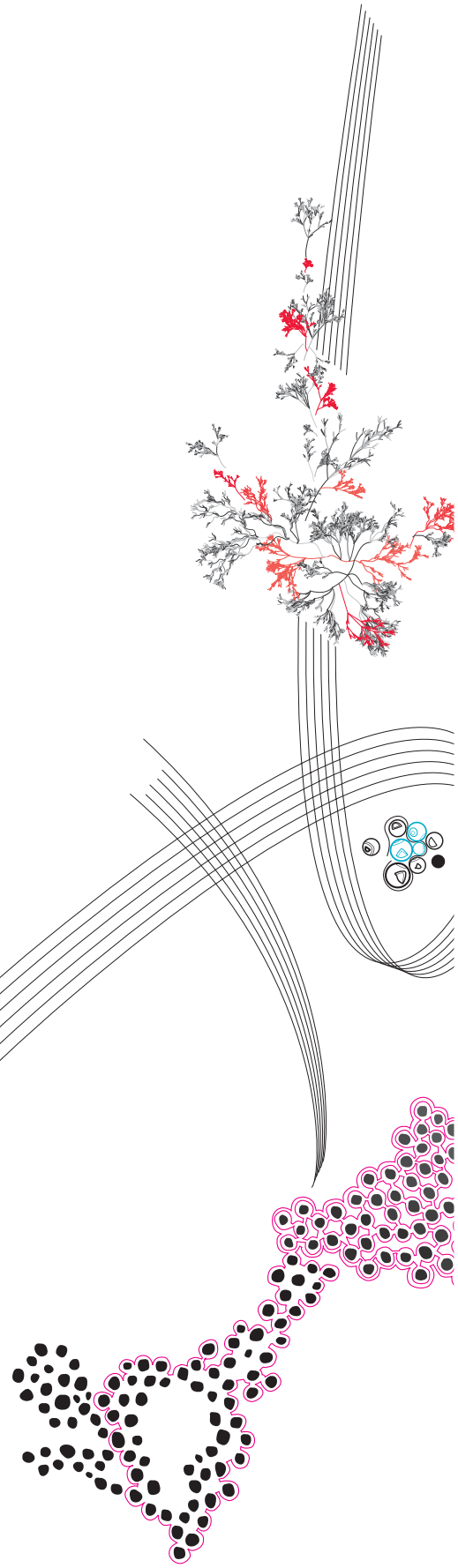


MSc Industrial Engineering
and Management
Final Project

# A novel hybrid machine learning metaheuristic approach to create nurse rosters in a Dutch hospital


Diederik Willem Quak

UT 1st Supervisor: Prof.Dr.Ir. M.R.K. Mes

UT 2nd Supervisor: Dr. D. Guericke

ORTEC B.V.1st Supervisor: Dr. D.D. Tönissen

ORTEC B.V.2nd Supervisor: Ir. S.A.I. Hassan


April, 2023


Department of Industrial Engineering and
Business Information Systems
Faculty of Behaviour Management and
Social sciences
University of Twente


UNIVERSITY OF TWENTE.

ORTEC

"*Plans are of little importance, but planning is essential.*"

Winston Churchill

# Acknowledgements

This thesis marks the end of my five and half year journey at the University of Twente. This project also represents the culmination of my time as a student. During this time, I have learned countless things and had the opportunity to partake in many great experiences.

First of all, I would like to thank ORTEC B.V. for providing me with this interesting and challenging graduation opportunity. I felt that the project was taken seriously, with many resources and side projects started for the overarching AI-assisted workforce project. Colleagues of ORTEC were always very helpful and eager to help with some challenges on the way. I am especially grateful to my supervisors, Denise and Shayekh, for their dedicated feedback and suggestions. The brainstorming sessions that we had together and the insightful feedback were crucial in shaping the direction of the thesis.

I would also like to extend my sincere appreciation to my supervisors, Martijn and Daniela, from the University of Twente, for their valuable feedback and suggestions. Their expertise and feedback were helpful in navigating the complexities of the graduation project. Their insightful feedback helped to clarify my writing and present my ideas more effectively.

Furthermore, the cooperating hospital was very helpful. This thesis was not possible without the consent to use their planning data. The interviews held at the hospital with the nurse planners were useful in getting a hands-on understanding of the planning process in a real-life setting. The answers to our questions were useful during the research project later on.

Lastly, I would like to thank my family and friends for their unconditional support and for helping me to achieve this academic and professional milestone.

*Diederik Quak*
*Leiden, March 2023*

# Management summary

**Purpose**: This MSc thesis project at ORTEC B.V. aims to develop a hybrid solution method for creating feasible schedules for a Dutch hospital. It combines Machine Learning (ML) with an improvement heuristic and creates schedules for nurses based on past schedule realisations. The purpose of the research is to find out how a combination of these ML techniques and improvement heuristics may aid human planners at the hospital.

