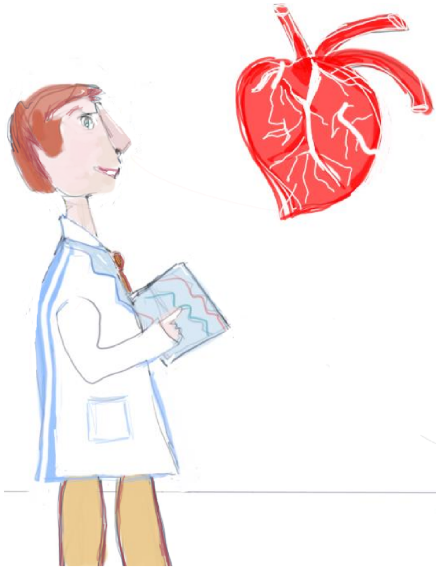
Universiteit Leiden
The Netherlands

SJD
Sant Joan de Déu
Barcelona · Hospital

Master Computer Science

Human-guided Rule List Learning for Length of Stay Classification.

Name:	Pol Mor-Puigventós
Student ID:	s3228673
Date:	July 28, 2023
Specialisation:	Artificial Intelligence
1st supervisor:	Matthijs van Leeuwen
2nd supervisor:	Francesco Bariatti
Daily advisor:	Ioanna Papagianni



Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Preface

Dr Joan Sánchez de Toledo is a hard worker and persuasive individual. As for myself, Pol Mor Puigventós, I am a very organized person and like to think in advance, plan my journeys and be in control. However, I thrive on challenges and hold a belief in destiny.

Almost two years ago, Joan invited me to visit Sant Joan de Déu, where he serves as the head of the Heart Area and the Pediatric Cardiology department. While his expertise and research primarily revolve around advancements in cardiology procedures, he actively encourages his colleagues to embrace the presence of data science experts and their influence on their work. During my visit, I was shown the *eCare*, an ensemble of teams, rooms and resources that focus on involving all members of the hospital in its mission to centralize and standardize data collection processes for future enhanced patient care.

Last Christmas, Joan once again extended his invitation, allowing me to meet the data scientists who work alongside clinicians. Patricia Garcia Cañadilla, Arnau Valls Esteve, and Adriana Modrego Muñoz explained the data they managed to gather and expressed their wish to tackle some of their challenges with machine learning. One of them is predicting the Length of Stay. This is, the duration of time that a patient spends admitted and receiving medical care at the hospital. They offered me the possibility of utilizing their data for this purpose, which potentially could become the focus of my master's thesis.

At that moment I already had submitted a proposal for a different project related to the exploration theory in the domain of Reinforcement Learning for my thesis. However, as mentioned earlier, Joan's persuasive nature and my inclination towards challenges led me to embrace the opportunity to collaborate on this endeavour. So here we are.

Medicine now compared to medicine in the future. From Rajkomar et al. (2019).

A 49-year-old patient notices a painless rash on his shoulder but does not seek care. Months later, his wife asks him to see a doctor, who diagnoses a seborrheic keratosis. Later, when the patient undergoes a screening colonoscopy, a nurse notices a dark macule on his shoulder and advises him to have it evaluated. One month later, the patient sees a dermatologist, who obtains a biopsy specimen of the lesion. The findings reveal a noncancerous pigmented lesion. Still concerned, the dermatologist requests a second reading of the biopsy specimen, and invasive melanoma is diagnosed. An oncologist initiates treatment with systemic chemotherapy. A physician friend asks the patient why he is not receiving immunotherapy.

A 49-year-old patient takes a picture of a rash on his shoulder with a smartphone app that recommends an immediate appointment with a dermatologist. His insurance company automatically approves the direct referral, and the app schedules an appointment with an experienced nearby dermatologist in 2 days. This appointment is automatically cross-checked with the patient's personal calendar. The dermatologist performs a biopsy of the lesion, and a pathologist reviews the computer-assisted diagnosis of stage I melanoma, which is then excised by the dermatologist.

Abstract

Our objective is to introduce an algorithm for predicting the Length of Stay (LoS) at Sant Joan de Déu hospital in Barcelona. Predicting the LoS in a hospital refers to forecasting the total duration of a patient's hospitalization after admission at the hospital. While numerous studies have tackled this problem, the prevailing solutions predominantly rely on non-interpretable algorithms, and often yielding low accuracies in their predictions.

In critical domains like healthcare, where interpretability and trust are essential, non-interpretable algorithms like neural networks may not be suitable for addressing the problems at hand. However, interpretable models, such as rule-based models, provide clear explanations for their predictions, making them more suitable and trustworthy in such scenarios.

In our study, we select S-CLASSY, a rule-based model, which uses beam search and human guidance in its learning process. Collaborating closely with domain experts, we create a human-guided pipeline to train the S-CLASSY model for LoS prediction, prioritizing performance and interpretability.

We compare its performance with other models, both interpretable and non-interpretable, and explore the effect of human guidance through the *preferred variables* that drive the initial patterns search in the data. Additionally, we assess the model's interpretability using existing and new quantitative metrics, and qualitative feedback from the experts.

Therefore, we achieve an understandable model, considered interpretable and transparent by the clinicians to predict the LoS. Nevertheless, medical relevance in the generated rules is not achieved, which is essential for gaining medical professionals' acceptance. To address this limitation, we suggest future steps for S-CLASSY to generate medically relevant rules, aiming to enhance the model's practicality, making it usable and trustworthy for medical professionals.

CONTENTS

1 Introduction **1**
- 1.1 Predicting the Length of Stay . . . 3
- 1.2 Research questions . . . 4
- 1.3 Approach and contributions . . . 4
- 1.4 Outline of this thesis . . . 6

2 Related Work **7**
- 2.1 Length of Stay . . . 8
- 2.2 Interpretable machine learning . . . 10

3 Rule Learning **11**
- 3.1 MDL-based rule learning . . . 12
  - 3.1.1 Rule learning . . . 12
  - 3.1.2 CLASSY MDL-based learning . . . 13
- 3.2 Human-guided rule learning . . . 14

4 Data Preparation and Means **16**
- 4.1 Data . . . 17
  - 4.1.1 Data selection . . . 18
  - 4.1.2 Variables . . . 20
  - 4.1.3 Exploratory Data Analysis . . . 22
  - 4.1.4 Pre-processing . . . 23
- 4.2 The clinical requirements . . . 29
- 4.3 The machine learning problem . . . 30
  - 4.3.1 Expert's variables selection . . . 30
  - 4.3.2 Final data for experimentation . . . 31

5 Experiments **34**
- 5.1 Experiment setup . . . 35
  - 5.1.1 Models for comparison . . . 35
  - 5.1.2 Configurations . . . 36

5.1.3 Metrics . . . 36
5.2 Performance . . . 38
5.2.1 Scores . . . 38
5.2.2 Confusion analysis . . . 40
5.3 Human guidance . . . 43
5.3.1 Scores . . . 43
5.3.2 Preferred variables frequency . . . 44
5.4 Interpretability . . . 45
5.5 Expert's feedback . . . 47

6 Discussion **49**
6.1 Predicting the Length of Stay . . . 50
6.2 S-CLASSY . . . 51
6.3 New line of research . . . 52

7 Conclusion **54**
7.1 Conclusion . . . 55
7.2 Limitations and future work . . . 55
7.3 Acknowledgements . . . 58

A Appendix **63**
A.1 Experimentation with the ward data . . . 64
A.2 Performance evaluation . . . 65
A.2.1 S-CLASSY, RF, MLP, and GB . . . 65
A.2.2 S-CLASSY, CLASSY, and decision tree . . . 68
A.3 Interpretability evaluation . . . 71
A.4 Human guidance evaluation . . . 74
A.4.1 AUROC and AUPRC curves . . . 75
A.5 Rule list examples . . . 76
A.6 Variable importances . . . 78
A.7 Decision tree plot . . . 81

CHAPTER

1

INTRODUCTION

Research has played a pivotal role in the progress of humankind, with each new discovery, we have gained a sense of security and confidence. In this era of ever-advancing technology, the domain of healthcare, particularly in hospitals, stands as a complex ecosystem where the well-being of patients and the efficient allocation of resources are of utmost importance. In this context, the integration of machine learning (ML) algorithms has become increasingly prevalent, offering powerful tools for analyzing and predicting patient outcomes.

Nevertheless, based on the criterion of interpretability, we can categorize machine learning algorithms into two distinct types. Non-interpretable models, such as neural networks, and interpretable models, such as decision trees. Existing literature addresses the issue of explaining neural networks, notably XAI by Gunning et al. (2019), but also many more in recent years (Xu et al., 2019) as the topic gathered much more attention, both in the research community and industry. However, the networks themselves do not offer explanations about how they arrive at a particular prediction, necessitating the development of external techniques to gain insights. This is why our research endeavours to explore alternative approaches that prioritize interpretability as a starting point.

In particular, we focus on the development of interpretable ML algorithms (Doshi-Velez & Kim, 2017), which have gained substantial relevance, particularly in domains where clarity is crucial due to the critical nature of the decision-making process, such as healthcare. Ensuring that our algorithm is interpretable is of utmost importance as we seek to address a healthcare problem where transparency is a must. We look for allowing healthcare practitioners to scrutinize and comprehend the factors influencing the outcomes of the predictor (Ahmad et al., 2018). Our motivation stems from the understanding that the collaborative involvement of experts can greatly enhance the development and effectiveness of our algorithm.

Consequently, we place emphasis on incorporating human guidance to steer the algorithm's search. Specifically, we aim for medical doctors to drive the search by using their specific criteria. This approach aligns with the concept of *informed machine learning* (von Rueden et al., 2021), where human feedback plays a crucial role in guiding the learning process. We can draw parallels to examples of this human influence in reinforcement learning algorithms, where extensive literature already exists on incorporating human input to improve the learning outcomes (Zhang et al., 2021).

For this thesis, we use the S-CLASSY algorithm introduced by Papagianni & van Leeuwen (2023) as it aligns well with the project's requirements. S-CLASSY is a rule-based algorithm used for classification tasks, focusing on discovering meaningful patterns in the data that lead to accurate predictions. It is an extension of the original CLASSY (Proença & van Leeuwen, 2020) and SSD++ (Proença et al., 2022) algorithms.
Rule learning is a field within interpretable machine learning that involves creating models like decision trees, rule sets, and rule lists.
CLASSY, can produce interpretable rule lists, making it suitable for applications where transparency and understandability are crucial. This model employs the minimum description length

(MDL) principle to select the best rule among competing rules, effectively compressing the data.
Moreover, S-CLASSY incorporates human guidance in the process. By preparing a list of preferred variables, the algorithm is directed to search for patterns in the data according to the input from human experts. This human guidance approach enhances S-CLASSY's interpretability, making it even more usable for understanding the model's predictions.

Our research primarily focuses on the prediction of the Length of Stay (LoS) for patients admitted to a hospital, with a specific emphasis on collaborating with Sant Joan de Déu's (SJD) pediatric hospital in Barcelona. The main objective of our study is to train an algorithm on the hospital's data that can accurately forecast the duration of a patient's stay until discharge, whether in the Intensive Care Unit (ICU) or the ward, following heart surgery for postoperative care. Together, we conduct interviews with the healthcare staff and we gather valuable insights into the specific needs and requirements to tackle this task.

1.1 Predicting the Length of Stay

The Length of Stay (LoS) for patients in a hospital refers to the number of days they spend during a single admission event (Huntley et al., 1998). This admission event is the time that one person stays in a hospital from the moment of their admission until their discharge. Predicting the LoS for patients is often challenging and usually done retrospectively. LoS can vary across different levels, such as ICU, step-down units, and general floor areas. It can also be evaluated separately for each area or combined to provide a total LoS. Apart from being a key indicator of hospital resource consumption, LoS offers valuable insights into the patient flow within care units and environments, playing a crucial role in evaluating the operational functions of different healthcare systems. It is often considered a metric to assess resource utilization, cost, and the severity of illness.

The LoS is a complex concept influenced by various factors, sometimes competing with each other. For example, the patient's genetics, the extent or seriousness of trauma or disease, the overall medical or surgical treatment received, the quality of care provided (including the availability of resources), environmental conditions outside the hospital (such as sanitation), and the accessibility of intermediary and long-term care facilities. LoS serves as a measurable outcome that directly correlates with hospital costs and the economic consequences of trauma and disease (Stone et al., 2022).

Addressing the Length of Stay issue provides an opportunity to hit two targets with one shot. Firstly, by discharging patients at the appropriate time, unnecessary healthcare costs can be curtailed, leading to improved resource allocation and financial efficiency. Secondly, a targeted approach to the LoS minimizes the risk of patients contracting secondary infections within the hospital environment, ultimately enhancing patient safety and well-being.

At Sant Joan de Déu, being a pediatric hospital, an additional factor comes into play. Due to the young age of the patients, they are generally unable to communicate their symptoms or express their desire for discharge if they feel in good shape. As a result, predicting the Length of Stay becomes crucial for the hospital to provide reassurance to the families who eagerly await the end of their children's hospital stay. Offering an accurate and reliable answer regarding the expected duration of the stay is of utmost importance. Dr Joan Sánchez de Toledo emphasizes the relevance of this, stating, "When an individual finds themselves in the 1% of the possibilities, it becomes a 100% reality for them". Therefore, it is always important to be cautious even when displaying highly certain predictions.

The relevance of this matter is currently noteworthy. The lack of evidence in this domain highlights the necessity for more extensive studies aimed at exploring approaches that can reliably forecast the LoS, especially in neonatal care (Seaton et al., 2016). In order to provide enhanced support and information to these patients and their families during their hospitalization, we address the crucial aspect of determining the LoS, which holds great importance for the well-being and satisfaction of the patients themselves (Borghans et al., 2012).

1.2 Research questions

The objective of this research is to provide insights and solutions to the following research questions pertaining to the Length of Stay in a hospital setting, specifically addressing the challenges faced at the Sant Joan de Déu hospital.

1. We aim to understand the characteristics of the LoS problem and evaluate different models to determine the most effective approach for making accurate predictions. We analyze and compare the performances of various off-the-shelf models and observe that their accuracies are comparable to S-CLASSY. Additionally, we assess the difficulty associated with solving the LoS problem by analysing the prediction errors of the algorithms.

2. We assess the effect of human guidance in the S-CLASSY model. We compare the output of the model both when using *simulated* and *real* knowledge. In addition, we compare it with its predecessor, CLASSY, and examine their predictions. We motivate our decision for the S-CLASSY model to tackle this task given the similar accuracy to CLASSY and the enhanced interpretability and adaptability.

3. We investigate the interpretability of the selected algorithm, S-CLASSY, to confirm our preference for this model. We focus on the accuracies obtained and the compactness of the rule lists. Fewer rules and conditions are generally preferred for better interpretability.

4. We iterate a second time with the medical experts, to get their impressions regarding the performance and interpretability of our approach. Furthermore, we propose ideas to enhance the usability of a new version of S-CLASSY, providing medically relevant rules. This avenue aims to improve trust and gain the acceptance of the experts involved.

1.3 Approach and contributions

Our objective is to comprehensively grasp the intricacies of this problem by leveraging data analysis and conducting interviews with the healthcare staff at Sant Joan de Déu in Barcelona.

It becomes clear that clinicians prioritize two key aspects in a predictive solution: accuracy and interpretability. They require a tool that can provide accurate predictions, serving as a reliable second opinion to their own estimations. This tool should not only confirm their initial assessments but also challenge their predictions when necessary, prompting them to reconsider their assessments and make more informed decisions.

To meet these requirements, the solution needs to be interpretable, allowing healthcare professionals to understand and trust the underlying reasoning and decision-making process, they need it to be intuitive. Moreover, it should align with the clinicians' thinking process by employing the same variables and metrics they use in their own assessments. Inspired by the human thinking process, our primary focus is on maximizing interpretability as a solid first step towards building trust and facilitating the future integration of the algorithm into the existing workflow of healthcare professionals.

To explore the LoS problem, we employ the data from Sant Joan de Déu and engineer multiple datasets utilizing distinct pre-processing techniques. Although the LoS problem is inherently a regression problem, we approach it as a classification problem in our study. Consequently, we generate four distinct label sets that define different LoS intervals. Therefore, we discretize the output in four different ways, to consider four different problems. This allows us to evaluate the performance of various methods while addressing diverse problem formulations.

In our first experiment, we validate the comparable accuracy achieved by the S-CLASSY algorithm in comparison to other off-the-shelf non-interpretable methods. For this assessment, we train the models with the different engineered datasets (exhibiting distinct variables) to solve the four classification problems. Following this procedure, separately for each problem, we can compare different metrics across the models. In addition, we prepare confusion matrices to understand what errors the different models are committing.

In our second experiment, we emphasize the model's ability to incorporate human guidance. In addition, we compare the performances of the different interpretable methods, namely a decision tree, CLASSY, and S-CLASSY with *simulated* knowledge and *real* knowledge.
We utilize the S-CLASSY model with *simulated* knowledge, following the same procedure as the authors of S-CLASSY, as one of our baseline methods. Through comparison to this model, we assess the relevance of actual *real* expert knowledge used in this work compared to what can be deduced using machine learning. To do that, we analyze the frequency of the preferred variables within the generation of the first rule, to confirm the benefits and impact of incorporating human guidance into the algorithm, while keeping track of the accuracies of the models.

In our third experiment, the focus of our investigation lies in assessing the interpretability of the algorithm. We evaluate interpretability quantitatively, employing various metrics to facilitate comparison between interpretable models. Further, we introduce an additional metric for rule-based models called *accumulated usage*. This measure illustrates the algorithm's capacity to provide comprehensive coverage of the data through robust rules.

In our last task, we gather feedback from healthcare professionals regarding the generated model. Our discussions with the professionals encompass various aspects, including the structure of the rule lists, interpretability, readability for clinicians, and the identified limitations of our solution.
Additionally, healthcare professionals propose ideas for better application. They recommend immediate improvements, including limiting the search space to a specific population and adapting again both the overall and preferred variables.
By implementing these changes, the generated rules become more transparent, enhancing their simplicity and comprehensibility. However, further research is needed to create patterns with clinical relevance that align with medical thinking procedures.

1.4 Outline of this thesis

Following this introduction, we delve into a review of the related work, in Chapter 2, in the domain of interpretable machine learning models and recent advancements in state-of-the-art models for predicting the Length of Stay in hospital settings.

Chapter 3 delves into the foundational concepts necessary for comprehending the CLASSY and S-CLASSY algorithms. We define the general rule learning framework and elucidate the workings of CLASSY, which leverages the minimum description length (MDL) principle as the rules selection criterion. Additionally, we introduce S-CLASSY, which incorporates human guidance and beam search into the rule learning process.

Building upon this groundwork, Chapter 4 focuses on the presentation of the data and requirements of Sant Joan de Déu. In Section 4.1, we outline the variables involved, perform exploratory data analysis, and propose a data pre-processing pipeline to engineer various datasets that are utilized later in the experiments chapter. Additionally, in Section 4.2, we delve into a description of the specific clinical requirements and subsequently translate it into a well-defined machine learning problem formulation in Section 4.3.

Moving on to Chapter 5, we provide details on the experimental setup and the metrics employed. We then proceed to address our four primary research questions, namely: the comparative performance of S-CLASSY against other non-interpretable models, the impact of human guidance, the interpretability of the model, and the analysis of the expert's feedback.

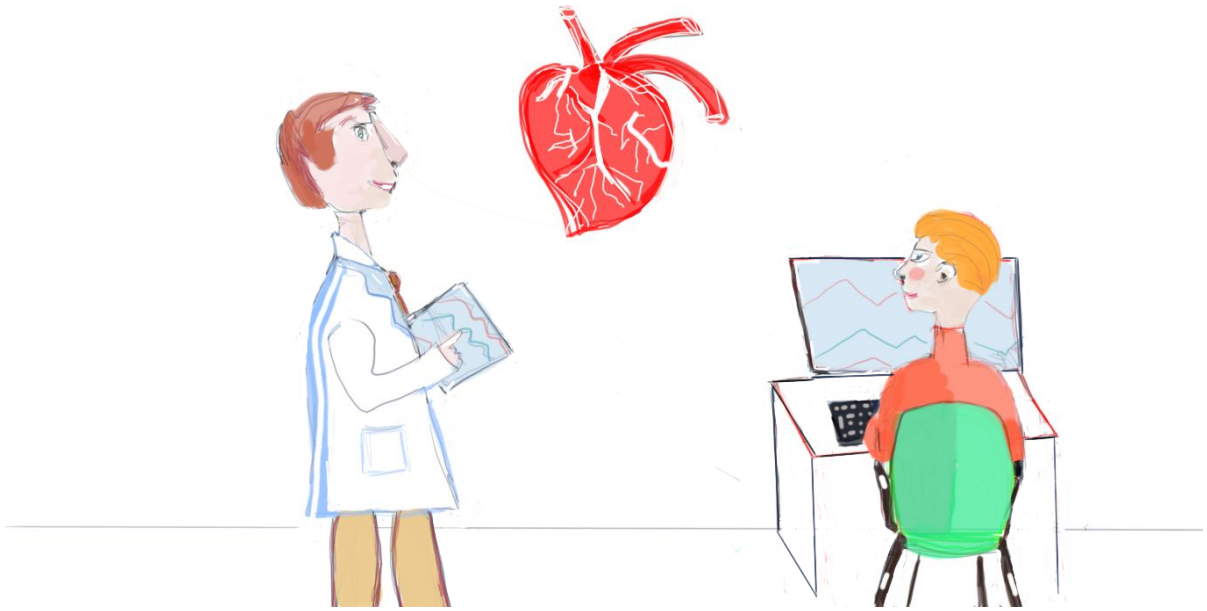
In Chapter 6, we engage in a discussion about the methods to address the Length of Stay (LoS) problem, the strengths and limitations of S-CLASSY to perform this task, and the new research line that emerges as a result of this work.

Lastly, in Chapter 7, we present our conclusions, summarizing the key findings of this study. Furthermore, we propose potential avenues for future research in this domain.

CHAPTER

2

RELATED WORK



In this chapter, we examine the literature related to the Length of Stay problem and interpretable machine learning techniques.
Firstly, we provide a comprehensive overview of the LoS concept, emphasizing its relevance across various fields and applications. Furthermore, we present a thorough review of existing works in the field of LoS, ranging from the utilization of heuristic metrics to the development of ad-hoc neural network models to solve this task.
Secondly, we delve into the literature concerning interpretable machine learning. We explore different rule-based and Bayesian methods that have been proposed and implemented, ultimately leading us to the adoption of the S-CLASSY method in this thesis.
Through this detailed exploration, we aim to establish a solid foundation for our research and identify the most appropriate method for addressing the LoS problem in an interpretable and effective manner.

