterina Bologova. Prediction of in-hospital mortality and length of stay in acute coronary syndrome patients using machine-learning methods. *Journal of the American College of Cardiology*, 71(11):A242, 2018.

[28] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.