**Problem definition**: The optimiser in the ORTEC Workforce Scheduler is used at almost no hospital in The Netherlands. Planning nurses in hospital environments is a complex task because of the challenging requirements of the nurses that are difficult to model mathematically. This project is part of the Artificial Intelligence assisted workforce planner project initiated by ORTEC to find novel ways to improve the performance of their optimiser by both understanding better what differentiates a good roster from a bad roster and implementing Artificial Intelligence techniques in their planning software products.

**Approach**: We constructed a schedule for a planning horizon by classifying with ML on every day in the prediction horizon what set of nurses is working based on past actual planning realisations. We selected five ML methods based on characteristics of the binary classification problem at hand: K-Nearest Neighbours, Logistic Regression, Artificial Neural Network, Random Forest and Gradient Boosting. A naive planning benchmark was designed to compare the value of the obtained ML results to a method that recycles the schedule from one year back. The performance was tested with a four-fold cross-validation method for three different departments in a hospital for a robust analysis of the results. Also, two different prediction methods were evaluated: the Perfect Information (PI) approach and the updating approach. Both the PI and the updating approach create a schedule for the prediction time horizon but differ in the way they do this. The PI approach uses the actually realised feature values to iteratively make a prediction one day ahead for every day in the planning horizon. The updating approach updates the selected features for the prediction based on the prediction made earlier by the prediction model. Three improvement heuristics were selected to ensure the feasibility while at the same time minimising the missing required nurses on a day and staying as close as possible to the ML schedule. Five important constraints were selected for this research that are often used in healthcare settings for labour rules of the nurses. The selected improvement heuristics are Simulated Annealing, Gradient Descent and a method that relaxes the hard constraints after a set number of iterations and repairs the schedule when this leads to an infeasible solution.

**Results**: Figure 1 shows the results for the best ML model for the PI and updating approach and compares them to the results of the naive method for the three selected departments. These results are based on the performance of the time horizon from 01-12-2021 to 28-02-2022. Among the improvement heuristic options, Gradient Descent with an alpha value of 0.999 yielded the best performance and was consequently selected for the comparison in Figure 1.

| | IC | | Radiology | | Haematology | |
|---|---|---|---|---|---|---|
| | F1-score | Constraints | F1-score | Constraints | F1-score | Constraints |
| Updating | 0.7 | 73 | 0.70 | 1465 | 0.62 | 18325 |
| After optimisation | 0.69 | 0 | 0.67 | 0 | 0.62 | 0 |
| Difference | -1% | -100% | -4% | -100% | 0% | -100% |
| | | | | | | |
| PI | 0.79 | 73 | 0.71 | 1668 | 0.7 | 2075 |
| After optimisation | 0.75 | 0 | 0.69 | 0 | 0.68 | 0 |
| Difference | -5% | -100% | -3% | -100% | -3% | -100% |
| | | | | | | |
| Naive method | 0.55 | 239 | 0.68 | 64 | 0.73 | 114 |
| After optimisation | 0.55 | 0 | 0.67 | 0 | 0.72 | 0 |
| Difference | 1% | -100% | -2% | 100% | -1% | -100% |

TABLE 1: Results machine learning before and after optimisation

**Conclusion**: The Machine Learning models performed better than the naive method in the IC and radiology departments, but the naive method performed better in the haematology department. The improvement heuristic led to a solution that did not contain any constraint violations. Although the improvement heuristic effectively minimized the daily missing capacity, achieving a schedule that closely resembled the predicted one posed a challenge.

**Discussion**: The performance of the Machine Learning models fluctuates over the departments and can give better results than a naive method when there are enough wishes to train on and schedules are not cyclic. The aim of this research is to show the capability of a hybrid method to create feasible schedules based on historical data. However, more research is required for the implementation of the research as an AI-assisted workforce scheduler tool.

Essential information was missing that could have improved the ML results, such as the skill sets of the nurses and their age. Also, the predicted schedule performance was evaluated based on the actual realised schedule. Since the actual schedule may not be the perfect schedule and there may be some interchangeability between the nurses, a more fair evaluation of the performance of the predicted schedules is required to understand their true feasibility. An idea to evaluate the true value of the predicted schedule is to present the schedules to the nurse planners and discuss the quality of the schedule.

Lastly, more research is required to improve the results of the ML models. ORTEC has started two projects, with one project focussing on understanding what qualities impact the roster quality, while another research investigates the prediction of shifts instead of what day a nurse is working as considered in this research. Both pieces of research may aid this research by better understanding what differentiates a good roster from a bad roster and predicting the shift on a certain day.