2.1 Length of Stay

The Length of Stay of patients in a hospital has been extensively investigated during the last decades. In Rotter et al. (2008) it was observed how future research had to focus on properly understanding the clinical pathways of the patients in the hospital to address the problem better. Only in the last years, have machine learning and artificial intelligence techniques started to replace the heuristic methods employed previously. As stated in the survey of Stone et al. (2022) there's the need for a unified framework that still has not been reached. That is, understanding the data, justifying the models employed, using appropriate evaluation measures and model acceptance from the clinical community, among others.

Recent advances use the publicly available MIMIC datasets (Johnson et al., 2018). For example, in Wang et al. (2020) where they study the third version of MIMIC. Later, with MIMIC-IV (Johnson et al., 2020), a pre-processing pipeline was also prepared for it by Gupta et al. (2022), not only addressing the LoS problem, but also mortality or phenotype prediction. In addition, they offer a pipeline to pre-process the data and allow other researchers to address those problems with their own algorithms. Trained on MIMIC, new architectures are proposed like Temporal Pointwise Convolutions (TPC) (Rocheteau et al., 2021) and GRU-D (a new type of RNN incorporating masking and time intervals) (Che et al., 2018).

Even if frameworks are proposed to address the problem, it can be approached with all kinds of prediction models, both regression and classification. For instance, facing the problem with Random Forest (Mansouri et al., 2020), Gradient Boosting trees (Nemati et al., 2020), Autoencoders (Zebin et al., 2019) or ConvNets (Rocheteau et al., 2021), among others.

The survey conducted by Verburg et al. (2014) compared several existing regression methods for predicting the Length of Stay (LoS) of individuals. These methods were used in real-case scenarios for predicting the LoS in patients or as tools for ICU quality and efficiency. Their survey concluded that the studied models did not meet the requirements for accurate predictions.

More recently, several studies attempted to reapply one of the studied models in the previous survey. This is the APACHE IV model for LoS prediction. However, the findings of de Carvalho et al. (2020) indicated that the APACHE IV model was not reliable for predicting LoS in the ICU they examined. Similarly, Zangmo & Khwannimit (2023) demonstrated that the APACHE IV model exhibited poor performance in predicting ICU LoS specifically in patients with sepsis. To date, no successful studies have been reported in this particular field.

Furthermore, to the best of our knowledge, there is a lack of literature considering the prediction of Length of Stay using probabilistic or interpretable models. This research gap highlights the need for novel approaches that can provide accurate and interpretable predictions for LoS in hospital settings.

2.2 Interpretable machine learning

When looking for interpretable classifiers, the existing literature provides us with several potential solutions, such as decision trees, rule sets, and rule lists, among others. These models possess the advantage of not requiring extensive amounts of data, lengthy training processes, or a large number of hyperparameters. Our focus lies in the development of models that are easily understandable by human users.

We refer to Molnar (2020) for an extensive overview of interpretable machine learning models. Here, we provide a brief summary of the most relevant work.

Initially, we delve into the analysis of rule sets, meaning an unordered set of rules. A few approaches have been developed, particularly in the context of multi-class problems. Existing examples in the literature include IDS (Lakkaraju et al., 2016) and DRS (Zhang & Gionis, 2020). However, these approaches still retain some level of order and lack a probabilistic nature, thereby making it challenging to categorize them solely as strategies operating with sets. To address this limitation, Yang & van Leeuwen (2022) propose TURS, introducing a *Truly Unordered Probabilistic Rule Sets* solution.

When examining the *one-vs-all* approach, classic algorithms like RIPPER (Cohen, 1995) or FURIA (Hühn & Hüllermeier, 2009) need hyperparameter tuning and return ordered lists of rule sets, offering non-trivial interpretability of their results. Additionally, a limitation of these existing models is the absence of probability predictions, which means they do not provide an uncertainty measure for the predictions. This is a crucial aspect, especially in applications such as medicine, where having an indication of the prediction's reliability is essential.

Bayesian methods such as SBRL (Yang et al., 2017) have been proposed, but they are constrained to binary classification tasks and a limited number of candidate rules. The work of Aoga et al. (2018) proposes a solution combining probabilistic rule lists and the Minimum Description Length (MDL) principle. Nonetheless, it is again limited to binary classification and limited scalability.

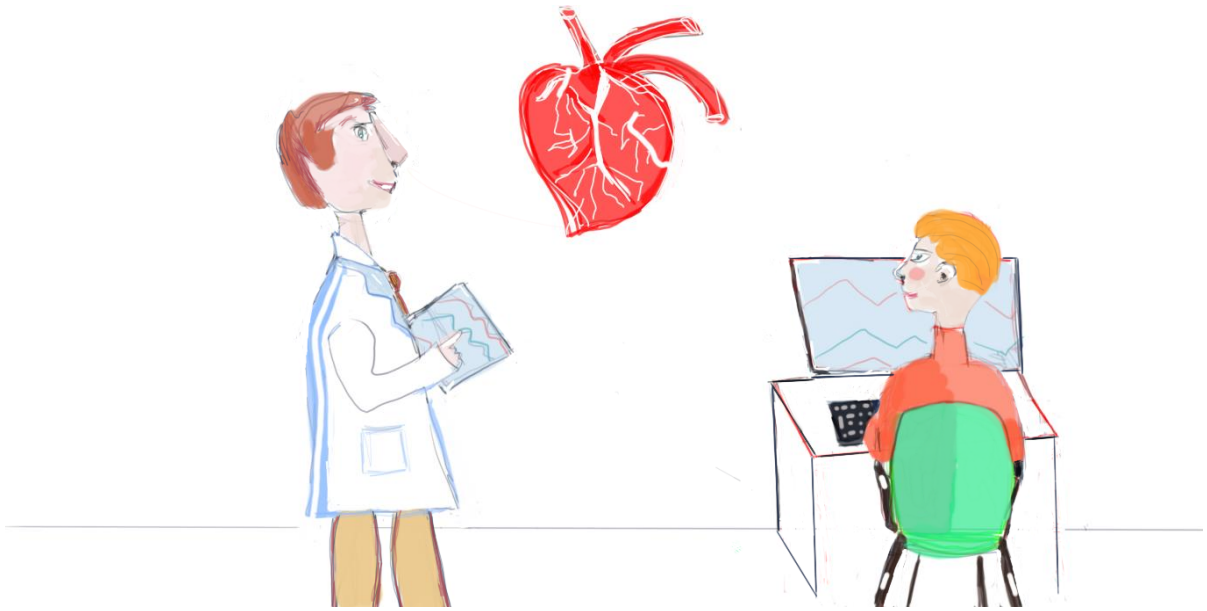
The limitations of being restricted to binary classification and limited scalability, as well as the emphasis on prediction tasks, were tackled by Proença & van Leeuwen (2020) with the development of the CLASSY algorithm. Later, Proença et al. (2022) proposed the SSD++ algorithm that uses beam search for candidate generation to find good rule lists in a straightforward fashion. Finally, S-CLASSY was proposed by Papagianni & van Leeuwen (2023) combining the two previous works and emphasizing the importance of leveraging the expertise of domain specialists in enhancing prediction tasks.

S-CLASSY does have a drawback in its utilization of lists, which introduces an additional layer of complexity in terms of interpretability due to the explicit order in the rule list. On the other hand, TURS, which generates rule sets, lacks the advantage of leveraging human guidance, which is particularly valuable given the scope and actors involved in this project. Consequently, we opt to use S-CLASSY for this work.

CHAPTER

3

RULE LEARNING



In this chapter, we provide a comprehensive overview of the interpretable models employed in this research. First, we introduce the fundamental principles of rule learning and present the MDL principle learning approach used in the CLASSY algorithm.
Second, we delve into the concept of human-guided rule learning, which involves incorporating domain experts' knowledge into the learning process. We outline how this concept is realized in the S-CLASSY algorithm and its potential application to the problem at hand.
By thoroughly defining both concepts, following the MDL principle and human-guided rule learning, we establish a solid framework for our study, enabling us to explore the effectiveness and interpretability of the S-CLASSY algorithm in addressing the LoS prediction problem.

3.1 MDL-based rule learning

3.1.1 Rule learning

Let us define $D = (X, Y)$ as a supervised dataset, comprising a variable set X and a multi-class label vector Y. Here, X represents the instance space, while Y represents the set of all $|Y|$ classes. Furthermore, we have $V = v_1, v_2, ..., v_m$ as the set of all $m = |V|$ variables in X, where each v_i corresponds to a one-dimensional variable with a domain of $dom(v_i)$. Within the dataset, each record $(x, y) \in D$ consists of an instance $x = (x_1, x_2, ..., x_m) \in X$, which is a vector of values where $x_i \in dom(v_i)$ for each v_i, and a class label $y \in Y$ that pertains to the respective instance.

Rule		**antecedent**		**consequent**	**usage**
1	IF	32 < bypass time (minutes) < 110	THEN	Pr(fast recovery)=75% Pr(slow recovery)=25%	26%
2	ELSE IF	surgery duration (minutes) >= 406	THEN	Pr(fast recovery)=12% Pr(slow recovery)=88%	28%
3	ELSE IF	average systolic blood pressure < 76.3 AND 18 < surgery overtime (minutes) < 95 AND standard deviation SpO2 >= 2.63	THEN	Pr(fast recovery)=0% Pr(slow recovery)=100%	10%
∅	ELSE		THEN	Pr(fast recovery)=52% Pr(slow recovery)=48%	36%

Table 3.1: Example of a probabilistic rule list obtained with S-CLASSY on the SJD data. *SpO2* refers to oxygen saturation in blood. We define "surgery overtime" as the minutes that the surgery has been extended longer than planned.

In this supervised learning setup, our aim is to learn rules from the data. In this context, a rule, denoted as r, is a conditional statement that connects patterns to class probabilities. Specifically, a rule consists of two components: an *antecedent*, represented by a pattern p, and a *consequent*, represented by a probability distribution $\pi(p)$. A pattern is a conjunction of conditions over variables, and $\pi(p)$ assigns probabilities to each class label in Y. See a rule list example in Table 3.1.

We describe the concept of a probabilistic rule list (PRL), denoted as R, which is an ordered list of $l+1$ rules: $(r_1, r_2, ..., r_l, r_\varnothing)$. The last rule in the list, $r_\varnothing$, is referred to as the default rule and has an empty antecedent along with a probability distribution $\pi_\varnothing$. The usage of a pattern p in the PRL is defined as the number of times the pattern occurs in the dataset D while disregarding instances that have already been covered by previous patterns in R.

Specifically, we illustrate the concept of label-oriented (or class-specific) usage, which focuses on the number of occurrences of a specific pattern p_i in the dataset D that corresponds to a particular class label l. This is calculated using the subset $D^{y=l}$, which consists of instances in D where the class label is equal to l. The label-oriented usage is expressed as

$$usg\left(p_i \mid R, D^{y=l}\right) = |\{x \subset D^{y=l} \mid p_i \sqsubseteq x \wedge (\bigwedge_{\forall j<i} p_j \not\sqsubseteq x)\}|.$$

All in all, a PRL addresses the problem of rule learning for multiclass classification, where the goal is to learn a rule list from a given supervised dataset. This rule list should be capable of accurately predicting the class labels for unseen instances as well as compact and reliable.

3.1.2 CLASSY MDL-based learning

One notable algorithm that addresses the need for concise rule lists is CLASSY (Proença & van Leeuwen, 2020). This algorithm utilizes the minimum description length (MDL) principle to construct rule lists. By employing a greedy approach, CLASSY explores the given dataset to identify informative rules that form the rule list. Following the MDL principle, often referred to as *induction by compression*, CLASSY seeks to find the rule list that best compresses the data, with compression serving as the guiding criterion. Notably, this approach eliminates the need for hyper-parameter tuning and mitigates the risk of overfitting.
In this supervised setting, the goal is to learn a mapping from instances to class labels. This means that the focus is not on discovering patterns within the instance data X itself, but rather on identifying patterns in X that aid in predicting the class labels Y.

To establish a relationship between instances and class labels, the PRL model discussed in the previous section is considered and the MDL principle is applied. Within the space of models $\mathcal{R}$, which comprises all ordered sets of patterns over X, given a Dataset for training D, the optimal model R^* is given by

$$R^* = \arg\min_{R \in \mathcal{R}} L(D, R) = \arg\min_{R \in \mathcal{R}} [L(R) + L(Y|X, R)],$$

where $L(R)$ is the length of the rule list encoding and $L(Y|X, R)$ is the encoded length of class labels Y given X and R.

The proposed heuristic by Proença & van Leeuwen (2020) is based on the concept of compression gain achieved by adding a rule r to an existing rule list R, denoted as $R \oplus r$. We present first the *absolute gain*, and then the *normalized gain*, as the latter is more suitable for the task at hand.

The absolute compression gain, denoted as $\Delta L(D; R \oplus r)$, measures the difference in code length before and after adding rule r to R. It can be decomposed into two components: *data gain*, $\Delta L(Y|X; R \oplus r)$, and *model gain*, $\Delta L(R \oplus r)$. The data gain captures the reduction in code length for the class labels given the instance data, while the model gain reflects the reduction in code length for the rule list itself.

In contrast, the normalized compression gain, denoted as $\delta L(D; X \oplus r)$, is calculated by dividing the absolute gain by number of instances activated by the pattern $p \in r$. By considering the normalized gain, the authors state that the model prioritizes rules that cover fewer instances but yield more accurate predictions compared to rules with higher coverage. This approach helps prevent the premature selection of large yet moderately accurate rules, which may lead to local optima and hinder the exploration of the search space.

The authors hypothesize that by utilizing a normalized gain instead of absolute gain, we can construct better rule lists during the greedy covering of the data, improving the overall performance and avoiding potential pitfalls associated with local optima. This last consideration is relevant to this thesis related to healthcare. In the healthcare domain, accurate predictions related to a few data instances are prioritized over more uncertain predictions that cover more data instances given the criticality of the problems.

3.2 Human-guided rule learning

The S-CLASSY algorithm (Papagianni & van Leeuwen, 2023) uses expert knowledge in the learning process. The type of knowledge taken into account is related to preferred variables, which are variables identified by the domain experts as having greater relevance compared to other variables when learning a predictive model. These preferred variables are given higher priority during the learning process, reflecting their importance according to the expert's guidance.

The authors of S-CLASSY combined the original CLASSY algorithm with the SSD++ beam search algorithm (Proença et al., 2022). S-CLASSY performs a beam search adding rules to the rule list following the compression criterion as its predecessors. However, firstly, a beam search is conducted for each preferred variable, starting the search from the first specific preferred variable. When the set of preferred variables is empty or the previous beam searches did not yield any candidate rules, a beam search is carried out considering all possible rules, as in the previous models.
This strategy guarantees that the search for candidate rules encompasses both the preferred variables and all variables. This aspect holds great relevance in our research, as we actively involve healthcare professionals in our study.

Since no expert knowledge was available for the datasets they tested in their work, they simulated this expert knowledge. For this task, they trained a random forest model on the entire dataset and ranked all variables based on entropy-based feature importance scores. Then, they chose a subset of variables as preferred variables for the model following that ranking. Additionally, they experimented with different preferred variable subsets, including a random subset and the most and least important ones based on the feature importance

scores. In the results, they notice that when selecting the most important variables or random variables, the model's performance remains similar. However, when using the least important variables to guide the search, they observe worse predictive performance in certain tasks.

CHAPTER

4

DATA PREPARATION AND MEANS

Within this chapter, we present our methodology for preparing the data to conduct our experiments.
In the first section, we provide a comprehensive description of the data utilized in our study. This includes an analysis of the various sets of variables available, as well as an exploratory data analysis to gain insights into their characteristics and distributions. Furthermore, we outline our pre-processing pipeline, which encompasses the application of different feature engineering approaches to refine the input variables. Additionally, we describe the creation of distinct engineered datasets by discretizing the LoS values in various manners to train the different classification models.
In the second section, we engage discussions with the medical team to gain more precise information about the medical problem under investigation. This collaboration allows us to leverage the expertise of the medical professionals and further refine our approach. By clarifying the requirements and simplifying the data, we ensure that our methodology aligns closely with the specific needs and expectations of the medical team.
Finally, in the third section, we transform the refined requirements into a machine learning problem . We define three sets of variables tailored to address the specific medical problem at hand and establish various problem configurations that serve as the basis for experimentation. This step enables us to bridge the gap between the medical requirements and the ML problem formulation, ensuring that our approach is well-suited to address the research questions effectively.

4.1 Data

In this section, we introduce the data used for this project.

The dataset of the pediatric hospital Sant Joan de Déu gathers the record of patients who underwent heart surgery and includes information spanning a period of two years.

We define the terminology utilized in Sant Joan de Déu. In this dataset, patients are distinguished by their unique patient identifier. Added to the patient's identifier, there is also the episode identifier, as over the course of their life, each patient may have multiple hospital admissions, each representing a distinct episode that commences with admission and concludes with discharge or mortality. These episodes can span a range from some hours to several months.
Within each episode, patients may experience one or more stays in various areas of the hospital, such as the Intensive Care Unit (ICU), the Operation Room (OR), or the ward. We refer to these multiple stays, within a hospitalization episode, as encounters.

At Sant Joan de Déu, patients are monitored both when being in the ward and in the ICU. In the ICU they are connected to the monitoring central device and in the ward they carry a mobile device that keeps track of their vital signs. In addition, once every eight hours, the nursery team notes the medications given, the laboratory results, the conditions, and the complications related to the patient. Furthermore, during surgery, the details of the procedure

are noted, which provides us with valuable information on how the procedure was, especially to interpret the posterior recovery process.

When it comes to the data, we sign an agreement with Sant Joan de Déu, agreeing that their data will only be used for this thesis' purposes and will be erased once the work is finished. They send us the data through a safe data transfer channel from Leiden University and we directly stored it on ALICE[1].

This section primarily centers on the data cohort selection, followed by a description of the variables, including those preferred by the clinicians. Subsequently, we conduct exploratory data analysis (EDA) before proceeding with the data pre-processing steps. Finally, we present the final processed data, ready for training the machine learning models.

4.1.1 Data selection

Following the guidance of medical doctors, our study focuses on a specific patient profile. At Sant Joan de Déu, the primary interest lies in predicting the LoS for patients undergoing complex surgeries, particularly the ones labelled as *cardiac surgery*. This subset of surgeries poses greater challenges in terms of predicting the recovery time. Therefore, we exclude encounters that followed cardiac electrophysiology or catheterization procedures. These procedures were deemed less invasive and did not present substantial difficulties in predicting the patient's recovery duration.

Figure 4.1 visually represents the sequential steps undertaken to select the appropriate data for the study. Several criteria are applied to ensure the relevance and precision of the dataset. Firstly, as only a small number of patients (specifically, three) experienced a fatal outcome, the focus of the study is shifted towards the recovery process.

To interpret the dataset properly, consecutive ICU or ward encounters are merged. In the hospital system, new encounters are created even if a patient remained in the same ICU or ward but was moved from one bed to another.

To concentrate solely on patients in the postoperative recovery phase, encounters following different paths, which deviated from the standard flow of patients from the operation room (OR) to the ICU or from the OR to the ward through the ICU, are removed. This selective approach ensures that our study focuses on patients undergoing their immediate post-operative recovery. As a result, in Chapter 5, we will predict the time that lasted this post-operative encounter.

Finally, in order to capture the initial episode of hospitalization, stays related to subsequent interventions are removed from the dataset. The decision to select only the first hospital stay for predicting the Length of Stay in the study is based on several considerations and objects of discussion during the meetings with medical doctors and data analysts. Firstly,

[1]ALICE (Academic Leiden Interdisciplinary Cluster Environment) is the high-performance computing facility of Leiden University and Leiden University Medical Center (LUMC). See `https://www.universiteitleiden.nl/en/research/research-facilities/alice-leiden-computer-cluster`.

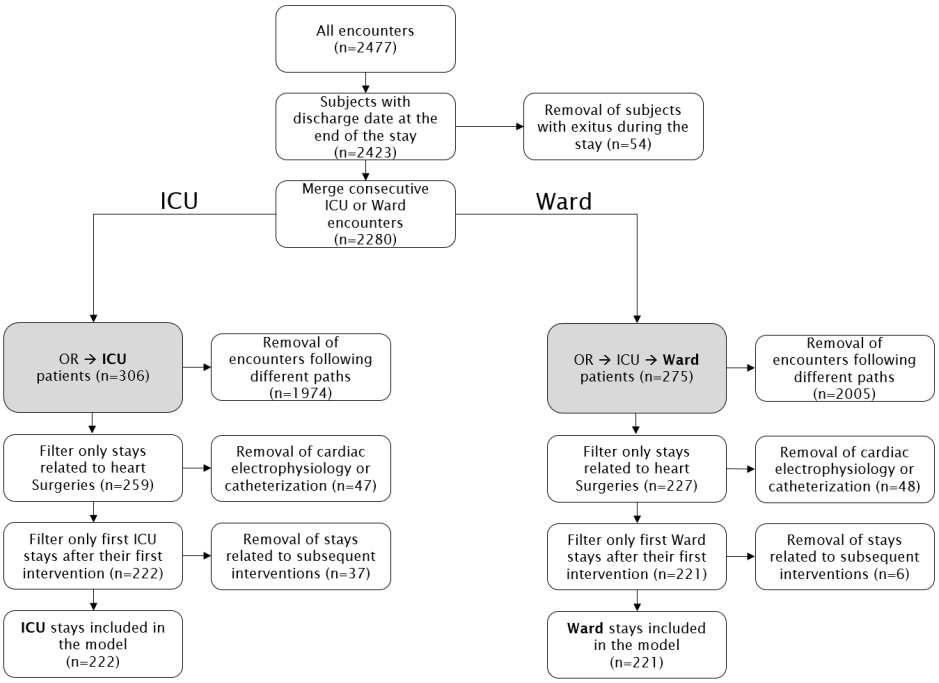


Figure 4.1: Data selection pipeline for Sant Joan de Déu's dataset.

focusing on the first hospital stay allows us to capture the initial episode of care and the primary surgical intervention. This approach enables us to minimize the confounding effects of subsequent interventions or complications that may arise during the patient's recovery. Subsequent interventions could notably impact the Length of Stay and introduce additional variability that might not be directly related to the initial surgical procedure. Therefore, by excluding subsequent stays, we aim to isolate the primary surgical intervention's influence on the LoS. Another reason for selecting the first hospital stay is the need to maintain consistency and ensure comparability across the dataset. Including multiple stays for the same patient would introduce dependencies and potential biases in the analysis. It is important to acknowledge that the exclusion of subsequent hospital stays does limit the scope of analysis to the immediate recovery period. Factors that influence readmissions or subsequent hospitalizations are not explicitly captured in this particular study. However, by focusing on the first stay, we can gain valuable insights into the immediate postoperative period and identify factors associated with the LoS during that specific timeframe.

As a result, we include 222 ICU and 221 ward encounters to train and test our model. These encounters belong to the same patients, after OR or after OR+ICU, with the difference of just one patient. This one patient, underwent their first surgery one day after their birth. Thirteen days after the surgery and the start of their ICU stay, they underwent another surgery (this is why they did not go to the ward after the ICU). This patient had multiple complications and surgeries. Their stay lasts for 92 days at the hospital until discharge. The patient would pass away 5 months after discharge.

4.1.2 Variables

Available variables

Sant Joan de Déu's database contains multiple data from the patient's stays, including demographics, surgery details, complications, medications, lab results, and vital signs. These multifaceted variables provide a comprehensive understanding of the factors that influence a patient's hospitalization duration. By exploring the relationships between these features and the Length of Stay, we can identify key determinants and potentially develop strategies to optimize resource allocation in healthcare settings.

Demographic information. We consider the variables age and sex. These variables allow for the examination of potential age or sex-related differences in surgical outcomes, helping the algorithm understand how these factors may influence the results. Age is a continuous variable that encompasses patients from zero days to 19 years old. Sex is a binary variable.

Surgery details. They encompass a range of important parameters related to the surgical procedure. These include the minimum body temperature recorded during surgery, the duration of the bypass method, the expected duration of the surgery, and the actual time taken to perform the surgery, among others. Another relevant factor are the scores, such as the STS score, which serves as an indicator of the surgical procedure's complexity. Analyzing these details allows us to explore the relationship between procedural factors and patient recovery, enabling us to find patterns between the surgical intervention and the posterior time to recover from it. In general, the times for performing techniques during surgery are noted in minutes, and the different scores are discrete variables.

Complications. The occurrence of post-surgical complications is an essential aspect to consider when studying the length of hospital stay. Complications such as cardiocirculatory arrest, pleural effusion, and reintubation can notably impact a patient's recovery trajectory and prolong their hospitalization. For instance, any of these complications can greatly increase the LoS of that patient in the ICU or the need for readmission in ICU when being in the ward. Understanding the relationship between these complications and the LoS provides valuable insights into the factors that contribute to extended hospital stays. The complication variables are discrete, noting how many times a complication occurred.

Medications. The drugs administered during the surgical process also play a role in determining the LoS. Anesthetic agents like midazolam or propofol, pain management drugs such as paracetamol or metamizole, and medications used to control stroke volume and cardiac output, like milrinone or adrenaline, may influence the patient's recovery progress. Examining the effects of these medications on the LoS can help identify optimal drug regimens that are useful during surgery but also promote faster recovery and shorter hospital stays. This set of variables comprises the following information: the timestamp indicating the time of medication administration, the dose of the medication (continuous variable), the dose normalized by the patient's weight (continuous variable), the unit of measurement (categorical variable), the medication code (categorical variable), and the route of administration (categorical variable, e.g., oral or intravenous).

Lab results. The laboratory assessments obtained during the patient's recovery phase provide valuable insights into their physiological status and progress. Parameters such as lactic acid levels and oxygen saturation levels are indicative of the patient's overall condition and response to the surgical intervention. Analyzing the relationship between these lab results and the LoS can provide valuable information on the impact of the patient's recovery trajectory on their hospitalization duration. This set of variables comprises the following information: the timestamp indicating the time of measurement, the value for that parameter (continuous variable), the interpretation of the value (categorical variable: *abnormal high*, *abnormal low*, or *normal*), and the unit of measurement (categorical variable).

Vital signs. Recorded as time-series data, they are vital indicators of the patient's physiological stability throughout their hospital stay. Monitoring parameters such as diastolic and systolic blood pressure, heart rate, respiration rate, SpO2 (oxygen saturation level) and temperature allows for the continuous assessment of the patient's health status. Certain changes or deviations in these vital signs, as well as peak values, can signal potential complications or recovery progress, thus influencing the LoS. The height and weight measurements are also part of the time-series, as those values can change during the patient's stay. However, they are often not measured properly so we decide to discard them. All these variables are continuous.

Expert's variables

In this section, we list the 21 variables that medical doctors generally can use to assess the criticality of a patient. Given the values of these variables, they decide the discharge moment for their patient.

1. Demographics
    - Age
2. Surgery metrics
    - Arrest time
    - Clamp time
    - Bypass time
    - Minimum temperature of the body
    - Emergency surgery or not
    - Surgery duration
3. Vitals
    - Average SpO2
    - Average Heart rate
4. Laboratory
    - Hemoglobin
    - Platelets

- Lactate
- Blood Glucose
- Urea

5. Complications

- Cardiac tamponade
- Pleural effusion
- Pneumothorax
- Sepsis
- Extracorporeal Membrane Oxygenation (ECMO)
- Cardiocirculatory arrest
- Reintubation

4.1.3 Exploratory Data Analysis

We explore the data, first by analysing the Length of Stay of the selected cohort patients both in the ICU and the ward. See Figures 4.2 and 4.3 respectively. The medical team explains to us that after the heart surgery, an extra dermal suture with staples is performed. These staples cannot be removed before 6 days after the surgery, this is why the stays generally last for one week in total, around 2 days at the ICU and 4 days at the ward after the surgery.

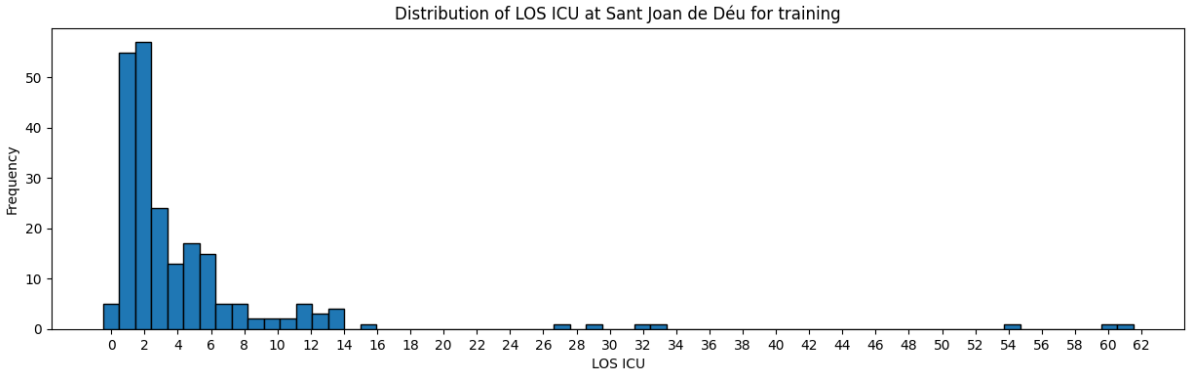


Figure 4.2: LoS values at the ICU, measured in days.

We observe that patients tend to stay more time in the ward compared to the ICU. We find that in the ICU the vital signs are gathered every 30 minutes, while in the ward, every minute.

We also look at the times when the patient's blood is pumped outside of the patient's body, allowing the surgeons to perform different practices. See Figure 4.4. First, the aortic clamp time allows the heart to rest. Then, the Bypass time allows the blood to bypass the heart and lungs through a machine. Finally, the cardiac arrest time, which is only used in extremely delicate situations, for time slots of a maximum of 40 minutes. Scanning in detail the data used for part (c) of the figure, only 5 patients are induced in arrest for a bit less than 40 minutes and one for almost 80 minutes.

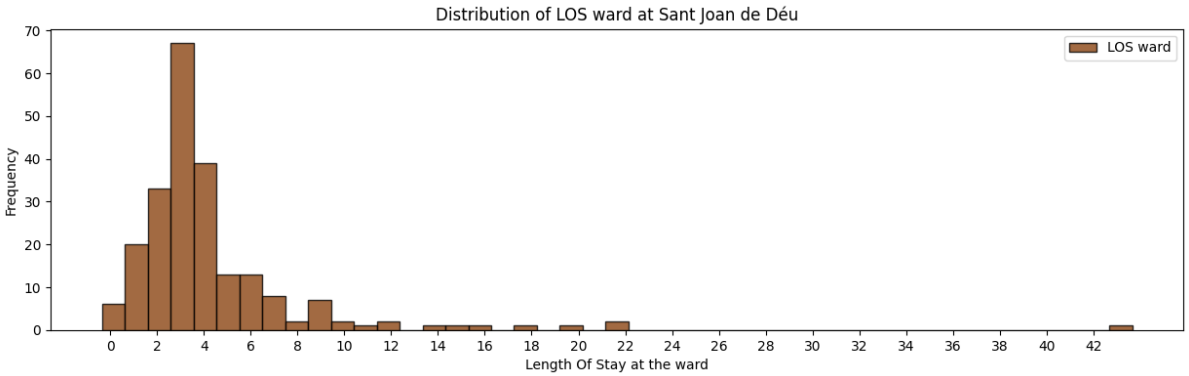
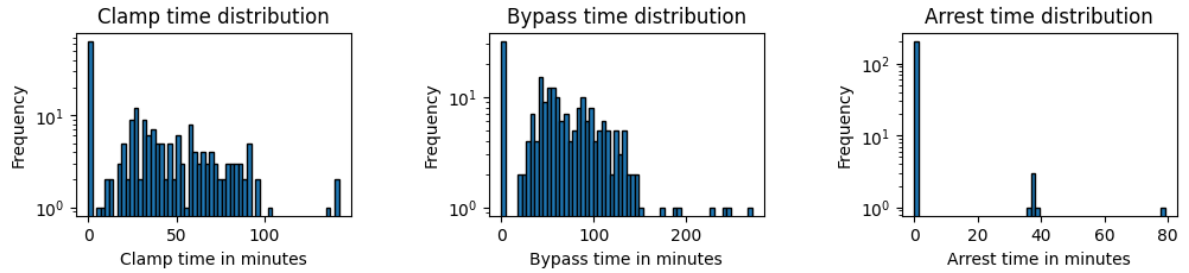


Figure 4.3: LoS values at the ward, measured in days.



(a) Time of clamp (b) Time of Bypass (c) Time of cardiac arrest

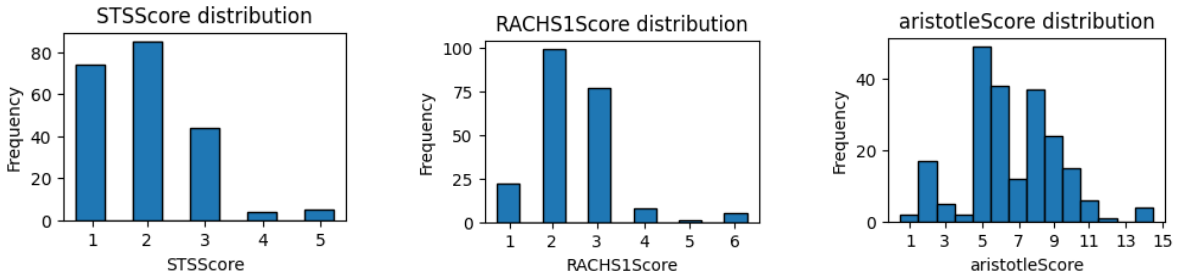
Figure 4.4: Surgery methods' times in the operation room (OR).

In Figure 4.5, we study the different metrics used to qualify the difficulty of the surgery. These metrics are highly correlated between them but follow different heuristics criteria. We try to find which one is the most related to the LoS output for the specific case of Sant Joan de Déu's patients and evaluate which one suits the hospital better. The STS score depicts $\mu = 1.96$ and $\sigma = 0.92$. The RACHS1Score presents $\mu = 2.44$ and $\sigma = 0.92$. Finally, the Aristotle score has $\mu = 7.18$ and $\sigma = 2.27$. A priori, we would prioritize using the Aristotle score, as the metric exhibits a wider range of values. Also, we observe increased variability in the data.

Figure 4.6 depicts the age distribution of the patients under study. It is worth saying that in the case of heart pathologies, these conditions are frequently identified even before birth in unborn babies. Consequently, a considerable number of newborn patients require surgical interventions within a few days following their birth.

4.1.4 Pre-processing

In this section, we show the different pre-processing techniques used to generate the three different resulting datasets for experimentation. We follow the pre-processing pipeline from Figure 4.7. As a result of slight differences between the three paths, we generate the datasets *plain data, windowing plain data*, and *windowing quantiles data*.



(a) STS score distribution (b) RACHS1 score distribution (c) Aristotle score distribution

Figure 4.5: Scores measuring the surgeries' difficulty.

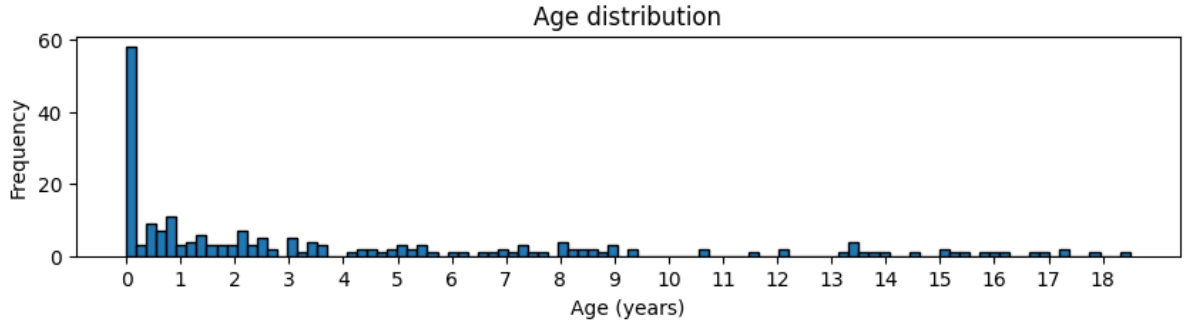


Figure 4.6: Age distribution of the patients when being admitted at the hospital.

Missing data handling

Recommended by the hospital team, we minimize the existence of null values by forward-filling the empty variables with their previous values. For instance, in the case of temperature measurements taken by the nursery team, it is common practice to not measure the patient's temperature precisely every 8 hours if the patient's recovery is progressing smoothly and there are no indications of a temperature change. The principle of "no news, good news" is applied, implying that if there are no recorded measurements for a specific variable at a given time, it is assumed that there is no cause for concern. Also, we discard the measures of height and weight as they are not measured for all patients and contain mistakes often.

Outlier detection and handling

We manually fix some values wrongly annotated, for instance, temperatures of "3737.0" degrees Celsius. Other values that are above the working range are discarded. In addition, the respiration rate is not measured with precision in the ward, so we discard this feature for the ward LoS predictions.

Windowing transformation

We propose a hypothesis regarding how the hospital team handles the LoS problem on a daily basis. To mimic the nurses' procedure, we decide to predict the patient's LoS at their current location every 24 hours based on the metrics gathered from the previous 24 hours.

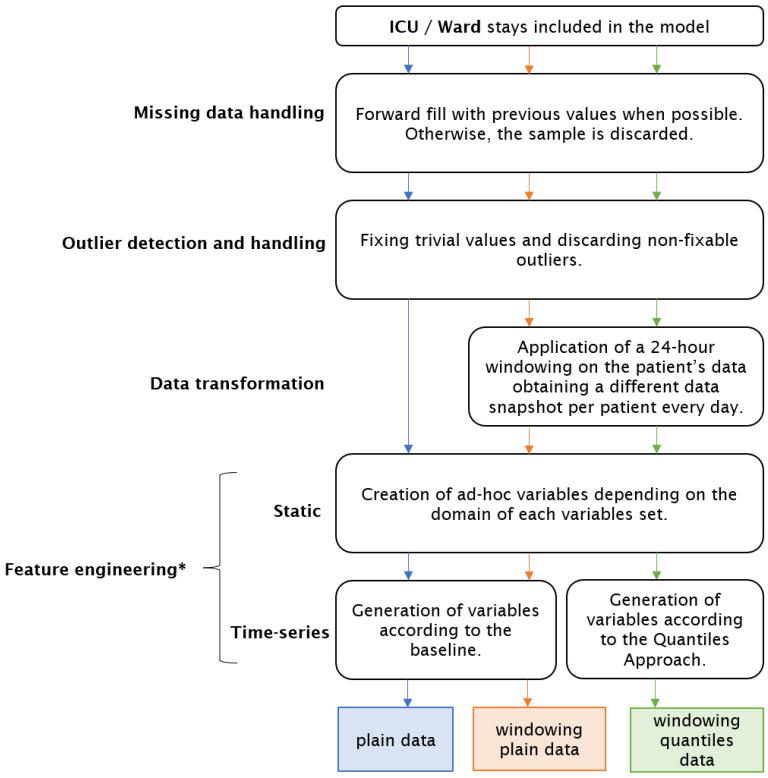


Figure 4.7: Feature selection and engineering pipeline. Further, the * in the feature engineering step indicates that the engineering of the variables for the static and the time-series data is done separately depending on their type and later merged to create each engineered dataset.

This approach aligns with their daily question of *"How many days does this patient need to be in ICU/ward given his surgery details and previous 24h of stay?"*.
Following this procedure, the number of data samples generated per patient is the number of days that last their encounter. Therefore, each data sample of a patient represents a snapshot of the patient's status on a particular day. See Figure 4.8.
Consequently, our training data includes the vital signs features from each day, combined with the rest of the static variables related to the medical procedure and the postoperative hospital stay. For example, let's consider an example where a patient stays in the ICU for three days after heart surgery. In this case, three distinct data samples are generated, each representing a specific time frame within the episode. The first sample comprises the data from the initial 24 hours and has a LoS of 2. The second sample captures the data from the subsequent 24 to 48 hours, with a LoS of 1. Finally, the third sample encompasses the data from the 48 to 72 hours period, with a LoS of 0.

Static variables engineering

In this section, we explore the engineered variables from the original static variable sets. Besides, simple operations are made, like calculating the age of the patient subtracting the admission date from the birth date and formatting that value in months. As it is a pediatric

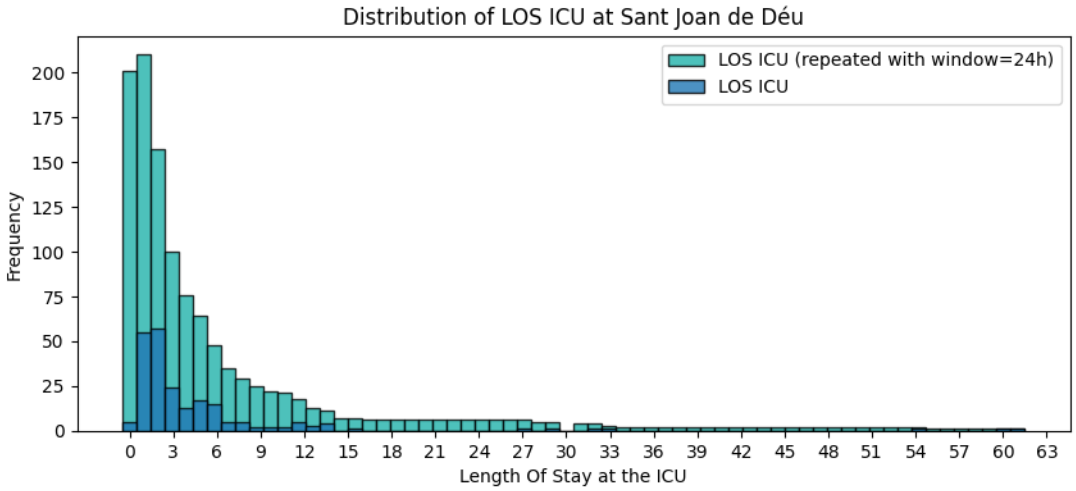


Figure 4.8: LoS values at the ICU for the original and windowed data samples.

hospital, the difference between ages is crucial.

First, for the surgery details, we engineer two additional variables. The first one is the estimated duration of the surgery compared to the actual time it took. This value *surgery overtime* can help us predict possible complications during the surgery, affecting the LoS of the patient. Second, patients generally have their surgeries on Tuesdays and Wednesdays, following the hospital's schedule, for which we create the feature *surgery weekday*. Any surgery not performed on those days could lead to different outcomes in the LoS.

Second, for the medications, we have 62 categories available, this is, 62 different medications that appear as administered at least once to any of the patients. The distribution of those is very unbalanced, with common drugs administered very often and others just once in the whole dataset. The variable that we use for training is the administered dose divided by the patient's weight, this way, we normalize the value to the patient's characteristics.

Third, for the laboratory results, 30 categories are present in the dataset. In this table, added to the times, amounts and units of the components measured, we have an indicator of either *normal*, *abnormal high*, and *abnormal low* measurement interpretation. We use this indicator to train the model, instead of using the measured amounts. Also, we create the variable *interpretations sum*, depicting the sum of abnormal values that the results are showing for that patient. This approach allows us to maintain an overall view of the laboratory data and provides an additional, even though potentially oversimplified, metric for the predictor.

Fourth, for the complications, there are 37 categories available in the dataset. However, when discussing them with the medical team, they indicate that many of them are not applicable to our model, as they don't affect the patient's recovery. So, we focus on the detection of the following, based on expert knowledge: *Unexpected ICU admission*, *Extracorporeal Membrane Oxygenation (ECMO)*, *Cardiocirculatory arrest*, *Reintubation*, *Sepsis*, *Pneumothorax*, *Pleural*

effusion, and *Cardiac tamponade*. Additionally, we add the variable *some event*, accumulating the number of adverse events that occur to this patient in that time period. We believe that this way we still keep track of the overall situation, even if not defining specifically which event is occurring.

Time-series variables engineering

To further explore the vital signs variables with their added temporal dimension, we opt to conduct additional experiments. Our objective is to evaluate the effectiveness of various time-series feature engineering methods in predicting the Length of Stay using the SJD dataset. In order to achieve this, we create multiple versions of the existing datasets specifically tailored for training purposes. These feature engineering methods enable us to generate diverse sets of variables that capture different facets and characteristics of the data.