*Keywords*: Nurse Scheduling Problem, Machine Learning, hybrid methods, metaheuristics

# Contents


**1 Problem introduction** **14**
- 1.1 Introduction ORTEC B.V. . . . 14
- 1.2 ORTEC Workforce Scheduling . . . 14
- 1.3 Motivation research . . . 16
  - 1.3.1 Need for workforce scheduling improvements in healthcare . . . 16
  - 1.3.2 Nurse Scheduling Problem . . . 17
  - 1.3.3 Rostering process hospital X . . . 18
  - 1.3.4 AI-assisted workforce scheduler tool . . . 19
- 1.4 Identification of the research problem . . . 20

**2 Literature search** **24**
- 2.1 Hierarchical nurse scheduling management . . . 24
  - 2.1.1 Strategic level . . . 24
  - 2.1.2 Tactical level . . . 25
  - 2.1.3 Operational level . . . 25
- 2.2 Introduction NSP . . . 26
  - 2.2.1 Constrained solution space . . . 26
  - 2.2.2 Schedule objectives . . . 27
  - 2.2.3 Team rostering . . . 27
- 2.3 NSP solution methods . . . 28
  - 2.3.1 Mathematical model steps . . . 29
  - 2.3.2 Solution methodologies in literature . . . 30
  - 2.3.3 AI approaches for NSP in literature . . . 31
  - 2.3.4 Metaheuristic approaches in NSP literature . . . 32
- 2.4 Gap in literature . . . 33

**3 Data preparation** **34**
- 3.1 Data analysis . . . 34
  - 3.1.1 Data entities . . . 35
  - 3.1.2 Analysis data entities . . . 35
  - 3.1.3 Analysis employees . . . 36
  - 3.1.4 Analysis shifts . . . 37
  - 3.1.5 Analysis of wishes . . . 38
- 3.2 Feature engineering . . . 39
  - 3.2.1 Feature selection . . . 40
  - 3.2.2 Feature extraction . . . 40
  - 3.2.3 One-hot encoding . . . 42
- 3.3 Last pre-processing steps . . . 43
  - 3.3.1 Train-test-validation split . . . 43

3.3.2 Feature scaling . . . 44
3.4 Conclusion . . . 44

**4 Solution approach** **45**
4.1 Classification approach . . . 45
4.1.1 Classification format . . . 45
4.1.2 Scheduling approach and simplifications . . . 46
4.2 ML model selection process . . . 47
4.2.1 No Free Lunch . . . 47
4.2.2 Model considerations . . . 47
4.3 Model selection . . . 48
4.3.1 KNN . . . 49
4.3.2 Logistic regression . . . 50
4.3.3 Random Forest . . . 52
4.3.4 Gradient Boosting . . . 53
4.3.5 Neural Networks . . . 54
4.4 Performance testing . . . 55
4.4.1 Confusion matrix . . . 55
4.4.2 ROC curve . . . 56
4.4.3 Wish score . . . 57
4.4.4 Missing capacity . . . 57
4.4.5 Hard constraint violations . . . 57
4.5 Improvement heuristic . . . 58
4.5.1 Simulated Annealing framework . . . 58
4.5.2 Objective function . . . 59
4.5.3 Making the predicted schedule feasible . . . 60
4.5.4 Operators . . . 60
4.5.5 Improvement heuristic experimentation . . . 60
4.6 Conclusion . . . 61

**5 Results** **62**
5.1 Results ML . . . 62
5.1.1 F1-score . . . 62
5.1.2 SHAP value analysis . . . 66
5.1.3 Confusion Matrix . . . 67
5.1.4 ROC curve . . . 68
5.1.5 Wish score . . . 68
5.2 Results metaheuristics . . . 70
5.2.1 Metaheuristics results for IC PI . . . 70
5.2.2 Metaheuristics results for IC update . . . 71
5.2.3 Metaheuristics results for IC naive method . . . 72
5.3 Comparison before and after improvement heuristic . . . 73
5.3.1 IC . . . 74
5.3.2 Radiology . . . 74
5.3.3 Haematology . . . 75
5.4 Implementation of the results . . . 75

**6 Conclusion and discussion** **77**
6.1 Conclusion . . . 77
6.2 Discussion . . . 79
6.3 Recommendations . . . 81

**A Hyperparameter tuning** **87**
A.1 K-Nearest Neighbours . . . 87
A.2 Logistic Regression . . . 87
A.3 Gradient Boosting . . . 88
A.4 Random Forest . . . 89
A.5 Artificial Neural Network . . . 90

**B Feasible schedule algorithm** **92**

**C Confusion Matrices** **93**
C.1 Confusion Matrix IC PI . . . 93
C.2 Confusion Matrix IC updating . . . 94
C.3 Confusion Matrix radiology PI . . . 94
C.4 Confusion Matrix radiology updating . . . 95
C.5 Confusion Matrix haematology PI . . . 95
C.6 Confusion Matrix haematology updating . . . 96

**D SHAP value analysis** **97**
D.1 IC update . . . 97
D.2 Radiology PI . . . 98
D.3 Radiology update . . . 98
D.4 Haematology PI . . . 99
D.5 Haematology update . . . 99

**E Improvement heuristic results** **100**
E.1 Radiology PI . . . 100
E.2 Radiology update . . . 101
E.3 Haematology PI . . . 102
E.4 Haematology update . . . 103
E.5 Radiology naive . . . 104
E.6 Haematology naive . . . 105

# List of Figures


1.1 Organogram of ORTEC . . . 15
1.2 AI-assisted workforce schedule . . . 20
1.3 Research question design of this project . . . 23

2.1 Management structure of nurse scheduling . . . 26
2.2 Placement of team rostering from Silvestro and Silvestro (2000) . . . 28