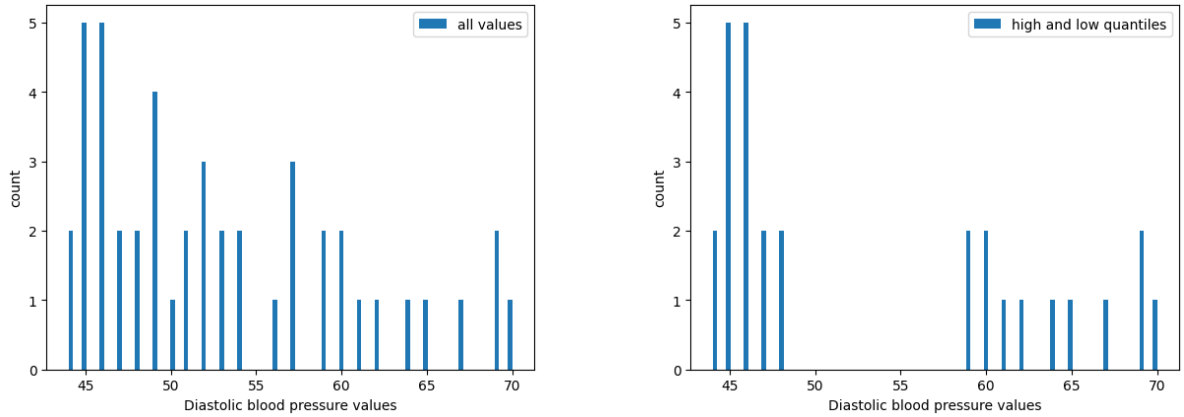
Baseline approach. We gather the vitals data containing *oxygen saturation (SpO2), heart rate, systolic and diastolic blood pressure, temperature*, and *respiration rate*. Then, we extract the following features for every variable: *mean, standard deviation, maximum, minimum, first quartile*, and *third quartile*.

The Quantiles Approach. For this feature engineering method, we follow the *Quantiles Approach* proposed by Alghatani et al. (2021) and perform the following pre-processing of the time series data.
First, for each vital sign, we calculate the mean and standard deviation. Second, we calculate the percent point function for each variable to get its low and high values using the calculated features in the previous step. The low values are the ones below the first quartile and the high values are the ones above the third quartile. Third, we remove the data points belonging to the interquartile range. Fourth, we calculate the mean and standard deviation of the resulting data. Fifth, to get the last new feature, we divide the number of resulting observations by the original number of observations, obtaining the quartile percentages of the new data.
In Figure 4.9, we describe the effect that this transformation results on our data. For instance, we pick the time-series data representing the Diastolic blood pressure of a patient for 24 hours. After the transformation, the data close to the mean values is removed, giving more presence to the anomalous values and influencing the new mean and standard deviation of the data.

All in all, we display *mean, standard deviation, low quartile, high quartile, new distribution's mean, new distribution's standard deviation*, and *quartile percentage* for each vital sign.

Final data for training.

Input variables X. The static data available consists of the demographics, surgery details, complications, lab results, and medications of the patient in the location (ICU or ward). In addition to all those static values, we also use the time-series data representing the vital signs of the patient, processed through different techniques.
With the different methods presented, we will experiment with three different engineered datasets: *plain, windowing plain*, and *windowing quantiles*.



(a) Original data

(b) After applying the transformation

Figure 4.9: Effect on the diastolic blood pressure data of one patient when performing the Quantiles Approach transformation.

1. **plain data.** We directly select the available data and apply the baseline method for time-series feature engineering.

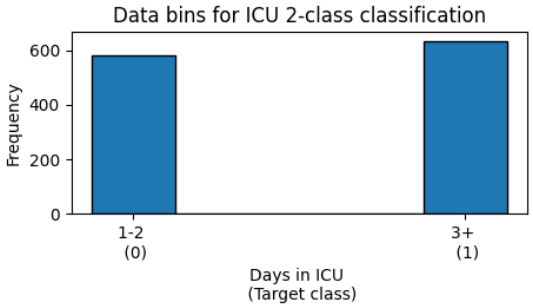
2. **windowing plain data.** We perform daily windowing on the data and apply the baseline method for time-series feature engineering.

3. **windowing quantiles data.** We perform daily windowing on the data and apply the Quantiles Approach method for time-series feature engineering.

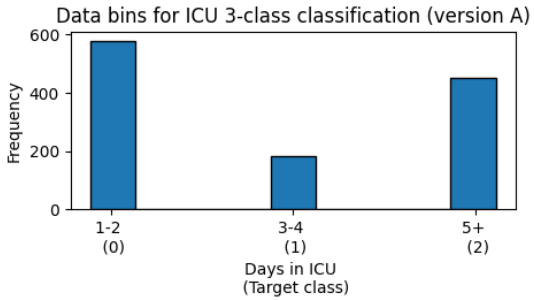
Terminology definitions

We would like to provide definitions that will aid in understanding the experiments, which involve various dimensions.
The resulting datasets X engineered with the *plain, windowing plain*, or *windowing quantiles* methods are referred to as *engineered datasets* or simply *datasets*. For instance, we may refer to the *windowing* method that generates the *windowing plain dataset*.
Subsequently, in the next section, we establish distinct shapes for the predictor's output Y that arise when transforming the regression problem into a classification problem. These output shapes determine the number of targets to be predicted and their distribution within the training data. We refer to them as *problems* or *data binnings*. Examples of such problems include the *2-class* or *3-class-A* data binnings, defining the labels for that problem.
Lastly, the supervised learning algorithms utilized for predicting the Length of Stay from a specific dataset are referred to as *algorithms*, *predictors* or *models*, including S-CLASSY, Random Forest, and Gradient Boosting, among others.

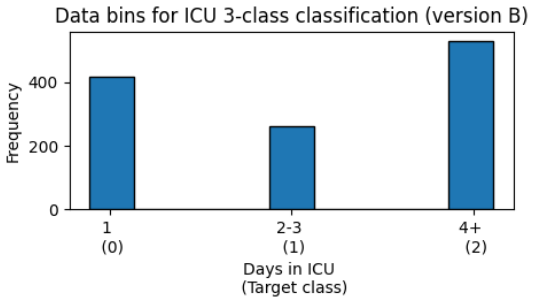
Output labels Y. The output features will also be a subject of study as we create data bins to accumulate several different LoS intervals (measured in days). This way, we transform this regression problem into a multi-class classification problem. This process only involves transforming the LoS timestamp to a category, therefore it is transparent to the engineered dataset used.



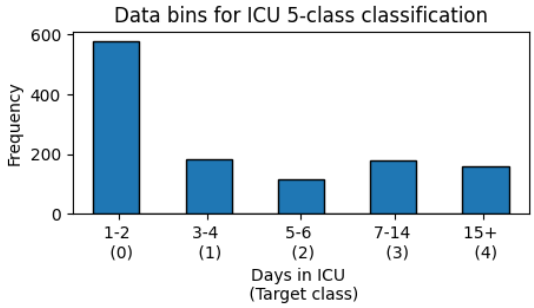
(a) Data binning for binary classification.



(b) Data binning for 3-class-A classification.



(c) Data binning for 3-class-B classification.



(d) Data binning for 5-class classification.

Figure 4.10: Output data distributions for the ICU at SJD, creating four different datasets.

The motivation for creating different problems, using different model targets is that in preliminary experiments, we find that the model's accuracies are very sensitive to changes in the manner of output discretization. Consequently, we can engineer different datasets from just one, offering us the capacity for more experiments. We prepare four discretization schemas: 2-class, 3-class-A, 3-class-B and 5-class, see Figure 4.10 for the case of the ICU. We perform similarly for the ward experiments, see Figure A.1 in the Appendix.

Therefore, we create four different problems to perform 2-class, 3-class (using two different versions) and 5-class classifiers to show how the different models perform. The more classes, the more difficult the problem is and the medical doctors should decide which classifier adapts the best to their needs, based on this trade-off.

4.2 The clinical requirements

Prior to conducting the final experiments, we engage in an essential exchange of insights and advancements with the medical experts, ensuring that our approach aligns closely with their needs and requirements. We share the methods and data utilized for training, as well as

preliminary results, which serve as a foundation for further discussion. During this interaction, the medical experts provide valuable input and suggestions for enhancing our approach.

Firstly, they emphasize the importance of developing rules that can be assessed quickly and easily by medical doctors, enabling them to comprehend the predictions effectively. To achieve this, they underscore the importance of using simple and clear variables in the rule list, facilitating the interpretability and usability of the model.

Secondly, the medical experts propose simplifying the origin of the variables used. Their objective is to simplify the dataset and generate rules utilizing variables from fewer sets of variables. They suggest exploring the possibility of excluding certain subsets of variables from the training process. This approach aims to streamline the dataset, guided by expert knowledge, and ensure that rules are generated using the most transparent variables.

Thirdly, recognizing the inherent differences in complexity between binary and 5-class classification problems, the medical experts recommend conducting further experiments to determine the most suitable approach for the hospital's specific needs.

Taking these valuable inputs into consideration, we refine our approach accordingly, focusing on the development of rules that are easily interpretable, and utilizing simpler and less number of variables.

4.3 The machine learning problem

In this section, our focus is on defining the classification problem, particularly regarding the selection of variables used for training. Additionally, we provide an overview of the final dataset used for our experiments.
To ensure meaningful performance comparisons, we evaluate the S-CLASSY model, informed by *real* expert knowledge, against S-CLASSY with *simulated* expert knowledge as a first baseline, and well-established benchmark methods commonly recognized in the literature.

4.3.1 Expert's variables selection

During the variable selection process for our experimental setup, we focus on adhering to the requirements and recommendations provided by the medical team at Sant Joan de Déu. They review an example of the rules generated using all 162 available variables. Subsequently, they make some observations about the output. They indicate that clinicians often make Length of Stay (LoS) predictions by considering only surgery details and vital signs in the Intensive Care Unit (ICU), so the algorithm should focus on using these variable sets. Additionally, reducing the number of variable sets used in the rule lists enhances interpretability, as fewer variables need to be taken into consideration. Therefore, we proceed to experiment with reducing the number of variables used for training to achieve improved interpretability.

Preferred subset for S-CLASSY

Furthermore, following the guidelines proposed by the authors of S-CLASSY, we collaborate closely with the medical team to establish a set of preferred variables. These preferred variables serve as guidelines for creating our rule lists within the S-CLASSY model.
Together with the medical experts, we identify the seven variables considered by the clinicians as the most important for predicting a patient's recovery time, resulting in the selection of the following variables, as the subset of variables to be *preferred* by the model: **age** (in months) from the demographics section; **surgery duration**, **clamp time**, **bypass time**, and **minimum temperature in OR**, from the surgery details section; and finally **average heart rate**, and **average SpO2** from the vitals data section.

Complete set for training

In preliminary experiments, we train the model with all the variables present, a total of 162 variables fed into the predictor for each sample. We observe long running times and complex output rules. That is, with a high number of conditions per rule and more than ten rules per rule list. We also observe not much presence of the features from the medications, laboratory results or complications. We run a random forest algorithm and explore the feature importances of the variables used. We observe that the variables from these categories have almost no effect on the prediction of the random forest model either, not being present enough on the dataset and the algorithms are not capable of learning from them. We test removing the medications and laboratory results variables from the training data, reducing the number of variables to 70. We understand that the model is greatly simplified, reducing the number of variables to less than half.

We run the S-CLASSY and Random Forest algorithms again, observing a low decay of the ROCAUC of around 0.01 in both but much more clear rules from this simplified dataset. We also plot the feature importances from the random forest algorithm, both Figures A.4 (for all the variables) and A.5 (for the selected 70 variables) can be found in the Appendix section.

After consulting with the medical team, we discuss the preliminary results. The team emphasizes their priority of having an interpretable but also transparent tool, expressing discomfort with the lack of transparency and simplicity that depict the variables associated with medications, laboratory results, and complications. Their guidance can be summarized as "the simpler, the better". Taking this into consideration, we proceed to further simplify the dataset, following the expert's guidance. Consequently, we exclude the variables related to complications and the variables associated with the operating surgeon from the final training of the models as the medical team highlights that these variables are not considered reliable indicators for the prediction task.

As a result, the final list of variables is defined in Table 4.1.

4.3.2 Final data for experimentation

Upon reviewing the characteristics of the engineered datasets presented in Table 4.2, notable observations can be made. The *plain* data exhibits a relatively small number of samples $|D|$

Variable Name	Type	Domain	Expert Variable?
age	**Numeric**	**Demographics**	**Yes**
sex	Binary	Demographics	No
surgery duration	**Numeric**	**Surgery Procedure**	**Yes**
surgery overtime	Numeric	Surgery Procedure	No
aristotle score	Numeric	Surgery Procedure	No
STS score	Numeric	Surgery Procedure	No
RACHS1 score	Numeric	Surgery Procedure	No
bypass time	**Numeric**	**Surgery Procedure**	**Yes**
clamp time	**Numeric**	**Surgery Procedure**	**Yes**
arrest time	Numeric	Surgery Procedure	No
minimum temperature in OR	**Numeric**	**Surgery Procedure**	**Yes**
surgery weekday	Numeric	Surgery Procedure	No
extubated in OR or not	Binary	Surgery Procedure	No
defibrillation in OR or not	Binary	Surgery Procedure	No
diastolic blood pressure	Time-Series *	Vital signs in ICU or ward	No
systolic blood pressure	Time-Series *	Vital signs in ICU or ward	No
heart rate	**Time-Series ***	**Vital signs in ICU or ward**	**Yes - the average value**
respiration rate	Time-Series *	Vital signs in ICU or ward	No
SpO2 (oxygen saturation level)	**Time-Series ***	**Vital signs in ICU or ward**	**Yes - the average value**
temperature	Time-Series *	Vital signs in ICU or ward	No

Table 4.1: The final list of variables for training. In bold the expert's variables. Further, * means that several variables are extracted from these time-series, depending on the method used.

available for training, specifically 220 samples for the ICU and 212 samples for the ward. These datasets also contain a substantial number of variables. It is worth mentioning that certain patients have missing measurements for vital signs such as temperature and blood pressure, leading to their exclusion from the dataset if we compare to the values presented in Section 4.1.1.
However, when applying a windowing approach to the data, where each window spans 24 hours, a substantial increase in the number of samples is observed for the remaining datasets. In addition, the *windowing quantiles* data results in a smaller number of samples, compared to the *windowing plain* data.

When taking a look at the available variables, the ward dataset consistently contains six fewer variables in $|X|$ compared to the ICU dataset, due to the exclusion of the *respiration rate* vital sign. Furthermore, it is important to note that the Quantiles Approach generates seven variables for each vital sign, whereas the baseline utilizes only six variables.

Additionally, we are interested in discovering associations in parts of the data that affect the results when using different target distributions.
Two versions of the 3-class output binning are introduced with a one-day difference in their binning. Specifically, we focus on the prediction of ICU LoS following the transformations depicted in Figure 4.10. The two versions are 3-class-A and 3-class-B. In the A version, the first bin encompasses patients who stay in the ICU for 1 to 2 days, while the B version selects patients with only 1 day in the ICU for the first bin. Moving on, the second bin includes patients who stay for 3-4 days in the A version, and 2-3 days in the B version. Lastly, the third bin accounts for patients who stay 5 or more days in the A version, and 4 or more days in the

B version.

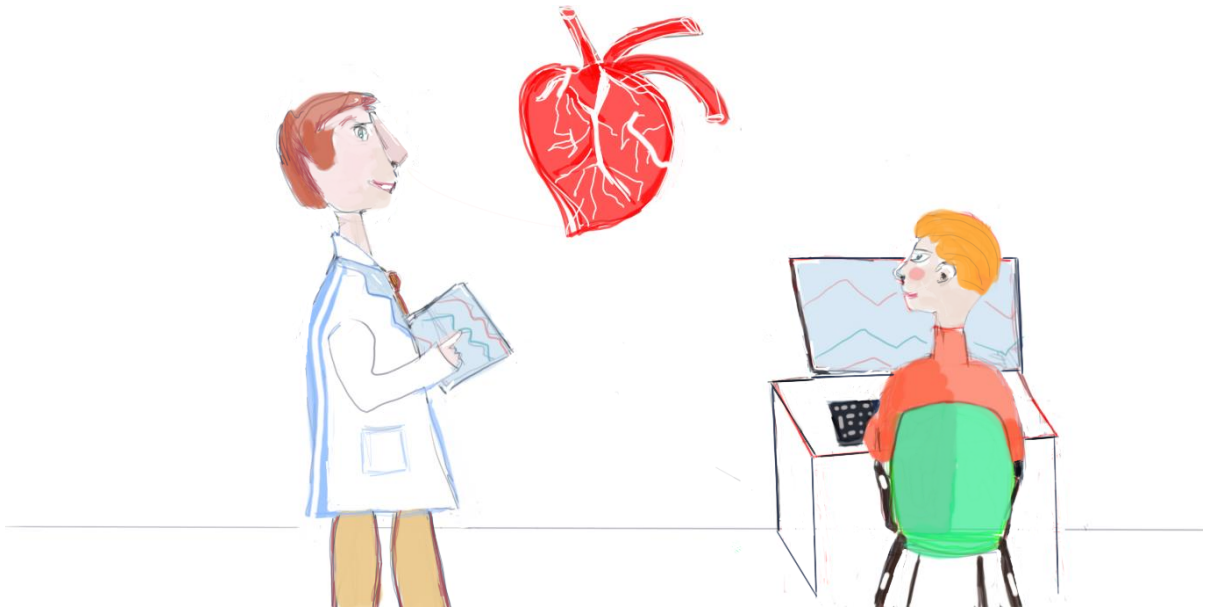
Dataset \ location		ICU						Ward					
		$\|D\|$	$\|X\|$	$\|x^{dem.}\|$	$\|x^{sur.}\|$	$\|x^{vit.}\|$	$\|Y\|$	$\|D\|$	$\|X\|$	$\|x^{dem.}\|$	$\|x^{sur.}\|$	$\|x^{vit.}\|$	$\|Y\|$
2-class	plain	220	50	2	12	36	2	212	44	2	12	30	2
	windowing plain	1.213	50	2	12	36	2	1.051	44	2	12	30	2
	windowing quantiles	1.188	56	2	12	42	2	962	50	2	12	35	2
3-class-A	plain	220	50	2	12	36	3	212	44	2	12	30	3
	windowing plain	1.213	50	2	12	36	3	1.051	44	2	12	30	3
	windowing quantiles	1.188	56	2	12	42	3	962	50	2	12	35	3
3-class-B	plain	220	50	2	12	36	3	212	44	2	12	30	3
	windowing plain	1.213	50	2	12	36	3	1.051	44	2	12	30	3
	windowing quantiles	1.188	56	2	12	42	3	962	50	2	12	35	3
5-class	plain	220	50	2	12	36	5	212	44	2	12	30	5
	windowing plain	1.213	50	2	12	36	5	1.051	44	2	12	30	5
	windowing quantiles	1.188	56	2	12	42	5	962	50	2	12	35	5

Table 4.2: Datasets characteristics: $|D|$ refers to the number of samples, and $|X|$ to the total number of variables, all binary or continuous. We specify the origin of these values in the different categories: $|x^{demog.}|$ are the two demographic values age and sex, $|x^{surgery}|$ are the surgery details variables, and $|x^{vitals}|$ are the variables generated based on the time-series data from the vital signs. Finally, $|Y|$ specifies the number of targets on each dataset.

CHAPTER

5

EXPERIMENTS



In this chapter, we begin by outlining the experimental setup, encompassing the details about the machine learning models and their configurations, and the metrics employed throughout the different experiments. Subsequently, we delve into the examination of the four experiments conducted in this thesis, addressing the research questions posed.

5.1 Experiment setup

To mitigate bias in model evaluation and obtain more reliable performance metrics, we conduct standard 5-fold cross-validation for all the problems. The *windowing plain* and *windowing quantiles* datasets utilize data windowing, generating multiple data samples from a single hospital stay. In the process of data splitting, there exists a possibility that some data points from the same hospital stay might be present in both the training and testing sets, which could lead to slightly inflated performance metrics. We further discuss the implications of this later in the document.
All metrics are calculated separately for each fold and we display their averaged values. For the confusion matrices and plots, we assemble all the predictions in the test set from the different runs. Nevertheless, the exhibited rules examples are model instances of one single run.

5.1.1 Models for comparison

In this section we describe the interpretable and non-interpretable models used in our experiments.

decision tree. This model serves as the sole off-the-shelf interpretable model we use for comparison. We initiate this experiment based on the observation that medical doctors can employ decision trees derived from medical literature to predict the LoS.

random forest. We compare the performance of the ensemble method random forest (RF) and S-CLASSY, which incorporates *real* expert knowledge. We initiate this experiment based on the observation that the RF algorithm assigns scores to the variables utilized, employing a ranking system based on entropy-based feature importance. The random forest method was used by the S-CLASSY authors as a baseline, so we decide to start our analysis by comparing it to this method.

multilayer perceptron. The multilayer perceptron (MLP) (Glorot & Bengio, 2010) designed for this comparison is a simple neural network with two hidden layers, ReLU activation functions for the hidden layers, Adam optimizer (Kingma & Ba, 2014) and a maximum training of 500 iterations (epochs). The hidden layer sizes are adapted to the input and output sizes of the model. The input layer varies between 50 and 55 nodes, while the first hidden layer ranges between 40 and 45 neurons. The second hidden layer consists of 20 neurons, and the output layer varies between 2 and 5 neurons, depending on the specific problem configuration. In preliminary experiments, we perform simple hyperparameter tuning through grid search. We test various sizes for the hidden layers, activation functions, and learning rates to design this configuration.

gradient boosting. The Gradient Boosting (GB) classifier (Friedman, 2002) combining 200 weak learners (regression trees). We use the default learning rate of 0.1. In addition, both RF and GB offer the advantage of being an ensemble method, encompassing multiple predictors, whereas S-CLASSY relies on a single rule list for its predictions.

5.1.2 Configurations

In this section, we present the configurations utilized for the different models and data employed in our experiments. In Section 4.3 , we outlined the population selected for our study and defined three distinct sets of input variables. We also engineered four different problems by discretizing the output LoS variable in varying ways.

For the experiments presented in this section, we focus primarily on the ICU data. While we performed similar experiments with the ward data for most of our research questions, we have included the results of those experiments in the Appendix section for the sake of simplicity.

We employ specific configurations for the algorithms used in our experiments. By standardizing these configurations, we aim to ensure consistency and fair comparison across the different models.

Firstly, for the S-CLASSY and CLASSY models, all experiments involve a minimum support threshold of $m_s = 5\%$, limiting the inclusion of infrequent patterns (even if with high certainty). The maximum pattern length is set to $|r|_{max} = 4$, which ensures that each rule contains no more than four conditions. This restriction is in line with the guidelines provided by the medical team, as the interpretability of the output should not be overly time-consuming. However, we do not limit the maximum amount of rules $|R|$.
Furthermore, we modify the number of cut points from the default value of 5 to 10. The number of cut points refers to the number of points used to discretize a single-numeric variable. Increasing the number of cut points provides the model with greater flexibility to identify more accurate patterns.

Secondly, for the ensemble methods, namely random forest and gradient boosting, we fix the number of weak learners to 200 for both models. For these models and also the decision tree, we set the maximum depth of the binary trees to four. This last parameter is equivalent to the pattern length $|r|$, mentioned above, required to define a pattern within the rule list models. We are limiting the weak learners of the ensemble methods to have a similar shape as our CLASSY algorithms.

5.1.3 Metrics

In this section, we discuss the metrics used for the evaluation of the models in the different experiments. We define metrics to evaluate performance, human guidance, and interpretability.

Performance

The following evaluation metrics provide different perspectives on model performance to assess the models' effectiveness.

Area Under the Receiver Operating Characteristic (AUROC). The AUROC is a commonly used metric for evaluating classification models. It measures the model's ability to distinguish between positive and negative samples across different probability thresholds. The Receiver Operating Characteristic (ROC) curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various threshold values. The AUROC represents the area under this curve and provides an aggregated measure of the model's performance. An AUROC score of 1 indicates a perfect model, while a score of 0.5 suggests a random classifier. We use this metric as it is commonly used in the works we are comparing with.

Area Under Precision-Recall Curve (AUPRC). The AUPRC is another metric used to assess classification models. It evaluates the trade-off between precision and recall at various probability thresholds. The Precision-Recall curve plots precision (positive predictive value) against recall (sensitivity) at different thresholds. The AUPRC represents the area under this curve. It is particularly useful when dealing with imbalanced datasets, as is often our case. Higher AUPRC scores indicate better model performance.
In the sections where we assess human guidance and interpretability, we also use this metric, in conjunction with human guidance and interpretability metrics, to maintain a perspective on the models' performance.

F1 Score. The F1 score is a metric that combines precision and recall into a single value, providing a balanced measure of a classification model's performance. It is calculated as the harmonic mean of precision and recall. The F1 score ranges from 0 to 1, with 1 indicating perfect precision and recall, while 0 suggests poor performance. It is often used when the dataset is imbalanced or when both precision and recall are equally important. In our experiments, we use this metric averaging the result in a *weighted* fashion. So, we average each label's metric based on their support, for better comprehension of the performance.

Log loss The log loss, also known as logarithmic loss or cross-entropy loss, is also commonly used for evaluating the performance of classification models. It measures the accuracy of the predicted probabilities generated by a model compared to the true target labels.
The log loss penalizes incorrect predictions more severely, especially when the predicted probability is far from the true label. The added value of this metric for our experiments is mainly that it provides a continuous measure of the model's confidence in its predictions, making it particularly useful when dealing with probabilistic classification models.
A lower log loss value indicates better model performance, with perfect predictions yielding a log loss close to zero. Conversely, higher log loss values indicate poor model performance and a lack of confidence in the predictions.

Overfitting degree. Overfitting occurs when a model learns the training data too well and fails to generalize to unseen data. To evaluate overfitting, we calculate the average AUROC difference between the training and testing sets in our 5-fold schema. If the model performs notably better on the training set than the test set, it indicates overfitting.

Human guidance

Frequency of Preferred Variables. We use the frequency of the preferred variables in the first rule of each rule list, denoted as $f@1$. This metric is introduced in the S-CLASSY

work by Papagianni & van Leeuwen (2023). For this analysis, we take into account both the frequency and position of the preferred variables within the generated rule lists. This score could range from zero to four in our setup, as the maximum pattern length is set to $|r|_{max} = 4$. The models can employ the preferred variables for all the conditions in one rule or not use them at all. Higher values of $f@1$ denote more influence of human guidance.

Interpretability

The following metrics provide quantitative measures to assess the interpretability of rule-based machine learning models. The aim is to strike a balance between simplicity and accuracy, ensuring that the generated rules are both understandable and effective in capturing the patterns in the data.

Average Rule Length. The average rule length $\mu|r|$ measures the average number of conditions present in each rule within the rule list. A shorter rule length indicates simpler and more concise rules, which are generally easier to interpret and understand. On the other hand, longer rule lengths can make it harder to comprehend the decision-making process of the model. As defined earlier, rule length is limited to a maximum of four.

Average Number of Rules. The average number of rules in a rule list $\mu|R|$ measures the average count of rules in the full rule list. A smaller number of rules typically leads to more interpretable models. Having fewer rules makes it easier to identify and comprehend the multiple data patterns used by the model. Conversely, a larger number of rules may introduce additional complexity and make it more challenging to interpret and extract insights from the model.

Accumulated Rule Usage Percentage. We define the accumulated rule usage percentage as $\sum usg\ \%$ (or its equivalent $1 - usg_{\varnothing}\ \%$). This metric evaluates the accumulated coverage of the rules within the rule list, without taking into account the default rule. This metric is expressed as a percentage, as we divide the sum of rule usage by the number of samples in the training set. A higher percentage suggests that a larger portion of the dataset is explained by the rules, which can aid in understanding the model's decision-making process. Conversely, a lower percentage indicates that the default rule is covering a bigger fraction of the data, making it harder to interpret the model's predictions.

5.2 Performance

5.2.1 Scores

One of our primary research questions focuses on the rationale behind selecting an interpretable model, specifically S-CLASSY, over a non-interpretable model, such as neural networks, considering the potential differences in their performance. Therefore, we investigate whether S-CLASSY, as an interpretable model, can achieve performance levels that are on par with non-interpretable models of similar complexity, while also providing the crucial advantage of interpretability.

Problem	Metric	Model \Pre-processing	plain	windowing plain	windowing quantiles
2-class	AUROC	S-CLASSY	0.719	0.674	0.657
		RF	0.88	0.751	0.746
		MLP	0.845	0.687	0.694
		GB	0.848	0.732	0.73
	AUPRC	S-CLASSY	0.654	0.649	0.628
		RF	0.856	0.731	0.738
		MLP	0.843	0.684	0.698
		GB	0.826	0.719	0.73
	F1	S-CLASSY	0.669	0.667	0.635
		RF	0.726	0.7	0.698
		MLP	0.745	0.654	0.672
		GB	0.785	0.696	0.681
	log loss	S-CLASSY	0.6	0.668	0.671
		RF	0.466	0.566	0.565
		MLP	1.017	1.005	0.849
		GB	1.053	0.684	0.717
	overfit	S-CLASSY	0.117	0.153	0.163
		RF	0.114	0.136	0.148
		MLP	0.155	0.227	0.197
		GB	0.152	0.268	0.27

Table 5.1: Performance results comparing S-CLASSY to the non-interpretable models. We display the results for the 2-class classification problem.

In Table 5.1 we present the different performances of S-CLASSY compared to the non-interpretable models for the 2-class classification problems in the ICU data. We show the results of each model with the different datasets generated through the three different methods: *plain, windowing plain*, and *windowing quantiles*.

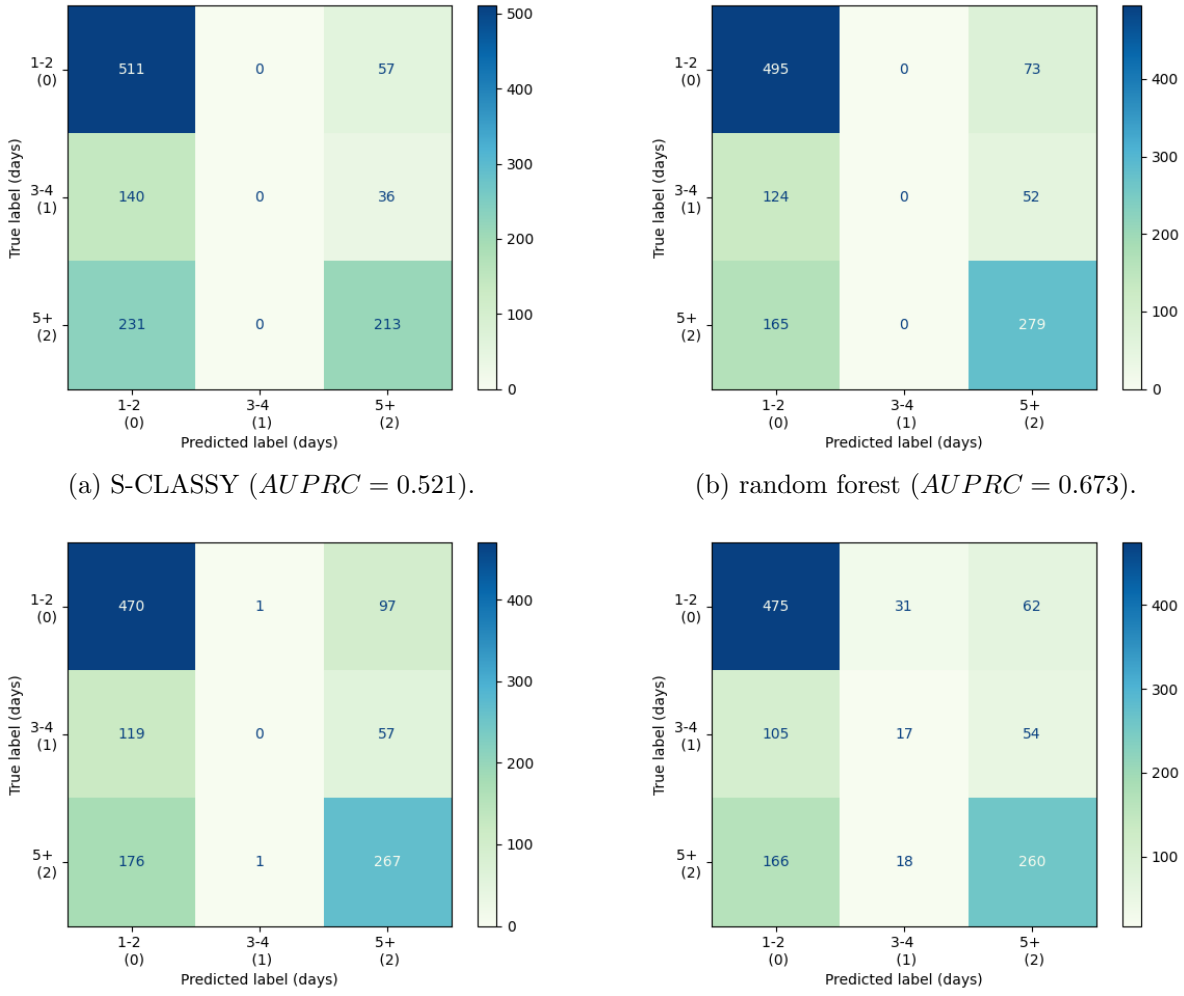
When examining the binary problem, we find that ensemble methods are the most effective performers, followed by the multilayer perceptron and S-CLASSY.
In the case of training on the *plain dataset*, all the compared models demonstrate significantly better performance than S-CLASSY. However, the MLP and GB models exhibit notably higher loss and overfitting values compared to the random forest, which emerges as the best performer. Nonetheless, when trained on data with windowing, S-CLASSY achieves performance values much closer to those of the other models. Specifically, for the windowing data, the difference in AUROC values between S-CLASSY and the best performer, RF, is 0.082, while in the previous case, the difference was 0.202.

In general, the *windowing plain*, and *windowing quantiles* datasets allow the models to learn better from the data than the *plain* data. For S-CLASSY, the AUROC values are 0.581 for *plain*, 0.596 for *windowing plain*, and 0.601 for *windowing quantiles*. However, other methods, like gradient boosting, might benefit from other data representations, as shown depicting better AUPRC values with *plain* than with *windowing quantiles* data.

We provide additional experimentation between all these models, comparing their performances in all the designed setups, and observing similarities in the different metrics. The results of these experiments can be found in the Appendix, in Tables A.1 and A.2. For more details

about the *windowing quantiles* results, see the ROC and precision-recall curves in Figures A.2 and A.3 in the Appendix.

To gain a deeper understanding of how these metrics reflect the challenges in learning, we undertake further investigation to analyze the specific errors made by the models. We aim to uncover insights into the areas where the models struggle.



(a) S-CLASSY ($AUPRC = 0.521$).

(b) random forest ($AUPRC = 0.673$).

(c) multilayer perceptron ($AUPRC = 0.63$).

(d) gradient boosting ($AUPRC = 0.664$).

Figure 5.1: Confusion matrices comparison between models for the 3-class version A ICU problem, using the *windowing quantiles data*. A total of 1188 data points are presented.

5.2.2 Confusion analysis

Our interest is understanding the specific choices of a determined model. By evaluating the true and predicted labels for each prediction on the test sets, we can discern the specific choices made by a particular model. When looking at Figure 5.1, we first observe the class imbalance. The data bins *1-2days*, *3-4days*, and *5+days* are in 568, 176, and 444 instances

respectively in the *windowing quantiles* dataset (this is the sum of values in each row of each matrix). We observe that the bin *1-2days* is also the most predicted by all the algorithms, as it is also the most common to occur.

However, the bin *3-4days* is difficult to predict, as can be easily confused with the previous or the posterior one and occurs much less than the others. In this example, this bin (predicting 3-4 days in ICU) is frequently misclassified by the MLP and GB models, leading to incorrect predictions most of the time. S-CLASSY and RF are never predicting it.
In the case of S-CLASSY, we observe how all rules decide either bin *1-2days* or bin *5+days*. The algorithm focuses on finding the differences in the data from class *5+days*, compared to the majority of data belonging to class *1-2days*. The target *3-4days* is never predicted, not being able to find any pattern for this label. The default rule of the algorithm is predicting the most common class, *1-2days*, after discarding the two others. See the rule list for S-CLASSY in Listing 5.1.
When it comes to random forest, the trees that are part of the ensemble are mainly not able to learn the patterns in the data that would lead to bin *3-4days*. This is why in this example this class is never predicted, focusing more on properly differentiating class *1-2days* from class *5+days*.

Based on the findings from this section, it can be concluded that the dataset poses relevant challenges for prediction, as all the methods employed struggle to achieve accurate results. However, it is important to note that the S-CLASSY model shows lower accuracies than the rest of the models, but those are still comparable, especially when trained on the windowed data. Therefore, S-CLASSY emerges as a promising model, showcasing its potential for effectively predicting the Length of Stay of patients in a hospital setting, even in the face of a challenging dataset.

```
//      Preferred variables: ['age_months', 'surgery duration',
//      'clamp time', 'bypass time', 'min. temperature in OR',
//      'average Heart rate', 'average SpO2']
//      Rule list:
IF      min. temperature in OR < 19.9
        AND percent.Q Respiration rate < 0.35
        AND average SpO2 < 96.3 THEN usage = 58;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.0 Pr(5+days) = 1.0

ELSE IF surgery duration >= 396.0
        AND high_q-Diastolic blood pressure < 54.8
        AND 24.3 <= avg_q-Respiration rate < 48.7
        AND percent_q-Heart rate < 0.59 THEN usage = 44;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.0 Pr(5+days) = 1.0

ELSE IF average Systolic blood pressure < 75.33
        AND 11.0 <= surgery overtime < 69.0
        AND 2.3 <= std SpO2 < 4.2 THEN usage = 36;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.0 Pr(5+days) = 1.0

ELSE IF low_q-Heart rate >= 138.31
        AND std_q-Respiration rate < 5.34 THEN usage = 17;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.0 Pr(5+days) = 1.0

ELSE IF 19.9 <= min. temperature in OR < 22.0
        AND 11.0 <= surgery overtime < 29.0
        AND average Heart rate >= 119.38
        AND high_q-Diast. blood pressure < 54.80 THEN usage = 26;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.0 Pr(5+days) = 1.0

ELSE IF std_q-SpO2 >= 4.62
        AND 0.5 <= percent_q-Temperature < 0.66
        AND percent_q-SpO2 >= 0.38 THEN usage = 22;
//      Pr(1-2days) = 0.0 Pr(3-4days) = 0.045 Pr(5+days) = 0.955

ELSE IF average Respiration rate < 29.05
        AND surgery duration < 240.0
        AND 1.0 <= STSScore < 2.0 THEN usage = 50;
//      Pr(1-2days) = 1.0 Pr(3-4days) = 0.0 Pr(5+days) = 0.0

ELSE IF high_q-Diastolic blood pressure >= 58.71
        AND extubated in OR
        AND percent_q-Diastolic blood pressure < 0.44
        AND surgery duration < 325.5 THEN usage = 47;
//      Pr(1-2days) = 1.0 Pr(3-4days) = 0.0 Pr(5+days) = 0.0

ELSE IF age_months >= 64.0
        AND percent_q-Respiration rate >= 0.35
        AND std Heart rate < 8.32 THEN usage = 58;
//      Pr(1-2days) = 0.931 Pr(3-4days) = 0.069 Pr(5+days) = 0.0

ELSE IF std SpO2 < 1.25
        AND 36.28 <= high_q-Temperature < 37.07
        AND 88.36 <= high_q-Systolic blood pressure < 103.34
        AND low_q-Diast. blood pressure < 55.03 THEN usage = 31;
//      Pr(1-2days) = 1.0 Pr(3-4days) = 0.0 Pr(5+days) = 0.0

ELSE IF 35.98 <= average Temperature < 36.15
        AND 83.5 <= bypass time < 134.0
        AND average Respiration rate < 33.79 THEN usage = 22;
//      Pr(1-2days) = 1.0 Pr(3-4days) = 0.0 Pr(5+days) = 0.0

ELSE IF 196.0 <= surgery duration < 287.0
        AND avg_q-Diastolic blood pressure >= 53.93
        AND std_q-Diastolic blood pressure >= 9.31
        AND std Respiration rate < 7.10 THEN usage = 26;
//      Pr(1-2days) = 1.0 Pr(3-4days) = 0.0 Pr(5+days) = 0.0

ELSE usage = 513;
//      Pr(1-2days) = 0.437 Pr(3-4days) = 0.265 Pr(5+days) = 0.298
```

Listing 5.1: Rule list produced by S-CLASSY with *real* expert knowledge for the 3-class *windowing quantiles* problem. All time values are denoted in minutes. The total training samples for this example is 951.

5.3 Human guidance

5.3.1 Scores

The objective of this section is to provide justification for the involvement of human experts in guiding the search process. Similar to the previous section, we conduct a comparison of accuracies and losses to ascertain the impact of expert insights on classification performance. Specifically, we compare two versions of S-CLASSY: one utilizing *real* expert knowledge and another utilizing *simulated* expert knowledge. Additionally, we compare these versions to the decision tree and CLASSY models that do not incorporate human guidance constraints.

In conducting these experiments, we establish the expert variables as follows. Firstly, for the *real* expert knowledge, we employ the variables that were described in the methods section, consistently across all four different problems. Secondly, for the *simulated* expert knowledge, as each problem is distinct, the feature importance scores obtained from the random forest model yield different results.

Consequently, the S-CLASSY model incorporating the *simulated* expert knowledge adapts by selecting different preferred variables for each specific problem, effectively tailoring itself to the task at hand. For instance, the feature importance ranking for the 2-class problem is detailed in Figure A.6, in the Appendix.

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
2-class	AUROC	S-CLASSY *real* kn.	0.719	0.674	0.657
		S-CLASSY *simulated* kn.	0.732	0.55	0.626
		CLASSY	0.711	0.681	0.669
		Decision Tree	0.676	0.627	0.623
	AUPRC	S-CLASSY *real* kn.	0.654	0.649	0.628
		S-CLASSY *simulated* kn.	0.637	0.558	0.61
		CLASSY	0.616	0.65	0.642
		Decision Tree	0.643	0.617	0.615
	F1	S-CLASSY *real* kn.	0.669	0.667	0.635
		S-CLASSY *simulated* kn.	0.688	0.474	0.616
		CLASSY	0.649	0.686	0.641
		Decision Tree	0.705	0.612	0.628
	log loss	S-CLASSY *real* kn.	0.6	0.668	0.671
		S-CLASSY *simulated* kn.	0.6	0.695	0.716
		CLASSY	0.671	0.638	0.66
		Decision Tree	3.927	1.114	1.146
	overfit	S-CLASSY *real* kn.	0.117	0.153	0.163
		S-CLASSY *simulated* kn.	0.098	0.063	0.192
		CLASSY	0.125	0.147	0.16
		Decision Tree	0.285	0.216	0.225

Table 5.2: Performance results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the 2-class classification problem.

The seven *simulated* preferred variables for this task are the following: **first quantile systolic blood pressure**, **average systolic blood pressure**, **arrest time**, **age**, **minimum**

temperature in OR, **bypass time**, and **average diastolic blood pressure**.
Only overlapping with the *real* expert variables defined in Section 4.3 with **age**, **minimum temperature in OR**, and **bypass time**.

Upon analyzing the performance results presented in Table 5.2, several noteworthy observations can be made. Firstly, the S-CLASSY model utilizing expert knowledge demonstrates remarkably similar performance to the CLASSY model across all metrics. This suggests that both models effectively learn and capture the patterns in the data, even though they employ different variables for their rule lists in each iteration.
However, a notable contrast is observed when considering the S-CLASSY model with *simulated* knowledge. This particular variant exhibits noticeably lower performance across all metrics, accompanied by higher loss values compared to the other models. This indicates that the *simulated* knowledge approach employed by S-CLASSY fails to effectively learn from the data and adequately capture the essential patterns, especially for the *windowing plain* data. Considering the findings of the authors of S-CLASSY, the *simulated* knowledge performs worse than selecting preferred variables randomly, which contrasts with their previous results.
By examining the rule lists generated by the three models, we gain further insights into the reasons behind these performance results. This discrepancy in performance can be attributed to the limitations of the *simulated* knowledge approach, highlighting the importance of genuine expert knowledge in achieving better performance, especially given the hard dataset to learn. See the generated rule lists in Listing A.1 for S-CLASSY with *real* expert knowledge, Listing A.2 for S-CLASSY with *simulated* expert knowledge, and Listing A.3 for CLASSY, all in the Appendix section.
Finally, when comparing the decision tree model to the rule lists, we observe comparable results in the AUPRC and F1 metrics. However, it shows much greater loss and overfitting values than the rest.

5.3.2 Preferred variables frequency

We study the frequency of preferred variables in the first rule of each rule list, denoted as $f@1$. To gain an understanding of this influence, we calculate the average value of $f@1$ across all folds. In Table 5.3 we can see the obtained results for both 2-class and 5-class problems.

Notably, for both problems, we observe that the S-CLASSY model when solving the *plain* data, incorporates a preferred variable always once in the first rule. As those rule lists are generally short and with low number of conditions, we hypothesize that the algorithm is not able to find different combinations of variables, given the small number of training samples. The CLASSY model is almost not using the expert's variables.

Additionally, comparing the problems, we consistently observe AUROC values over 0.55 for the 2-class and values below 0.48 for the 5-class problem. Despite this discrepancy in learning capacities, both models present similar expert's variables adoption in their rules.