3.1 Visualisation of the data approach steps . . . 34
3.2 Number of employees over time . . . 36
3.3 Number of shifts over time . . . 37
3.4 Relative number of shifts per time unit . . . 38
3.5 Distribution of the wishes in the database for the four selected departments. 39
3.6 Feature selection per data entity . . . 42
3.7 Rolling time horizon as cross validation . . . 44

4.1 Difference between PI and updating approach . . . 46
4.2 The KNN classification based on k=3 and k=6 . . . 50
4.3 Comparison between a linear regression model and a logistic function. . . . 51
4.4 Example of a Decision Tree for the NSP. . . . 53
4.5 Example of GB from "Gradient Boosting Machines" (2023) . . . 54
4.6 Example of ROC curves from Wikipedia contributors (n.d.). . . . 57
4.7 Flowchart of the SA approach. . . . 59

5.1 Min, max and average f1-score results per model and prediction type for the IC department . . . 63
5.2 Min, max and average f1-score results per model and prediction type for the radiology department . . . 64
5.3 Min, max and average f1-score results per model and prediction type for the haematology department . . . 65
5.4 SHAP values for GB for IC department with PI approach . . . 66
5.5 Confusion matrices for the best performing models . . . 67
5.6 ROC curves of the best-performing models . . . 68
5.7 Wish score comparison for the IC department . . . 69
5.8 Wish score comparison for the radiology department . . . 69
5.9 Wish score comparison for the haematology department . . . 70
5.10 Objective value IC department with PI approach . . . 71
5.11 Similarity score and missing capacity over time for IC department with PI approach . . . 71
5.12 Objective value IC department with updating approach . . . 72
5.13 Similarity score and missing capacity over time for IC department with updating approach . . . 72

5.14 Objective value IC department with naive method . . . . . 73
5.15 Similarity score and missing capacity over time for IC department with naive method . . . . . 73

C.1 Confusion Matrix IC PI . . . . . 93
C.2 Confusion Matrix IC updating . . . . . 94
C.3 Confusion Matrix radiology PI . . . . . 94
C.4 Confusion Matrix radiology updating . . . . . 95
C.5 Confusion Matrix haematology PI . . . . . 95
C.6 Confusion Matrix haematology PI . . . . . 96

D.1 SHAP value analysis IC update LR model . . . . . 97
D.2 SHAP value analysis Radiology PI LR model . . . . . 98
D.3 SHAP value analysis radiology update KNN model . . . . . 98
D.4 SHAP value analysis haematology PI RF model . . . . . 99
D.5 SHAP value analysis haematology update LR model . . . . . 99

E.1 Objective value radiology department with PI approach . . . . . 100
E.2 Similarity score and missing capacity over time for Radiology department with PI approach . . . . . 100
E.3 Objective value radiology department with updating approach . . . . . 101
E.4 Similarity score and missing capacity over time for Radiology department with update approach . . . . . 101
E.5 Objective value haematology department with PI approach . . . . . 102
E.6 Similarity score and missing capacity over time for Haematology department with PI approach . . . . . 102
E.7 Objective value haematology department with updating approach . . . . . 103
E.8 Similarity score and missing capacity over time for Haematology department with updating approach . . . . . 103
E.9 Similarity score and missing capacity over time for radiology department with naive method . . . . . 104
E.10 Similarity score radiology department with naive method . . . . . 104
E.11 Objective value haematology department with naive method . . . . . 105
E.12 similarity score aematology department with naive method . . . . . 105

# List of Tables


1 Results machine learning before and after optimisation . . . . . 5

3.1 Analysis data entities . . . . . 36
3.2 Analysis of employees for selected departments. . . . . 37

4.1 Selected hyperparameters models . . . . . 49
4.2 Example of Confusion matrix. . . . . 56

5.1 Comparison of f1-score for the best-performing ML models for the IC department. . . . . 64
5.2 Comparison of f1-score for the best-performing ML models for the radiology department. . . . . 65
5.3 Comparison of f1-score for the best-performing ML models for the haematology department. . . . . 66
5.4 Summary results IC department before and after improvement heuristic . . 74
5.5 Summary results IC department before and after improvement heuristic . . 75
5.6 Predicted schedule IC PI approach for the second week in the 1-12-2021 to 28-12-2022 prediction horizon . . . . . 76

6.1 Simplified scheduling example. . . . . 80

A.1 Hyperparameter tuning GB . . . . . 89
A.2 Hyperpameter tuning RF . . . . . 89
A.3 Hyperparameter tuning batch size and epochs . . . . . 90
A.4 Hyperparameter tuning optimiser . . . . . 90
A.5 Hyperparameter tuning drop out rate . . . . . 91
A.6 Hyperparameter neurons . . . . . 91