Problem	Metric	Model \Pre-processing	plain	windowing plain	windowing quantiles
2-class	AUPRC	S-CLASSY *real* kn.	0.654	0.649	0.628
		S-CLASSY *simulated* kn.	0.637	0.558	0.61
		CLASSY	0.616	0.65	0.642
	F@1	S-CLASSY *real* kn.	1	1.4	1.4
		S-CLASSY *simulated* kn.	1	1.8	1.6
		CLASSY *	0.2	0.6	0.6
5-class	AUPRC	S-CLASSY *real* kn.	0.459	0.427	0.408
		S-CLASSY *simulated* kn.	0.469	0.402	0.369
		CLASSY	0.47	0.428	0.42
	F@1	S-CLASSY *real* kn.	1	1.6	1.6
		S-CLASSY *simulated* kn.	1	2	2
		CLASSY *	0	0.8	1

Table 5.3: Human guidance results comparing S-CLASSY with *real* and *simulated* expert knowledge, and CLASSY. We display the results for the 2-class and 5-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU. Further, in the case of S-CLASSY *real* knowledge and CLASSY, we count the average use of the preferred variables in the first rule, using the list of the *real* expert's variables. For the case of S-CLASSY with *simulated* knowledge, we use the list of preferred variables using the feature importance feature of a random forest model trained on all the data.

5.4 Interpretability

In order to achieve interpretability, we adhere to the widely accepted notion that smaller models are easier to comprehend (Doshi-Velez & Kim, 2017). Accordingly, we evaluate the number of rules and the number of conditions per rule. In addition, we assess the accumulated rule usage $\sum usg$ in the rules different than the default rule.

The first crucial criterion is the compactness of the rule list, which entails having a relatively small number of rules with a manageable number of conditions in each rule. To quantify compactness, we consider two major metrics: the average rule length $(\mu|r|)$ and the average number of rules in the rule list $(\mu|R|)$. Additionally, we aim to maximize the accumulated usage of the rules $\sum usg$, excluding the default rule. This metric provides insight into the level of support that the patterns shown in the rules receive from the training data.

Upon reviewing the results presented in Table A.5, a comparison is made between the interpretability metrics of the different models.

The number of conditions in a rule $(\mu|r|)$ is limited to four, which aligns with the observed results as all values are close to this threshold. Additionally, upon examining the generated rules, we observe how the models S-CLASSY and CLASSY employ notably more conditions for the datasets *windowing plain* and *windowing quantiles* than *plain* while keeping similar accuracies.
However, there is an exception, S-CLASSY with *simulated* knowledge is not capable of properly learning from the *windowing plain* data, having lower values for all metrics.
Furthermore, when looking at the depth of the decision tree, it is always four.

Problem	Metric	Model \Pre-processing	plain	windowing plain	windowing quantiles
2-class	AUPRC	S-CLASSY *real* kn.	0.654	0.649	0.628
		S-CLASSY *simulated* kn.	0.637	0.558	0.61
		CLASSY	0.616	0.65	0.642
		Decision Tree	0.643	0.617	0.615
	$\mu\|r\|$	S-CLASSY *real* kn.	1.983	2.901	3.088
		S-CLASSY *simulated* kn.	1.767	1.4	3.112
		CLASSY	1.883	2.857	3.002
		Decision Tree *	4.0	4.0	4.0
	$\mu\|R\|$	S-CLASSY *real* kn.	2.2	6.0	5.3
		S-CLASSY *simulated* kn.	2.2	1.0	5.6
		CLASSY	2.2	6.2	5.8
		Decision Tree *	13.5	15.2	15.6
	$\sum usg$	S-CLASSY *real* kn.	0.48	0.544	0.508
		S-CLASSY *simulated* kn.	0.456	0.163	0.477
		CLASSY	0.472	0.525	0.547
	runtime	S-CLASSY *real* kn.	7.248	17.33	18.77
		S-CLASSY *simulated* kn.	7.262	0.175	19.695
		CLASSY	7.589	18.53	21.021

Table 5.4: Interpretability results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the 2-class classification problem. Further, the number of conditions per rule and the number of rules in the list are compared with the average depth and average number of leaves in the tree, respectively. The $\sum usg$ value does not have a similarity to any parameter of a tree. We do not have the data for displaying the runtime of the tree.

In terms of the number of rules in the rule list ($\mu|R|$), the situation is similar to the previous metric, it depends on the dataset trained on, with the exception mentioned earlier for the case of S-CLASSY. In addition, the decision tree is always presenting a similar number of leaves, independently of the dataset, duplicating or triplicating the values of the rest of the models.

Furthermore, another way to confirm the anomaly in the S-CLASSY *simulated* knowledge trained on *windowing plain* is the metric $\sum usg$. Here, all models and configurations exhibit a $\sum usg$ value of around 50%, while the model not performing well has a $\sum usg$ value of 16%. This discrepancy shows that the expert rules used by S-CLASSY can have a negative effect on its performance.

Finally, when taking a look at the runtimes, we observe how consistently S-CLASSY is faster than CLASSY while obtaining similar accuracies. We hypothesize that this is due to the pattern search times. S-CLASSY, using expert's knowledge, is directed towards variables that potentially create patterns with high usage, while CLASSY must search for those variables. However, this advantageous feature can also be a drawback, as the algorithm may encounter challenges when provided with suboptimal expert variables, leading to convergence on local minima and resulting in poorer performance.

In summary, the results suggest that both S-CLASSY and CLASSY exhibit comparable levels of interpretability, as indicated by the average rule length and the number of rules, while the decision tree depicts much higher values for those metrics. In addition, S-CLASSY shows

sensitivity to the preferred variables utilized and outperforms CLASSY in terms of runtime efficiency.

5.5 Expert's feedback

In this section, our attention shifts to the analysis of the obtained rules in collaboration with medical doctors. In a prior iteration, as described for the clinical problem in Section 4.2, we collectively defined the problem to solve. Now, in a second iteration, we evaluate the benefits and constraints of this solution concerning their specific problem and offer new ideas for improvement.

First, they emphasize the importance of interpretability and adaptability for clinicians. They consider the current state of the algorithm highly valuable due to its immediate interpretability. However, the doctor needs a quick and easy-to-understand forecast for the LoS, so they insist on keeping the variables simple. This includes avoiding complex transformations and sticking to familiar variable features that are easy to express. Additionally, they prefer not to combine different variables referring to the same measurement in a single rule, e.g., the average and the maximum of a variable in the same rule.

Second, the obtained uncertainty values per rule are satisfactory for clinicians. The rules in the rule list accurately represent a relevant percentage of the training data, together with probability estimates that are generally close to zero or one.

Third, we emphasize the importance of the usage value, indicating how much of the training data is activated by the rule. However, it is a difficult metric to understand, and they would prefer not to communicate it. In the end, the minimum support threshold m_s is defined by this parameter, which is already sufficient.

Fourth, they express confidence that this model could be valuable for benchmarking across hospitals. For instance, when trained in the data of multiple hospitals, and comparing the rules, the differences between hospitals could also be assessed, allowing patients to choose the hospital that best suits their specific case and needs.

Fifth, upon examining a CLASSY rule that started with multiple engineered variables not immediately recognizable, they felt distrustful about it. However, when analyzing one of the S-CLASSY rule lists, they grasped the rule's process. For instance, when they interpreted a low systolic blood pressure average and high max temperature variable, they understood that the patient had hypotension and fever, potentially leading to complications. They emphasized the need for the variables in the rules to be expressed in a language they can readily understand.

Sixth, we raise a concern considering the rule *list* method rather than a rule *set*. We grasp that doctors potentially would not read the rules that come before the activated rule in their specific case. A more intuitive shape for the rules could improve readability.

Seventh, for this specific task, the values of several variables are not interpretable unless the variable age is considered simultaneously. Normal values for blood pressure, for example, differ between newborns and teenagers. Thus, rules that do not account for age, but are conditioned on blood pressure, are not trusted by medical doctors.

Overall, they suggest immediate future work to improve the process and make the rule lists more medically relevant. Without modifying the algorithm, they propose retraining the model using data from newborn patients only, to solve the previous limitation mentioned. Additionally, they recommend simplifying the variables by using the method *windowing plain*, as in previous experiments, but excluding the resulting variables related to quantiles distributions, and respiration rates. They note that the respiration rate value lacks general intuitiveness, as it does not indicate whether the patient is intubated or not.

CHAPTER

6

DISCUSSION

In this chapter, we delve into reflection on our research. Initially, our focus is centred on the method we devised for preparing the data, specially tailored to address the Length of Stay problem in hospitals.
Subsequently, we engage in an in-depth examination of the strengths and limitations of employing the S-CLASSY algorithm to tackle the Length of Stay problem, guided by the insights obtained after our experiments.
Lastly, we embrace the opportunity to discuss the new research line that emerges as a result of our work.

6.1 Predicting the Length of Stay

The Length of Stay proves challenging to be predicted in a medical context. In the literature, multiple techniques for feature selection and extraction have been explored, such as the approach of Alghatani et al. (2021). Additionally, multiple ad-hoc algorithms have been developed, like the temporal convolutions introduced by Rocheteau et al. (2021). However, achieving satisfactory performance has proven to be challenging, particularly when using only 2-class classifiers, resulting in performances that are far from excellent (Gupta et al., 2022). In this thesis, we face the additional difficulty of working with a small dataset focused exclusively on heart surgeries and limited to a two-year period.

In conventional supervised learning algorithms, the success of identifying patterns in data relies on the assumption that datasets are relatively devoid of noise. Nevertheless, the healthcare domain presents a distinctive scenario where multiple factors come into play, impacting the discharge decision-making process, and rendering the identification and categorization of noise a complex endeavour. Factors like the involvement of various medical team members, parental anxiety for a swift discharge, or even the well-being of the attending surgeon all contribute to the intricate and variable nature of the data under consideration.

Throughout this study, we have engaged in a series of iterative discussions and collaborations with the medical team, aiming to gain a comprehensive understanding of the data and the specific requirements of our research. A central focus has been placed on achieving an interpretable solution, necessitating a preliminary interpretation of the clinical messages inherent in the data. To achieve this, careful pre-processing and data selection have been undertaken, tailored to optimize the adaptation of our algorithms to the targeted problem.

In our pursuit of interpretability, we have sought to emulate medical approaches during the data pre-processing phase. This has involved techniques such as windowing the data to capture relevant temporal patterns, simulating the cognitive processes of nurses to better comprehend their decision-making rationale, and implementing the Quantiles Approach to strategically eliminate data points situated in close proximity to the median.

We have successfully refined our process by streamlining the variables . This has been a crucial step towards achieving the utmost interpretability of our algorithm. Nonetheless, we

acknowledge that there remains substantial room for further enhancement, continuing the iterative refinement process in close collaboration with the clinical experts. We seek to bridge the gap between advanced algorithmic methodologies and the domain-specific knowledge held by medical practitioners.

6.2 S-CLASSY

In this study, our research approach entails conducting numerous experiments to comprehensively evaluate and analyze the performance of S-CLASSY, incorporating both quantitative and qualitative experiments.

In evaluating the performance of S-CLASSY, we deliberately select models with comparable complexity, avoiding the use of overly simplistic models like predicting the majority class or high-complexity task-specific temporal neural networks. This careful consideration ensures a fair comparison, given the challenging nature of the dataset.

The findings suggest that the effect of human guidance may differ based on the quality of the preferred variables given.
The lower performance of S-CLASSY when using *simulated* expert knowledge compared to *real* expert knowledge prompts further analysis. As mentioned in the S-CLASSY paper, the algorithm exhibited similar performance when using the top variables defined by the feature importance of Random Forest (RF) and randomly selected variables. However, we observe notable discrepancies in performance when comparing the models using *real* and *simulated* expert knowledge in our experiments. We hypothesise that the dataset's complexity plays a relevant role in the algorithm's ability to learn effectively. That is, the *simulated* expert knowledge may not fully capture the nuances and intricacies present in the actual clinical domain, leading to suboptimal variable selections.
We observe an example where S-CLASSY is the only model with poor performance when guided with *simulated* expert knowledge. The preferred variables to use have to be carefully chosen, and adapted to the data and the rest of the variables, so the algorithm does not converge on local minima.
On the other hand, CLASSY is capable of discovering the relevant variables through its search process without the explicit guidance of preferred variables.
This motivates us to continue exploring the optimal integration of *real* expert knowledge in our predictions.

The effect of human guidance can still be enhanced by refining the overall selection of variables to use, aiming to focus on those that are more interpretable and clinically relevant. For instance, we should steer away from using metrics such as the standard deviation of blood pressure, which can be challenging to interpret in isolation. Instead, if the variables used in the training of the models directly relate to specific medical symptoms or conditions both performance and interpretability can increase.
Examples of such variables include the maximum temperature, which can indicate the presence of fever, and the minimum blood pressure, which may suggest hypotension. By focusing on these medically relevant variables, we enhance the interpretability of our results and ensure that our findings align more closely with the clinical domain.

Consequently, a trade-off has to be found when deciding the variables for the algorithm. Sophisticated techniques for data pre-processing, like *windowing quantiles*, often offer a better-denoised dataset for the algorithms to find accurate rules. In addition, for variables that might not be transparent, using transformations of the data like calculating the quartiles or the standard deviation of a variable can help the algorithms find good patterns in the data.
However, these variables are not useful for S-CLASSY for two reasons. First, the preferred variables have to be able to generalize across the data samples. When using very specific variables, especially in the subgroup of preferred ones, the algorithm is directed towards local minima. Second, these variables are not clear or usable by the clinicians. As a result, a trade-off about the simplicity of the variables has to be studied.

An essential consideration in our analysis is the interpretability of the rules. We find that approximately 54% of the examples in the binary problem dataset can be effectively delimited by the generated rules, indicating that S-CLASSY performs well in terms of the accumulated usage of the discovered rules.
Nevertheless, it is important to acknowledge that the remaining data is weakly explained. This observation underscores the need for further refinement in the approach. To address this, we recognize the relevance of incorporating additional variables that are better suited to the problem. Moreover, optimizing the algorithm's parameters could be crucial in enhancing its ability to extract meaningful and interpretable rules from the data.

The resulting rule lists provide a clear and transparent representation of the algorithm's decision-making journey, providing a good trade-off between classification performance and rule list sizes. It outlines how S-CLASSY selects specific populations based on different patterns observed in the data and subsequently determines the appropriate outcome for each group. This transparency and traceability of the algorithm's reasoning substantially contribute to its interpretability, making it easier for medical professionals to understand and trust the results.

However, it is essential to acknowledge that not all generated rules may align with medical criteria or exhibit logical coherence as an ensemble. The rapid understanding of the rules by doctors, and their ability to discern whether a rule aligns with medical criteria or not, serves as a crucial validation of the interpretability and applicability of the algorithm's outputs.
Moreover, when rules do not make sense following medical criteria, it serves as an opportunity for further refinement and optimization of the algorithm. By integrating the medical team's insights and feedback, we can iteratively improve the rule generation process to ensure that the rules derived by S-CLASSY are not only coherent but also medically meaningful.

6.3 New line of research

In the medical field, decision trees based on medical literature are used to assess the potential presence of complications following specific conditions. However, in our work, we advance beyond this level of model complexity. By exploring more sophisticated models, we seek to present a different perspective in interpreting the data, thereby enabling a deeper understanding of the underlying patterns and factors that influence patient outcomes.

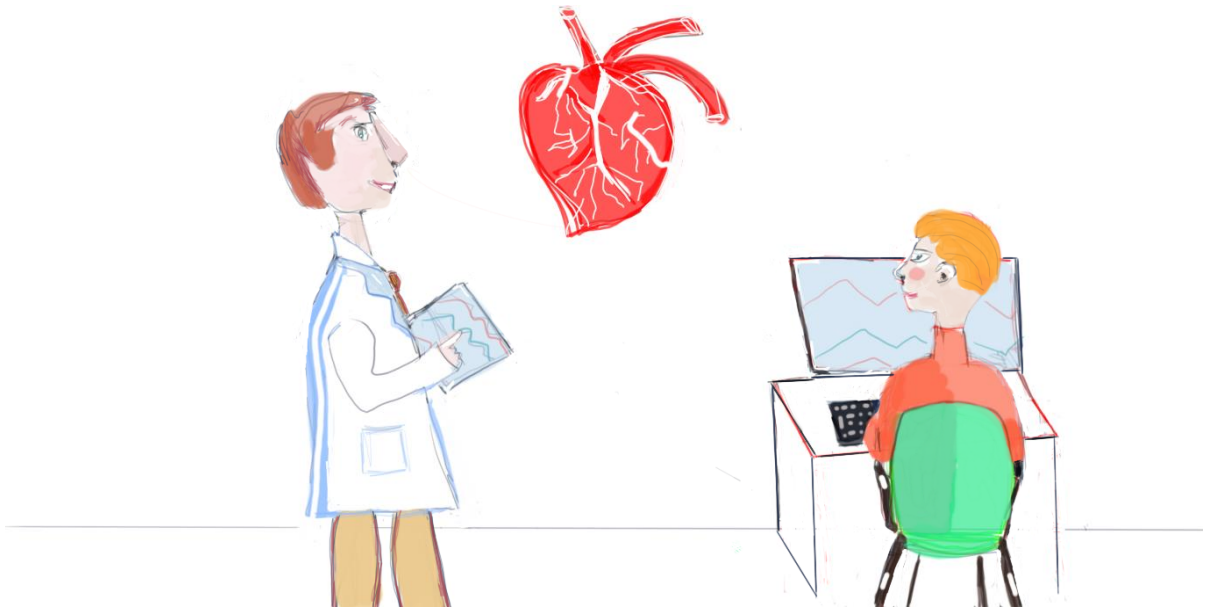
The reception of the S-CLASSY algorithm by medical professionals is one of cautious optimism, recognizing its potential as a novel way of explaining complex medical data. However, the integration of algorithms in the medical field is a gradual and meticulous process, demanding a robust and reliable approach at every stage.
One of the primary challenges lies in the interpretability of the rules generated by S-CLASSY. Medical professionals find it difficult to comprehend the selections of population patterns defined by the algorithm, as they do not correspond to any recognizable medical condition. This lack of alignment with existing medical knowledge hinders the immediate applicability of the algorithm's outputs in a clinical setting.
Nevertheless, it is essential to recognize that this is a common hurdle when incorporating algorithms in medicine. Iterative refinement and collaboration with medical experts are critical to overcoming this challenge. The insights and feedback provided by doctors play a crucial role in guiding the algorithm's development towards generating more medically relevant and interpretable rules.

The ultimate objective is to develop a predictor that follows a similar process as the clinicians. Therefore, we view rule lists as a suitable approach since they, like doctors, follow a sequential order to discard patterns (symptoms leading to diseases). In contrast, rule sets would require a more intricate representation of patterns.
Additionally, to align with clinicians' practices, the variables used in the algorithm need to be adapted to match those employed by clinicians to identify symptoms and determine outcomes. Ultimately, by generating patterns that hold medical relevance, clinicians can place their trust in the model's outcomes. The combination of variables in the activated rule would present a pattern that mirrors a specific symptom they recognize.
By fulfilling these conditions, clinicians could readily employ the algorithm without requiring any prior training.

CHAPTER

7

CONCLUSION



7.1 Conclusion

The objective of our research is proposing a novel method to solve the Length of Stay at Sant Joan de Déu in Barcelona. we perform different experiments and interviews with the medical doctors, and gather valuable insights into the specific needs and requirements of the clinicians regarding predicting the LoS.

In this study, we present a system that prioritizes two critical aspects in predictive solutions for the healthcare domain: accuracy and interpretability. S-CLASSY was designed to be interpretable, enabling healthcare professionals to comprehend and trust the reasoning behind the predictions. Additionally, we ensure that the model aligns with the clinicians' thinking process by incorporating clinically relevant variables for their assessments.

In our experiments, we validate the comparable accuracy achieved by the S-CLASSY algorithm in comparison to other models of similar complexity. Subsequently, we emphasize the model's ability to incorporate human guidance. Through the preferred variables feature of S-CLASSY, employing the *real* expert knowledge, we confirm the benefits of incorporating human guidance into the algorithm. Later, we evaluate the interpretability of S-CLASSY and CLASSY, and we introduce the *accumulated rule usage* metric to complement our analysis, showing how S-CLASSY covers more than half of the data with specific patterns and high accuracies. Finally, our discussions with the professionals encompass various aspects, confirming the interpretability of the studied model and proposing options for further development.

To sum up, the S-CLASSY algorithm shows promise as a novel way to explain complex medical data. However, integrating algorithms into the medical field is a gradual process, and interpretability remains a challenge. Collaborative refinement and feedback from doctors are essential for developing more medically relevant and interpretable rules in the future.

7.2 Limitations and future work

The curse of dimensionality. To ensure the interpretability of the rules, we aimed to keep the features straightforward and easy to comprehend. However, this approach has its limitations, as it can hinder the algorithm's performance. We recognize that employing more aggressive feature extraction techniques or dimensionality reduction methods could potentially enhance the accuracy of the models. Additionally, we used 24-hour time slots for feature extraction. However, future research could explore windowing the data in different hour intervals and with overlapping, and assess whether this leads to improved accuracies.

Class imbalance. In a typical setup, we strive to avoid class imbalance during training to ensure equal representation and prevent bias in the results. However, in the healthcare domain, the class imbalance is inherent due to a (fortunate) larger proportion of patients experiencing fast recoveries compared to those who develop complications. Moreover, our solution allows for experimentation with different target outcomes, incorporating varying the Length of Stay

time-frames. This adaptability enables us to align with the specific needs of the hospital and capture data patterns, as demonstrated by the comparison between the 3-class A and B dataset versions. Even with a mere one-day shift, the algorithm discovers more effective patterns for explaining the data.

Default rule of S-CLASSY. In general, across our results around half of the data is covered by rules with specific patterns with high accuracy. In this domain, prioritizing patterns that exhibit high confidence in their predictions proves beneficial, instead of prioritizing the usage of these patterns.
However, the other half of the data is explained by the default rule. In the cases of 3-class and 5-class, the default rule generally predicts the majority class, which is reasonable given that the preceding rules have focused on excluding the other outcomes. Nevertheless, we acknowledge a limitation where the default rule occasionally exhibits equal likelihood among all classes. For instance, in the 3-class example discussed in Section 5.2.2 the probabilities are 0.44, 0.26, and 0.30 with only a slight bias towards the first class.

Generalization over multiple datasets. In our study, we worked with a single dataset. Within this data, we created various versions of it by segregating the ICU from the ward, employing different discretization methods for the outputs, and using diverse data pre-processing techniques. However, relying solely on one dataset poses challenges in terms of generalization. By gathering data from multiple hospitals, we can assess the generalizability of our approach across different settings. This could then serve as a benchmarking measure, allowing for a comparison of the strengths and weaknesses of various procedures employed by different hospitals.

Medical decision trees. Decision trees predicting the LoS exist in the medical literature. A comparison between the decision tree trained on SJD and the tree proposed in the literature could bring more insights into the need for tailored variables for each problem. Perhaps the integration of preferred variables defined in the literature as the optimal ones could enhance the generalization of the algorithm.

Preferred patterns. The utilization of the preferred variables feature in S-CLASSY has resulted in substantial improvements in the task at hand. Despite these advancements, complete alignment with medical criteria has not been achieved yet. In light of this, a promising avenue for future research lies in defining patterns that are directly related to symptoms and illnesses. By incorporating these medically relevant patterns, we can guide the search process in a manner that aligns with the knowledge and trust of medical doctors, enhancing its interpretability.

Additional variables. Several aspects of a patient's stay are not captured by any variable in our dataset. For instance, medical doctors caution against utilizing the *respiration rate* variable, even if our algorithm identifies accurate patterns associated with it. The reason for this exclusion is that the normal values of the respiration rate depend on whether the patient is intubated or not, and we lack access to this specific variable in our dataset.

Cross-validation We employ standard 5-fold cross-validation. The *windowing plain* and *windowing quantiles* datasets utilize data windowing, generating multiple data samples from a single hospital stay which could be present both in the training and testing data.
However, it is important to note that this situation only results in a data leak for a maximum of two hospital stays when the test set is obtained from data in the centre of the dataset. Specifically, for folds two, three and four the first and last samples of the test dataset could be part of a hospital stay shared with samples in the training data. For folds one and five, this situation could occur only once.

7.3 Acknowledgements

This work was performed using the compute resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by Leiden University.

We extend our gratitude to the artist Anna Sibila for her contribution in creating the diverse artwork showcased throughout the thesis, including the cover page and the first pages of each chapter.

BIBLIOGRAPHY

Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pp. 559–560, 2018.

Khalid Alghatani, Nariman Ammar, Abdelmounaam Rezgui, and Arash Shaban-Nejad. Predicting intensive care unit length of stay and mortality using patient vital signs: Machine learning model development and validation. *JMIR Med Inform*, 9(5):e21347, May 2021. ISSN 2291-9694. doi: 10.2196/21347. URL https://medinform.jmir.org/2021/5/e21347.

John OR Aoga, Tias Guns, Siegfried Nijssen, and Pierre Schaus. Finding probabilistic rule lists using the minimum description length principle. In *Discovery Science: 21st International Conference, DS 2018, Limassol, Cyprus, October 29–31, 2018, Proceedings 21*, pp. 66–82. Springer, 2018.

Ine Borghans, Sophia M Kleefstra, Rudolf B Kool, and Gert P. Westert. Is the length of stay in hospital correlated with patient satisfaction? *International Journal for Quality in Health Care*, 24(5):443–451, 07 2012. ISSN 1353-4505. doi: 10.1093/intqhc/mzs037. URL https://doi.org/10.1093/intqhc/mzs037.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 6085, April 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-24271-9. URL https://europepmc.org/articles/PMC5904216.

William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pp. 115–123. Elsevier, 1995.

Gyzelly Alves de Carvalho, AA Rezende, Geovane Rossone Reis, and Giulliano Gardenghi. Use of the apache iv score as a predictor of mortality and length of stay in an intensive care unit. *J Physiother Res*, 10(1):9–15, 2020.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

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

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. *Science robotics*, 4(37):eaay7120, 2019.

Mehak Gupta, Brennan Gallamoza, Nicolas Cutrona, Pranjal Dhakal, Raphael Poulain, and Rahmatollah Beheshti. An Extensive Data Processing Pipeline for MIMIC-IV. In *Proceedings of the 2nd Machine Learning for Health symposium*, volume 193 of *Proceedings of Machine Learning Research*, pp. 311–325. PMLR, 28 Nov 2022. URL `https://proceedings.mlr.press/v193/gupta22a.html`.

Jens Hühn and Eyke Hüllermeier. Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19:293–319, 2009.

Dale A Huntley, Dong Won Cho, Jane Christman, and John G Csernansky. Predicting length of stay in an acute psychiatric hospital. *Psychiatric services*, 49(8):1049–1053, 1998.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimic-iv/1.0/(accessed August 23, 2021)*, 2020.

Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. The mimic code repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, 25(1):32–39, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684, 2016.

Ardeshir Mansouri, Mohammadreza Noei, and Mohammad Saniee Abadeh. Predicting hospital length of stay of neonates admitted to the nicu using data mining techniques. In *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 629–635. IEEE, 2020.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Mohammadreza Nemati, Jamal Ansary, and Nazafarin Nemati. Machine-learning approaches in covid-19 survival analysis and discharge-time likelihood prediction using clinical data. *Patterns*, 1(5):100074, 2020.

Ioanna Papagianni and Matthijs van Leeuwen. Discovering rule lists with preferred variables. In *International Symposium on Intelligent Data Analysis*, pp. 340–352. Springer, 2023.

Hugo M Proença and Matthijs van Leeuwen. Interpretable multiclass classification by mdl-based rule lists. *Information Sciences*, 512:1372–1393, 2020.

Hugo M Proença, Peter Grünwald, Thomas Bäck, and Matthijs van Leeuwen. Robust subgroup discovery: Discovering subgroup lists using mdl. *Data Mining and Knowledge Discovery*, 36 (5):1885–1970, 2022.

Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019. doi: 10.1056/NEJMra1814259. URL https://www.nejm.org/doi/full/10.1056/NEJMra1814259. PMID: 30943338.

Emma Rocheteau, Pietro Liò, and Stephanie Hyland. Temporal pointwise convolutional networks for length of stay prediction in the intensive care unit. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, pp. 58–68, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383592. doi: 10.1145/3450439.3451860. URL https://doi.org/10.1145/3450439.3451860.

Thomas Rotter, Joachim Kugler, Rainer Koch, Holger Gothe, Sabine Twork, Jeroen M van Oostrum, and Ewout W Steyerberg. A systematic review and meta-analysis of the effects of clinical pathways on length of stay, hospital costs and patient outcomes. *BMC health services research*, 8:1–15, 2008.

Sarah E Seaton, Lisa Barker, David Jenkins, Elizabeth S Draper, Keith R Abrams, and Bradley N Manktelow. What factors predict length of stay in a neonatal unit: a systematic review. *BMJ Open*, 6(10), 2016. ISSN 2044-6055. doi: 10.1136/bmjopen-2015-010466. URL https://bmjopen.bmj.com/content/6/10/e010466.

Kieran Stone, Reyer Zwiggelaar, Phil Jones, and Neil Mac Parthaláin. A systematic review of the prediction of hospital length of stay: Towards a unified framework. *PLOS Digital Health*, 1(4):e0000017, 2022.

Ilona WM Verburg, Nicolette F de Keizer, Evert de Jonge, and Niels Peek. Comparison of regression methods for modeling intensive care length of stay. *PloS one*, 9(10):e109684, 2014.

Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 222–235, 2020.

Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, pp. 563–574. Springer, 2019.

Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable bayesian rule lists. In *International conference on machine learning*, pp. 3921–3930. PMLR, 2017.

Lincen Yang and Matthijs van Leeuwen. Truly unordered probabilistic rule sets for multi-class classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 87–103. Springer, 2022.

Kinley Zangmo and Bodin Khwannimit. Validating the apache iv score in predicting length of stay in the intensive care unit among patients with sepsis. *Scientific Reports*, 13(1):5899, 2023.

Tahmina Zebin, Shahadate Rezvy, and Thierry J Chaussalet. A deep learning approach for length of stay prediction in clinical settings from medical records. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–5. IEEE, 2019.

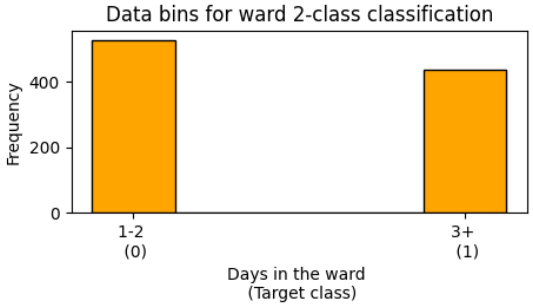
Guangyi Zhang and Aristides Gionis. Diverse rule sets. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1532–1541, 2020.

Ruohan Zhang, Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in leveraging human guidance for sequential decision-making tasks. *Autonomous Agents and Multi-Agent Systems*, 35(2):31, 2021.

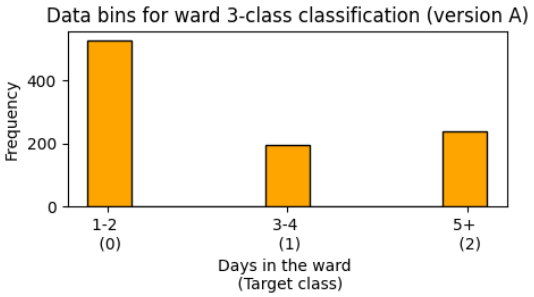
A

APPENDIX

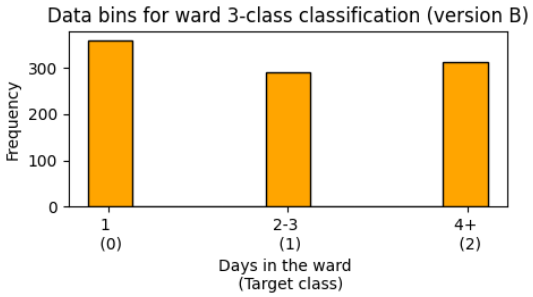
A.1 Experimentation with the ward data



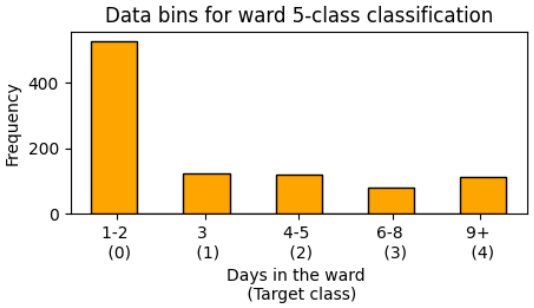
(a) Data binning the binary classification.



(b) Data binning for 3-class-A classification.



(c) Data binning for 3-class-B classification.



(d) Data binning for 5-class classification.

Figure A.1: Output data bins for the ward at SJD when windowing every 24 hours.

A.2 Performance evaluation

A.2.1 S-CLASSY, RF, MLP, and GB

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
2-class	AUROC	S-CLASSY	0.719	0.674	0.657
		RF	0.88	0.751	0.746
		MLP	0.845	0.687	0.694
		GB	0.848	0.732	0.73
	AUPRC	S-CLASSY	0.654	0.649	0.628
		RF	0.856	0.731	0.738
		MLP	0.843	0.684	0.698
		GB	0.826	0.719	0.73
	F1	S-CLASSY	0.669	0.667	0.635
		RF	0.726	0.7	0.698
		MLP	0.745	0.654	0.672
		GB	0.785	0.696	0.681
	log loss	S-CLASSY	0.6	0.668	0.671
		RF	0.466	0.566	0.565
		MLP	1.017	1.005	0.849
		GB	1.053	0.684	0.717
	overfit	S-CLASSY	0.117	0.153	0.163
		RF	0.114	0.136	0.148
		MLP	0.155	0.227	0.197
		GB	0.152	0.268	0.27
5-class	AUROC	S-CLASSY	0.687	0.629	0.62
		RF	0.809	0.702	0.698
		MLP	0.747	0.665	0.676
		GB	0.799	0.684	0.69
	AUPRC	S-CLASSY	0.459	0.427	0.408
		RF	0.671	0.524	0.516
		MLP	0.591	0.474	0.468
		GB	0.664	0.495	0.497
	F1	S-CLASSY	0.452	0.397	0.394
		RF	0.473	0.405	0.401
		MLP	0.554	0.443	0.405
		GB	0.547	0.431	0.443
	log loss	S-CLASSY	1.317	1.668	1.992
		RF	1.001	1.223	1.223
		MLP	2.348	2.44	1.77
		GB	3.067	2.128	2.14
	overfit	S-CLASSY	0.048	0.153	0.153
		RF	0.182	0.19	0.2
		MLP	0.239	0.211	0.179
		GB	0.201	0.316	0.31

Table A.1: Performance results comparing S-CLASSY to the non-interpretable models. We display the results for the 2-class and 5-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU. For the binary problem, all models seem to learn from the data, even when having low accuracy values. However, for the 5-class problem, almost all models and configurations struggle to learn, with high loss and AUPRC values often below 0.5.

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
3-A-class	AUROC	S-CLASSY	0.657	0.582	0.574
		RF	0.843	0.734	0.725
		MLP	0.749	0.668	0.692
		GB	0.833	0.729	0.713
	AUPRC	S-CLASSY	0.573	0.533	0.521
		RF	0.79	0.677	0.673
		MLP	0.72	0.628	0.63
		GB	0.789	0.674	0.664
	F1	S-CLASSY	0.616	0.515	0.556
		RF	0.624	0.573	0.581
		MLP	0.624	0.577	0.556
		GB	0.685	0.601	0.595
	log loss	S-CLASSY	1.117	1.399	1.363
		RF	0.752	0.848	0.845
		MLP	1.683	1.273	1.163
		GB	1.71	1.049	1.112
	overfit	S-CLASSY	0.045	0.034	0.032
		RF	0.148	0.156	0.17
		MLP	0.235	0.193	0.156
		GB	0.167	0.271	0.287
3-B-class	AUROC	S-CLASSY	0.455	0.508	0.497
		RF	0.755	0.713	0.704
		MLP	0.686	0.679	0.592
		GB	0.713	0.728	0.719
	AUPRC	S-CLASSY	0.396	0.459	0.438
		RF	0.682	0.619	0.609
		MLP	0.625	0.586	0.494
		GB	0.642	0.626	0.621
	F1	S-CLASSY	0.437	0.495	0.452
		RF	0.525	0.506	0.51
		MLP	0.503	0.532	0.397
		GB	0.507	0.567	0.553
	log loss	S-CLASSY	1.561	1.497	1.54
		RF	0.869	0.913	0.918
		MLP	1.601	1.153	1.114
		GB	2.154	1.052	1.094
	overfit	S-CLASSY	0.039	0.023	0.003
		RF	0.232	0.157	0.171
		MLP	0.276	0.168	0.157
		GB	0.287	0.272	0.281

Table A.2: Performance results comparing S-CLASSY to non-interpretable models. We display the results for the two 3-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU.

A.2.2 S-CLASSY, CLASSY, and decision tree

Problem	Metric	Model \Pre-processing	plain	windowing plain	windowing quantiles
2-class	AUROC	S-CLASSY real kn.	0.719	0.674	0.657
		S-CLASSY simulated kn.	0.732	0.55	0.626
		CLASSY	0.711	0.681	0.669
		Decision Tree	0.676	0.627	0.623
	AUPRC	S-CLASSY real kn.	0.654	0.649	0.628
		S-CLASSY simulated kn.	0.637	0.558	0.61
		CLASSY	0.616	0.65	0.642
		Decision Tree	0.643	0.617	0.615
	F1	S-CLASSY real kn.	0.669	0.667	0.635
		S-CLASSY simulated kn.	0.688	0.474	0.616
		CLASSY	0.649	0.686	0.641
		Decision Tree	0.705	0.612	0.628
	log loss	S-CLASSY real kn.	0.6	0.668	0.671
		S-CLASSY simulated kn.	0.6	0.695	0.716
		CLASSY	0.671	0.638	0.66
		Decision Tree	3.927	1.114	1.146
	overfit	S-CLASSY real kn.	0.117	0.153	0.163
		S-CLASSY simulated kn.	0.098	0.063	0.192
		CLASSY	0.125	0.147	0.16
		Decision Tree	0.285	0.216	0.225
5-class	AUROC	S-CLASSY real kn.	0.687	0.629	0.62
		S-CLASSY simulated kn.	0.69	0.605	0.567
		CLASSY	0.694	0.631	0.644
		Decision Tree	0.678	0.599	0.602
	AUPRC	S-CLASSY real kn.	0.459	0.427	0.408
		S-CLASSY simulated kn.	0.469	0.402	0.369
		CLASSY	0.47	0.428	0.42
		Decision Tree	0.506	0.414	0.402
	F1	S-CLASSY real kn.	0.452	0.397	0.394
		S-CLASSY simulated kn.	0.46	0.408	0.388
		CLASSY	0.447	0.4	0.417
		Decision Tree	0.504	0.395	0.408
	log loss	S-CLASSY real kn.	1.317	1.668	1.992
		S-CLASSY simulated kn.	1.34	1.432	1.687
		CLASSY	1.336	1.662	1.924
		Decision Tree	7.457	3.656	3.552
	overfit	S-CLASSY real kn.	0.048	0.153	0.153
		S-CLASSY simulated kn.	0.012	0.019	0.061
		CLASSY	0.008	0.154	0.133
		Decision Tree	0.244	0.213	0.206

Table A.3: Performance results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the 2-class and 5-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU.

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
3-A-class	AUROC	S-CLASSY *real* kn.	0.657	0.582	0.574
		S-CLASSY *simulated* kn.	0.604	0.493	0.526
		CLASSY	0.624	0.582	0.553
		Decision Tree	0.675	0.578	0.608
	AUPRC	S-CLASSY *real* kn.	0.573	0.533	0.521
		S-CLASSY *simulated* kn.	0.549	0.463	0.481
		CLASSY	0.566	0.522	0.512
		Decision Tree	0.608	0.54	0.541
	F1	S-CLASSY *real* kn.	0.616	0.515	0.556
		S-CLASSY *simulated* kn.	0.58	0.413	0.51
		CLASSY	0.569	0.492	0.558
		Decision Tree	0.56	0.518	0.514
	log loss	S-CLASSY *real* kn.	1.117	1.399	1.363
		S-CLASSY *simulated* kn.	1.21	1.25	1.247
		CLASSY	1.166	1.294	1.307
		Decision Tree	4.995	1.682	1.535
	overfit	S-CLASSY *real* kn.	0.045	0.034	0.032
		S-CLASSY *simulated* kn.	0.063	0.023	0.032
		CLASSY	0.052	0.035	0.055
		Decision Tree	0.259	0.241	0.215
3-B-class	AUROC	S-CLASSY *real* kn.	0.455	0.508	0.497
		S-CLASSY *simulated* kn.	0.422	0.495	0.492
		CLASSY	0.422	0.502	0.504
		Decision Tree	0.634	0.626	0.606
	AUPRC	S-CLASSY *real* kn.	0.396	0.459	0.438
		S-CLASSY *simulated* kn.	0.375	0.42	0.421
		CLASSY	0.375	0.455	0.438
		Decision Tree	0.521	0.514	0.499
	F1	S-CLASSY *real* kn.	0.437	0.495	0.452
		S-CLASSY *simulated* kn.	0.413	0.228	0.393
		CLASSY	0.413	0.483	0.449
		Decision Tree	0.482	0.489	0.434
	log loss	S-CLASSY *real* kn.	1.561	1.497	1.54
		S-CLASSY *simulated* kn.	1.465	1.257	1.34
		CLASSY	1.465	1.423	1.535
		Decision Tree	4.602	1.689	1.383
	overfit	S-CLASSY *real* kn.	0.039	0.023	0.003
		S-CLASSY *simulated* kn.	0.074	0.006	0.002
		CLASSY	0.074	0.031	0.004
		Decision Tree	0.257	0.18	0.194

Table A.4: Performance results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the two 3-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU.

A.3 Interpretability evaluation

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
2-class	AUPRC	S-CLASSY real kn.	0.654	0.649	0.628
		S-CLASSY simulated kn.	0.637	0.558	0.61
		CLASSY	0.616	0.65	0.642
		Decision Tree	0.643	0.617	0.615
	$\mu\|r\|$	S-CLASSY real kn.	1.983	2.901	3.088
		S-CLASSY simulated kn.	1.767	1.4	3.112
		CLASSY	1.883	2.857	3.002
		Decision Tree *	4.0	4.0	4.0
	$\mu\|R\|$	S-CLASSY real kn.	2.2	6.0	5.3
		S-CLASSY simulated kn.	2.2	1.0	5.6
		CLASSY	2.2	6.2	5.8
		Decision Tree *	13.5	15.2	15.6
	$\sum usg$	S-CLASSY real kn.	0.48	0.544	0.508
		S-CLASSY simulated kn.	0.456	0.163	0.477
		CLASSY	0.472	0.525	0.547
	runtime	S-CLASSY real kn.	7.248	17.33	18.77
		S-CLASSY simulated kn.	7.262	0.175	19.695
		CLASSY	7.589	18.53	21.021
5-class	AUPRC	S-CLASSY real kn.	0.459	0.427	0.408
		S-CLASSY simulated kn.	0.469	0.402	0.369
		CLASSY	0.47	0.428	0.42
		Decision Tree	0.506	0.414	0.402
	$\mu\|r\|$	S-CLASSY real kn.	2.1	2.985	3.262
		S-CLASSY simulated kn.	1.967	2.444	2.509
		CLASSY	2.1	2.981	3.175
		Decision Tree *	4.0	4.0	4.0
	$\mu\|R\|$	S-CLASSY real kn.	2.0	9.0	8.2
		S-CLASSY simulated kn.	2.4	4.2	4.4
		CLASSY	2.4	9.2	8.2
		Decision Tree *	15.6	15.8	15.8
	$\sum usg$	S-CLASSY real kn.	0.341	0.54	0.53
		S-CLASSY simulated kn.	0.358	0.269	0.271
		CLASSY	0.366	0.571	0.583
	runtime	S-CLASSY real kn.	8.078	32.106	34.309
		S-CLASSY simulated kn.	9.102	13.352	16.661
		CLASSY	9.691	33.742	35.331

Table A.5: Interpretability results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the 2-class and 5-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU. Further, for the Decision Tree, the number of conditions per rule and the number of rules in the list are compared with the average depth and average number of leaves in the tree, respectively. The $\sum usg$ value does not have a similarity to any parameter of a tree. We do not have the data for displaying the runtime of the tree.