# List of acronyms

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **ANN** | Artificial Neural Network |
| **AUC** | Area Under the Curve |
| **CV** | Cross-validation |
| **DNW** | Duty No Work wish |
| **DW** | Duty Work wish |
| **FPR** | False Positive Rate |
| **GA** | Genetic Algorithm |
| **GB** | Gradient Boosting |
| **KNN** | K-Nearest Neighbours |
| **LR** | Logistic Regression |
| **ML** | Machine Learning |
| **NN** | Neural Network |
| **NP** | Non-deterministic Polynomial time |
| **NSP** | Nurse Scheduling Problem |
| **NWR** | No Work Required wish |
| **NWNR** | No Work Not Required |
| **OR** | Operations Research |
| **PI** | Perfect Information |
| **RF** | Random Forest |
| **ROC** | Receiver Operator Characteristic curve |
| **SA** | Simulated Annealing |
| **SHAP** | Shapely Additive exPlanations |
| **TPR** | True Positive Rate |
| **TSP** | Travelling Salesman Problem |
| **WNR** | Work Not Required wish |

# Glossary of terms

**Artificial Intelligence** are a group of techniques to perform tasks that typically require human intelligence, such as learning, problem-solving, decision-making, and language understanding.

**Duty** is a set of activities performed by the employees, such as performing blood tests without specifying the time and date that this has to be performed.

**Machine Learning** is a subset of Artificial Intelligence that involves the development of algorithms that are able to learn and adapt without requiring explicit instructions by using algorithms and statistics.

**Metaheuristic** is a technique to efficiently explore the solution space to find near-optimal solutions.

**Nurse Scheduling Problem** is a discipline in scheduling research that involves the assignment of shifts among nurses.

**Operations Research** is a subset of mathematics that provides a quantitative basis for managerial decisions in scheduling, logistics and more.

**OWS** is a software solution created by ORTEC B.V. that supports scheduling activities for their customers.

**Shift** is the realization of duties in the planning horizon with an assigned time window and assigned employees.

# Chapter 1

# Problem introduction

This research was conducted at ORTEC B.V. as part of the graduation project for the Master program Industrial Engineering and Management at the University of Twente. This first chapter gives an introduction to the problem that this research concerns.

Section 1.1 gives a short description of ORTEC and its history. This research took place at the ORTEC Workforce Scheduling department. An introduction to the functionalities and applications of this software is given in Section 1.2. Next, Section 1.3 motivates the reason for performing this research. Lastly, Section 1.4 identifies the research problem and provides the research question design to solve this research problem.

## 1.1 Introduction ORTEC B.V.

ORTEC was founded by a few young students in the 1980s to ensure long-term sustainable growth for companies and society by leveraging data and mathematics. Accordingly, the slogan of ORTEC is 'Optimise your world'. Currently, the company has more than 1000 employees that work in 18 offices around the world, with the headquarter being in Zoetermeer. The customer base is diverse, with their solutions applied in 11 different business areas in 14 types of industries (ORTEC, 2022). ORTEC has developed multiple products over the years structured according to the organogram in Figure 1.1. This thesis is conducted at the technology department at ORTEC. This department deals with the software development and support activities for the software solutions that are created by ORTEC. Although the focus of this research is on the Workforce product, this project has been assigned to the Math Innovation Team located in the overarching empowerment section of the Technology branch. This Math Innovation Team is a supporting team for the technology products and focuses on innovative ways to improve their products further. The managerial allocation of this project in the overall company structure is highlighted orange in Figure 1.1.

## 1.2 ORTEC Workforce Scheduling

ORTEC has created the ORTEC Workforce Scheduler (OWS) software to help customers with their workforce scheduling. OWS can support the employee scheduling process in organisations by a collection of software components. Users of OWS can integrate the software into their planning process with the help of an ORTEC specialist to meet their company-specific needs. Some examples of the functionalities of OWS are invoicing, illness registration, payroll processing, and connection to payroll administration.

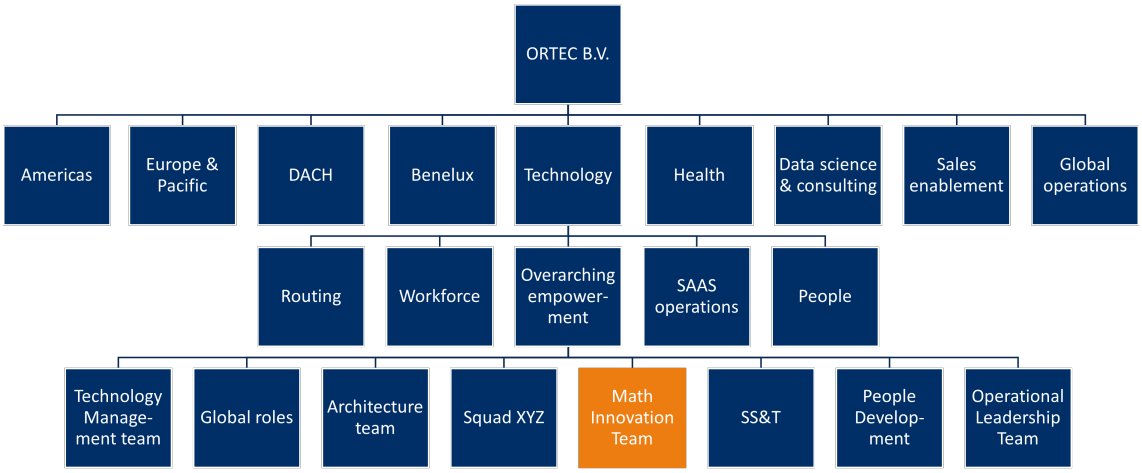



FIGURE 1.1: Organogram of ORTEC

The fundamental component of OWS is the workforce optimiser. Given the set of employees, shifts, constraints and all required planning information, the optimiser can generate shifts for the organisation while it tries to comply with the legislation and regulations as much as possible.