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
3-A-class	AUPRC	S-CLASSY *real* kn.	0.573	0.533	0.521
		S-CLASSY *simulated* kn.	0.549	0.463	0.481
		CLASSY	0.566	0.522	0.512
		Decision Tree	0.608	0.54	0.541
	$\mu\|r\|$	S-CLASSY *real* kn.	2.45	2.865	3.136
		S-CLASSY *simulated* kn.	1.933	2.0	2.502
		CLASSY	1.95	2.826	3.092
		Decision Tree *	4.0	4.0	4.0
	$\mu\|R\|$	S-CLASSY *real* kn.	2.6	8.0	6.9
		S-CLASSY *simulated* kn.	2.4	1.0	3.3
		CLASSY	2.3	8.0	8.0
		Decision Tree *	14.9	15.3	15.6
	$\sum usg$	S-CLASSY *real* kn.	0.405	0.632	0.569
		S-CLASSY *simulated* kn.	0.467	0.14	0.328
		CLASSY	0.428	0.574	0.522
	runtime	S-CLASSY *real* kn.	8.646	24.851	26.578
		S-CLASSY *simulated* kn.	8.125	0.178	10.479
		CLASSY	8.374	25.731	31.054
3-B-class	AUPRC	S-CLASSY *real* kn.	0.396	0.459	0.438
		S-CLASSY *simulated* kn.	0.375	0.42	0.421
		CLASSY	0.375	0.455	0.438
		Decision Tree	0.521	0.514	0.499
	$\mu\|r\|$	S-CLASSY *real* kn.	1.817	2.882	3.107
		S-CLASSY *simulated* kn.	1.533	2.0	2.369
		CLASSY	1.533	2.801	2.932
		Decision Tree *	4.0	4.0	4.0
	$\mu\|R\|$	S-CLASSY *real* kn.	2.1	6.9	6.2
		S-CLASSY *simulated* kn.	2.1	1.0	3.1
		CLASSY	2.1	7.0	6.3
		Decision Tree *	12.6	15.8	15.6
	$\sum usg$	S-CLASSY *real* kn.	0.471	0.597	0.554
		S-CLASSY *simulated* kn.	0.439	0.105	0.319
		CLASSY	0.439	0.599	0.524
	runtime	S-CLASSY *real* kn.	7.919	22.44	24.785
		S-CLASSY *simulated* kn.	7.608	0.185	10.224
		CLASSY	8.256	23.038	25.508

Table A.6: Interpretability results comparing S-CLASSY with *real* and *simulated* expert knowledge to interpretable models. We display the results for the two 3-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU. Further, for the Decision Tree, the number of conditions per rule and the number of rules in the list are compared with the average depth and average number of leaves in the tree, respectively. The $\sum usg$ value does not have a similarity to any parameter of a tree. We do not have the data for displaying the runtime of the tree.

A.4 Human guidance evaluation

Problem	**Metric**	**Model \Pre-processing**	plain	windowing plain	windowing quantiles
3-A-class	AUPRC	S-CLASSY real kn.	0.573	0.533	0.521
		S-CLASSY simulated kn.	0.549	0.463	0.481
		CLASSY	0.566	0.522	0.512
	F@1	S-CLASSY real kn.	1	1.8	1.4
		S-CLASSY simulated kn.	1.2	2	2
		CLASSY *	0	0.8	0.6
3-B-class	AUPRC	S-CLASSY real kn.	0.396	0.459	0.438
		S-CLASSY simulated kn.	0.375	0.42	0.421
		CLASSY	0.375	0.455	0.438
	F@1	S-CLASSY real kn.	1	1.8	1.8
		S-CLASSY simulated kn.	1	2	2
		CLASSY *	0	0.6	0.6

Table A.7: Human guidance results comparing S-CLASSY with *real* and *simulated* expert knowledge, and CLASSY. We display the results for the two 3-class classification problems, with different distributions in the discretization method to decide the target LoS in the ICU. Further, in the case of S-CLASSY *real* knowledge and CLASSY, we count the average use of the preferred variables in the first rule, using the list of the *real* expert's variables. For the case of S-CLASSY with *simulated* knowledge, we use the list of preferred variables using the feature importance method of a random forest model trained on all the data.

A.4.1 AUROC and AUPRC curves

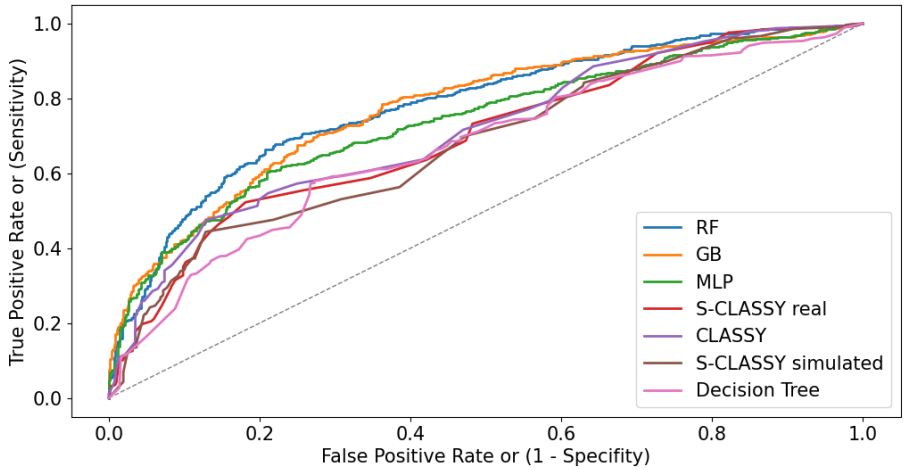


Figure A.2: AUROC plot for all models for the 2-class *windowing plain* problem.

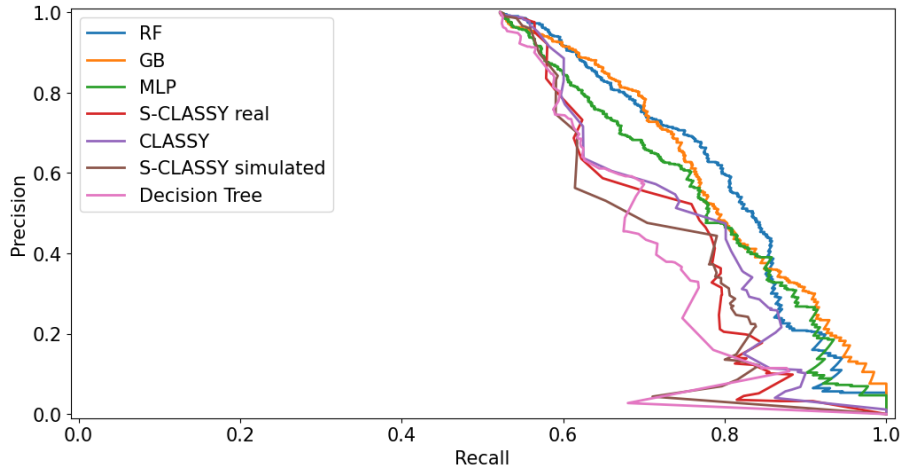


Figure A.3: AUPRC plot for all models for the 2-class *windowing plain* problem.

A.5 Rule list examples

```
// Preferred variables: ['age_months', 'surgery duration',
//       'clamp time', 'bypass time', 'min. temperature in OR',
//       'avg-Heart rate', 'avg-SpO2']
// Rule list: (970 training samples)
IF min. temperature in OR < 30.7
        AND min-Systolic blood pressure < 60.0
        AND q3-Diastolic blood pressure < 50.0 THEN usage = 89;
//  Pr(1-2days) = 0.0 Pr(3+days) = 1.0

ELSE IF surgery duration >= 396.0
        AND 36.35357142857143 <= avg-Temperature < 36.75
        AND q1-Diastolic blood pressure < 56.0 THEN usage = 53;
//  Pr(1-2days) = 0.0 Pr(3+days) = 1.0

ELSE IF min-Systolic blood pressure < 74.0
        AND min-Respiration rate >= 23.0
        AND min-SpO2 < 95.0 THEN usage = 144;
//  Pr(1-2days) = 0.188 Pr(3+days) = 0.812

ELSE IF min-Diastolic blood pressure < 44.0
        AND surgery duration >= 289.0
        AND gender >= 1.0 THEN usage = 96;
//  Pr(1-2days) = 0.25 Pr(3+days) = 0.75

ELSE IF max-Respiration rate >= 43.0
        AND min-SpO2 < 92.0 THEN usage = 212;
//  Pr(1-2days) = 0.524 Pr(3+days) = 0.476

ELSE usage = 376;
//  Pr(1-2days) = 0.801 Pr(3+days) = 0.199
```

Listing A.1: Rule list produced by S-CLASSY with expert's *real* knowledge for the 2-class *windowing plain* problem.

```
// Preferred variables:['arrest time', 'age_months', 'bypass time',
//      'avg-Systolic blood pressure','q1-Systolic blood pressure',
//      'min. temperature in OR',  'avg-Diastolic blood pressure']
// Rule list: (970 training samples)
IF age_months >= 25.0
        AND 31.0 <= bypass time < 58.0 THEN usage = 92;
//  Pr(1-2days) = 0.902 Pr(3+days) = 0.098

ELSE usage = 878;
//  Pr(1-2days) = 0.433 Pr(3+days) = 0.567
```

Listing A.2: Rule list produced by S-CLASSY with expert's *simulated* knowledge for the 2-class *windowing plain* problem.

```
// Rule list: (970 training samples)
IF q3-Diastolic blood pressure < 50.0
        AND min-Systolic blood pressure < 60.0
        AND min. temperature in OR < 30.7 THEN usage = 89;
//  Pr(1-2days) = 0.0 Pr(3+days) = 1.0

ELSE IF surgery duration >= 396.0
        AND 36.35357142857143 <= avg-Temperature < 36.75
        AND q1-Diastolic blood pressure < 56.0 THEN usage = 53;
//  Pr(1-2days) = 0.0 Pr(3+days) = 1.0

ELSE IF min-Systolic blood pressure < 74.0
        AND min-Respiration rate >= 23.0
        AND min-SpO2 < 95.0 THEN usage = 144;
//  Pr(1-2days) = 0.188 Pr(3+days) = 0.812

ELSE IF min-Diastolic blood pressure < 44.0
        AND surgery duration >= 289.0
        AND gender >= 1.0 THEN usage = 96;
//  Pr(1-2days) = 0.25 Pr(3+days) = 0.75

ELSE IF max-Respiration rate >= 43.0
        AND min-SpO2 < 92.0 THEN usage = 212;
//  Pr(1-2days) = 0.524 Pr(3+days) = 0.476

ELSE usage = 376;
//  Pr(1-2days) = 0.801 Pr(3+days) = 0.199
```

Listing A.3: Rule list produced by CLASSY for the 2-class *windowing plain* problem.

A.6 Variable importances

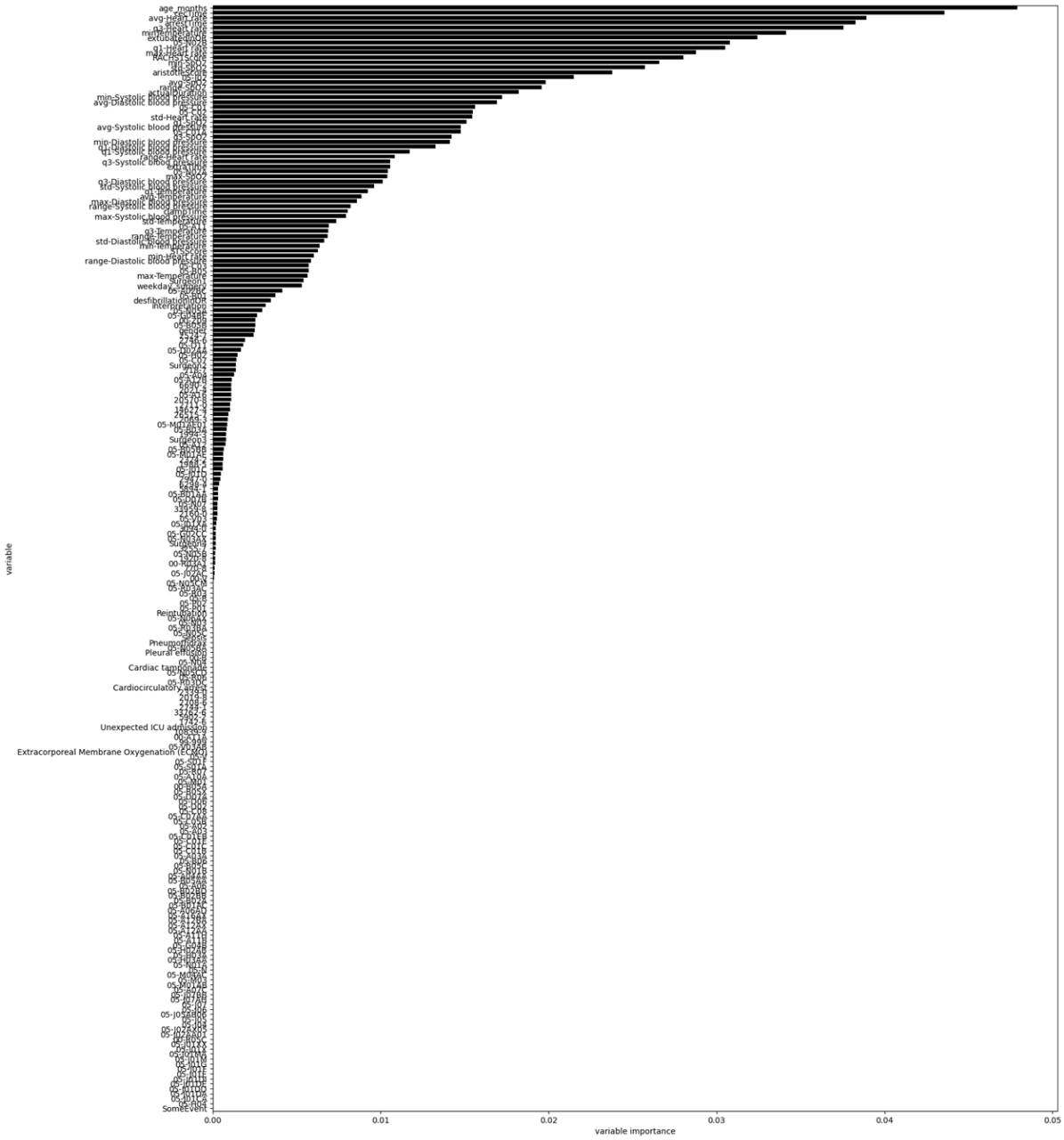


Figure A.4: All 162 features used to train the random forest algorithm, before performing data selection.

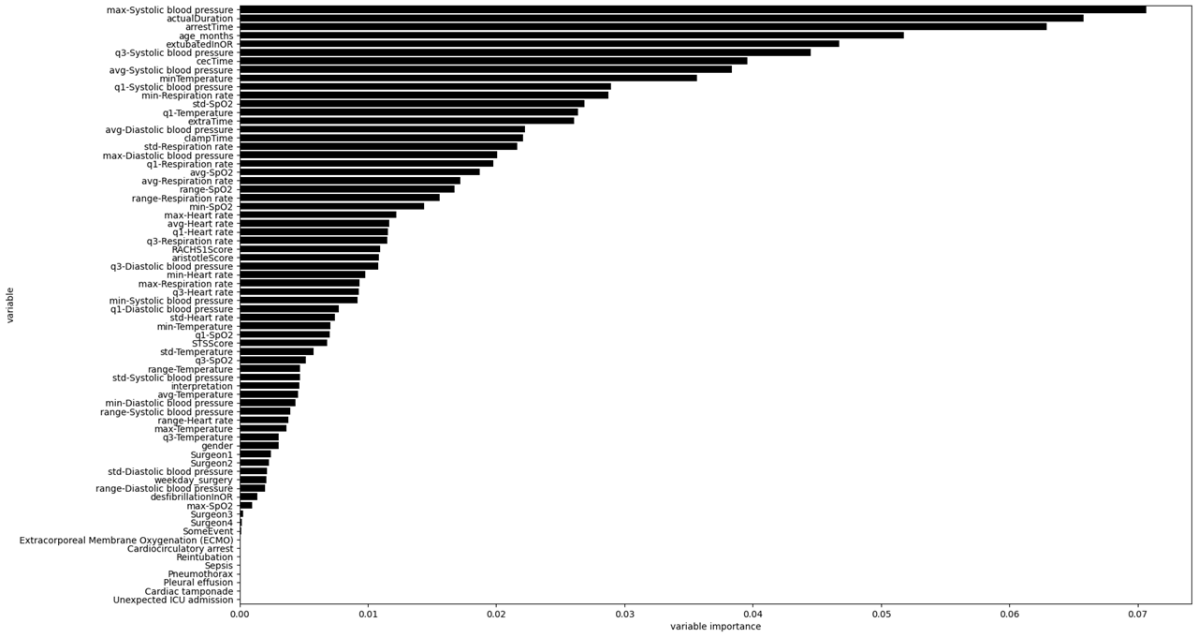


Figure A.5: The selected 70 features used to train the random forest algorithm, after performing data selection.

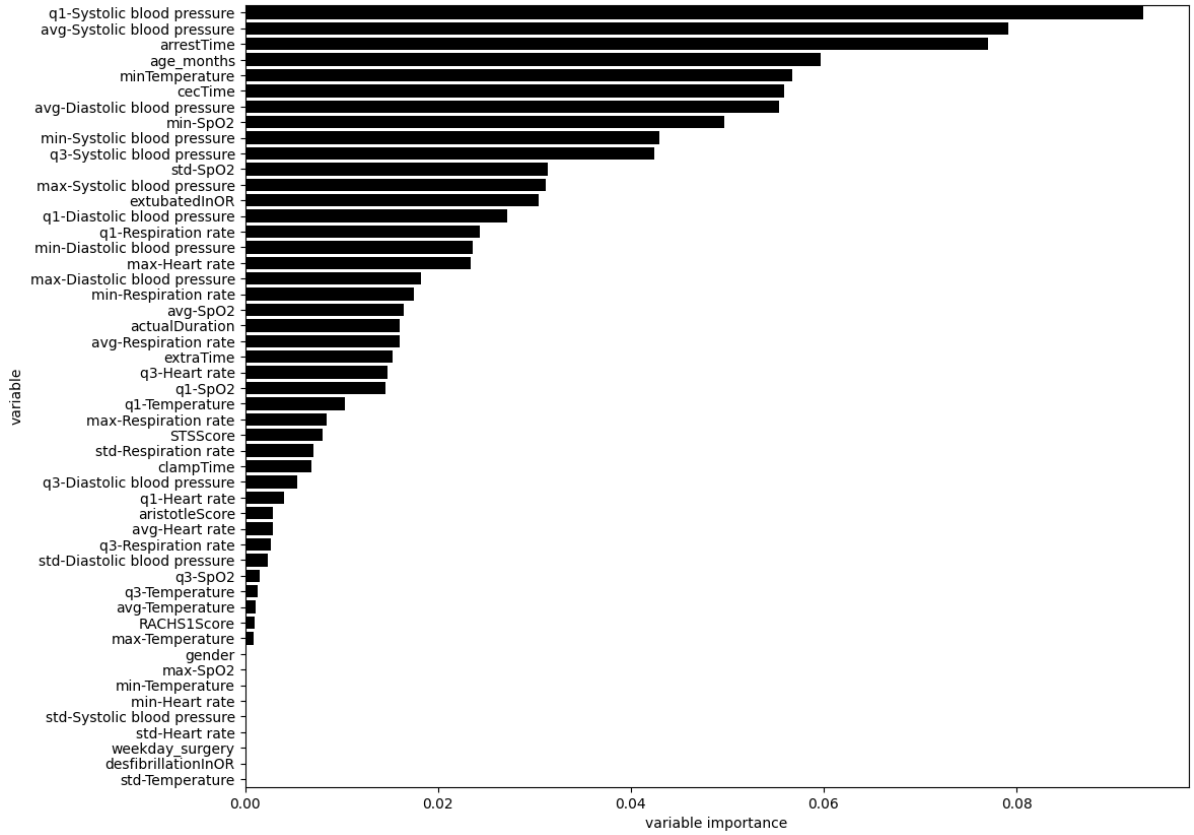


Figure A.6: The final 50 features used to train the random forest algorithm.

A.7 Decision tree plot

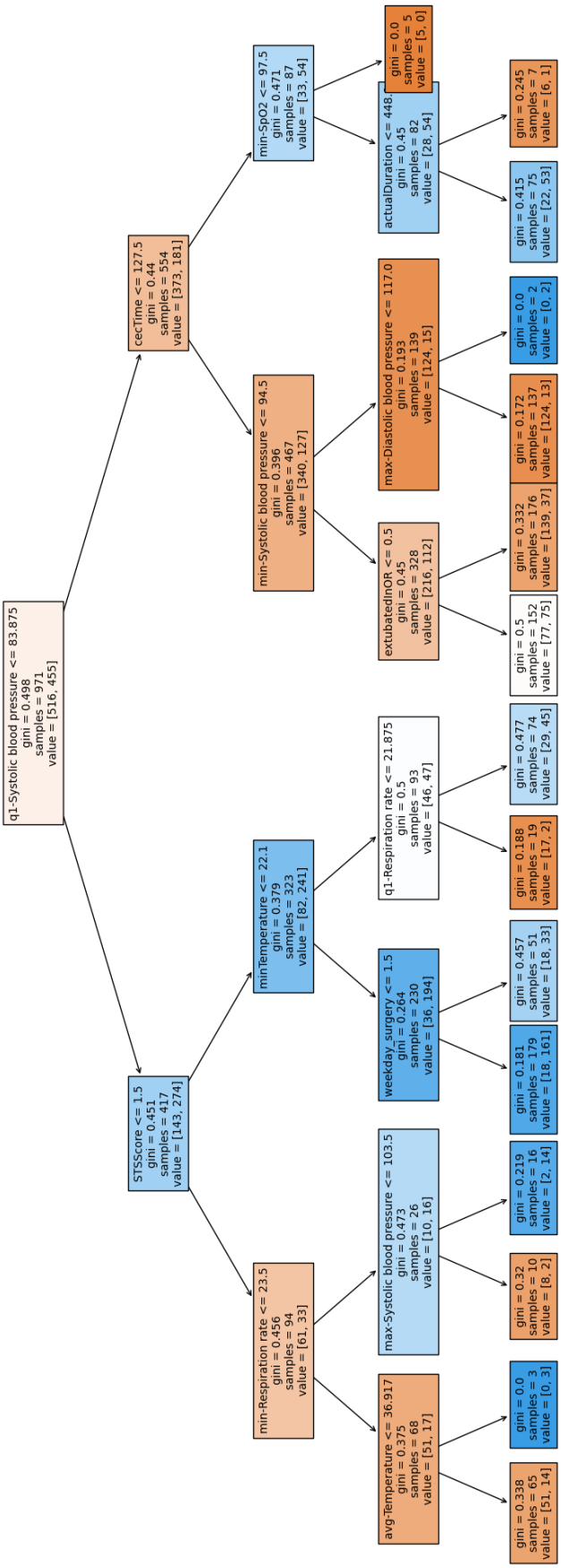


Figure A.7: Decision tree for the 2-class *windowing plain* problem.