OWS is used in the following five sectors: healthcare, facility services, transport, warehouse and distribution centres and governments. While for most sectors, the software is used as intended, the schedules created by the OWS optimiser are either significantly changed or not used at all by the planners for nurse scheduling. The complex preferences that nurses may have can be challenging to implement in the scheduler; therefore, the optimiser's schedule may not adhere to most of the preferences. Still, the majority of hospitals in The Netherlands have bought the OWS license because of the supporting planning modules that help the administrators save time. Payroll calculation is not straightforward for nurses with variable shifts, but the OWS software can help to calculate this. Hospitals also use OWS to check if any labour rules are violated when making the schedule themselves or making quick changes to the schedule.

Laurens et al. (2006) describe the working of the optimiser tool in OWS. The objective function of the optimiser is to minimise the number of violations of preferences and labour rules. Criteria that should be met at all times are given a very high weight, while less essential criteria are given less weight. The optimiser consists of three phases. The first phase is a genetic algorithm that uses a combination of two cross-over operators and three mutation operators. When the optimiser cannot improve its solution value after a set number of generations, the optimiser moves on to the next phase. After the first phase has created a complete roster, the second phase tries to improve this. For this, the second phase uses local search methods. The local search phase consists of one-opt and two-opt methods and solves the schedule to one and two-opt optimality (Laurens et al., 2006). The last phase is entered when all the best schedules have been checked. This last phase involves a large neighbourhood search phase in which the schedule is changed considerably and stops when it can not improve locally anymore.

The algorithms designed for the optimiser are built as general-purpose planners because of the large number of organisations that make use of the software. Therefore, the optimiser has robust performance but has little knowledge of the specific planning problem at a certain organisation that uses the software. This has the advantage that it is easy to add

new constraints quickly, but the algorithm may perform inferior to custom-build algorithms that have adapted to the specific problem context (Laurens et al., 2006).

## 1.3 Motivation research

This section describes the need for optimal utilization of hospital workforce, workforce scheduling in general and the overarching plan of ORTEC to find possibilities for improvement of the OWS ("ORTEC", 2022). First Section 1.3.1 explains the need for improvements in workforce scheduling in healthcare. Then Section 1.3.2 gives a description of the Nurse Scheduling Problem, which is the name given to the optimisation of workforce scheduling of nurses in literature. Then, Section 1.3.3 describes the rostering process of a Dutch hospital that provided us with the data required for the research. This hospital is mentioned as hospital X from now on to ensure the anonymity of the hospital. Lastly, Section 1.3.4 presents the AI-assisted workforce scheduler project that this research is part of.

### 1.3.1 Need for workforce scheduling improvements in healthcare

Waiting for healthcare delivery is a significant problem for hospitals and has become an inherent characteristic of healthcare delivery in The Netherlands and worldwide (Fogarty & Cronin, 2008). The origin of waiting for healthcare is threefold, according to Hall (2012):

1. In most countries, quality healthcare access is seen as a right. Governments, therefore, try to reduce the cost of healthcare by subsidisation and regulated pricing. The government's interference in the pricing structure of healthcare diminishes the free market economic working to match supply with demand. Patients may request more care than they need because of the low incentives, or healthcare providers may be reluctant to expand capacity when reimbursement is less than the total expansion costs.

2. Most often, patients' expectation for service quality is not high since medical professionals are regarded as society's elite. Patients are happy that they can be helped by them in the first place and are inclined to accept the waiting times.

3. Lastly, waiting can also be caused by sub-optimal management and inefficiency of healthcare delivery. There can be simply a mismatch in demand and supply, but it can also be the case that healthcare resources are underutilised or that, for example, cleaners have not been scheduled adequately to make the hospital room available promptly. Improving the schedule-making for the nurses can increase the workforce's efficiency and utilisation, thereby reducing patient waiting times and costs.

In this research, we focus on improving the automatic schedule-making for the nurses, focusing primarily on this third aspect.

The demand for nurses in hospitals is outgrowing the supply worldwide for multiple reasons, such as the ageing population and the low job salary. This nurse shortage was already observed in early 2000s (Murray, 2002) but is still an issue today. There was an estimated nine million nurses shortage worldwide in 2014, and although this number is expected to decrease towards 2030, the shortage will still be seriously impacting the care processes around the world (Drennan & Ross, 2019). In combination with the poor utilisation of these nurses, lack of potential educators and sub-optimal allocation of the nurses to the available tasks, the optimisation of nurse scheduling is becoming even more

essential (Haddad et al., 2022). While the influx of nurses from developing countries partially addresses the demand-supply mismatch in highly developed countries, there remains a significant shortage of nurses (Kingma, 2018). For these reasons, the correct workforce scheduling of the limited pool of nurses is critical.

The nurse shortage is a continuing problem in many countries, including The Netherlands (Hall, 2012; Wright & Mahar, 2013). This shortage of skilled nurses has even caused an unfavourable effect on patient outcomes and increased mortality (Aiken et al., 2002). Combined with growing concerns about the nurse's competence to deliver quality patient care, the need for good allocation of the available workforce is substantial.

### 1.3.2 Nurse Scheduling Problem

Workforce scheduling can be defined as the process of assigning shifts to employees to meet the demand of the workplace. Traditionally, workforce scheduling aims to minimise personnel costs while achieving a certain required service level (Castillo et al., 2009). Workforce scheduling problems have been studied intensely during the last decades, most likely due to the economic importance of the optimisation of the allocation of employees to cut costs (Van den Bergh et al., 2013). Since the influential papers by Edie (1954) and Dantzig (1954), the workforce scheduling problem has been optimised in many business areas. The content of the workforce scheduling problem has shifted since the 1950s to a stronger focus on meeting the employees' needs and preferences when making the schedules instead of only cost minimisation (Petrovic & Vanden Berghe, 2012). Examples of the needs and preferences of the employees are a preference to work with befriended colleagues, a predilection for shift types, and maximum working hours (Van den Bergh et al., 2013).

The workforce scheduling of nurses in The Netherlands is notoriously difficult because of the complex preferences and requirements of the nurses and the labour rules. While some of the restrictions are straightforward to implement, such as not working more than the hours stated in the contract and only assigning workers to the tasks they are allowed to perform, some preferences are more difficult to take into account when making the nurse rosters such as a preference to work with a specific group of colleagues. Moreover, some preferences are not even known in advance, and the preference for a roster over another can only be decided by a human planner when the rosters are made because of the subtle nuances in the preferences for the rosters.

The optimisation of the planning of nurses in hospitals is known in the literature as Nurse Scheduling Problem (NSP). The NSP can be defined as the planning of the work activities of the nurses while considering the multiple and possible contradictory objectives, such as maximising the satisfaction of the nurses while distributing the workload in the fairest way (Legrain et al., 2014). The subtle and complex preferences are difficult to implement in the OWS optimiser software effectively, and the optimiser may not be able to deal with these preferences at all. Therefore, in practice, nurse scheduling is done by human planners (Legrain et al., 2014). Solving the NSP mathematically to optimality is more difficult than other hard combinatoric problems such as the Travelling Salesman Problem (Burke et al., 2010). Because of the large and complex planning environment in real life that is difficult to solve by an optimisation program, the planning process is done manually by skilled nurse planners who can better deal with the complex planning requirements (Turhan & Bilgen, 2020). The quality of the solution approaches in the literature is not satisfactory for hospital administrators because the solution approach is not advanced enough to deal with real-life problem-specific situations (Ernst et al., 2004). The solution approach can, for example, not deal with the intangible preferences that the nurses or nurse planners may have when the rosters are made.

Not long ago, most of the scheduling tasks were performed manually by the head nurse of the department for the majority of hospitals in The Netherlands. This method of making the nurse's schedule is time-consuming and challenging for the head nurses and leads to schedules that are inauspicious for the department's operation and the nurses' working conditions (Rönnberg et al., 2013). Because the nurse shortage is becoming a more prominent problem and the working conditions for the nurses impact the quality of healthcare service, the need to allow for a more flexible approach has increased.

An example of a new approach for a more flexible scheduling method is self-rostering, in which the nurses assign the required shifts to themselves (Asgeirsson, 2014; Rönnberg et al., 2013). This new form of scheduling may improve the working conditions of the nurses, but it is difficult to use in practice. Team rostering is a scheduling process that is a more centralised form of self-rostering in which the nurses indicate their preferences and what shifts they want to perform, but a nominated member of the team has the final responsibility for rostering in consultation with the team members (Burke et al., 2004). For more information about the different forms of nurse scheduling, the reader is referred to Section 2.2.3.

### 1.3.3 Rostering process hospital X

This thesis is in collaboration with hospital X, which allowed us to use the planning data for this research. We held interviews with hospital X to map the rostering process of hospital X which is the focus of this section. The team rostering method has been piloted at hospital X, with certain departments now using a combination of team rostering and final optimisation by a central planner. The team rostering is named self-rostering in the software of ORTEC, but is mentioned as team rostering in this report since this is the name used by hospital X and is most in line with terminology in literature. The planning process consists of three phases:

**Phase 1** The nurse planners have discussions with the heads of the nurse department to discuss the required skills and duties that have to be performed each day according to the expected demand per department. From these required skills and duties over the days, the shifts are created for each department for a three-month time horizon. Then the nurse planners investigate when a person is not allowed to work either because he does not have the required skillset to perform a certain duty or needs to be absent. For example, this person has to attend work training. Lastly, the personalised shifts that the persons can work on based on the expected demand for the duties and the nurses' availability are sent to nurses. The nurses can then individually fill in the shifts they prefer to work for the coming three months. The nurses can also use three jokers to set the days they absolutely want to work or, more commonly, on days they do not want to work.

**Phase 2** The second phase is the negotiation phase. Every shift with the required capacity of nurses is locked so that the shifts are fixed with the correct number of nurses. The nurses can exchange their shifts in this phase. The nurses are awarded points if they accept a request to interchange shifts with another nurse. The nurse planners have indicated that the success of the self-rostering method depends strongly on the willingness to cooperate between the department's nurses.

**Phase 3** The last phase focuses on making the rosters complete by moving persons from shifts with too much capacity to shifts that still need more persons working. This is a complex process in which the planners have to consider the hard and soft constraints

but also the fairness of the schedule. Most nurses favour working a day shift compared to a night shift as mentioned during the interview with the nurse planners of hospital X. Fairness is considered by the number of points the nurses received in the second round and how many unfavourable shifts this person worked in the past.

Before the team rostering implementation at the hospital, the planning was solely determined by the planners based on the required shifts per day and the employees' wishes. However, as noted by a hospital planner, some of the nurses thought that the work of the planners could be improved and that team rostering would improve the quality of the schedule. However, for most of the departments, the consensus of the nurses and the planners is that the planning quality has deteriorated with the introduction of team rostering. This mainly concerns the fact that there is barely cooperation in exchanging shifts in the second phase of the planning process. Still, after they experienced the complexity of the rostering process, they readjusted their regards for the planner since they now work more unfavourable shifts than before. Healthcare professionals often do not have training in operations research and lack the ability to manage effective care for patients, which may hinder the creation of an optimal schedule (Hall, 2012). The consensus for most departments at hospital X is that the team rostering, therefore, has resulted in worse schedules.

### 1.3.4 AI-assisted workforce scheduler tool

Hospitals in the Netherlands widely use the OWS software. However, the optimiser is not or barely used by the software owners in hospitals because of insufficient performance of the optimiser. ORTEC uses a heuristic approach to find the optimal schedule given a set of hard and soft constraints that can be quantified mathematically. The solution quality is based on this given set of constraints. Human planners also try to minimise the violations of the constraints for all the employees as well as possible but also account for more aspects of planning that cannot be modelled efficiently in a mathematical model. In the classical sense, the optimisation problem for the NSP in literature mainly focuses on cost minimisation for employee deployment or constraint violation minimisation, but given the complexity of the problem, this provides insufficient results (Ernst et al., 2004).

ORTEC has started an initiative to explore new ways to improve the use of the OWS optimiser. The idea is to incorporate Artificial Intelligence (AI) techniques into a decision support system that aids the nurse planners with making the planning for their personnel. Artificial Intelligence may prove to be useful, especially when the problem at hand involves many human factors that cannot easily be modelled in an optimisation model, which is the case for the hospitals that make the planning for their personnel. Decision support may also facilitate the first step to adopting a complete automatic scheduler system because it helps to increase the support for a black-box system if the decision support proves successful (Ernst et al., 2004).

Figure 1.2 shows the plan for this AI-assisted workforce scheduling tool. Currently, five research projects are planned for the research and development of the tool, which can be grouped into two branches. All research projects are conducted at ORTEC B.V. and are in collaboration with two Dutch collaborating hospitals. This research mainly collaborated with one of those two hospitals. The five research projects can be subdivided into two branches: 'Learning roster' and 'Changing roster'. The Learning roster branch focuses on how ML can support scheduling decision-making. The second branch, 'Changing rosters', focus on what aspects of the roster characteristics are important for the stakeholders involved in the planning process.

Euser (2022) did interviews with the planners in the hospitals to investigate what properties and considerations in a roster are important for the quality of the roster as regarded by the human planners. Based on these found properties, Euser (2022) did an extensive SHAP analysis that showed that some hard constraint violations, such as violation of forward rotation (the difference between the start times of two consecutive shifts for an employee is less than 24 hours) and the number of weekends a nurse has to work, impact the quality of a schedule, but this also differs per department. Rooijen (2023) recently started her MSc assignment to find the characteristics of the schedule that impact the perceived roster quality from the nurse's perspective. Combined, these two researches identify the aspects of a roster that are perceived as most important from both the planner's and the nurse's perspective. Understanding these qualities may help to better differentiate between a good and bad roster and can help create an AI-assisted workforce scheduling tool. The branch 'Learning rosters' focuses on the research and development of ML techniques to make rosters based on the realisation of rosters in the past made by human planners. Cissen (2022) performed a proof-of-concept study for basic shift prediction without utilizing any features. Unfortunately, the problem became a lot more difficult when additional features and class imbalances were considered, therefore we could not directly use her results. This research focuses on the assignment of nurses to the days in the planning horizon by including relevant planning features and making the predicted schedule feasible. The deliverable of this research provides a schedule of when the nurses worked during the planning horizon based on actual realised planning in the past.

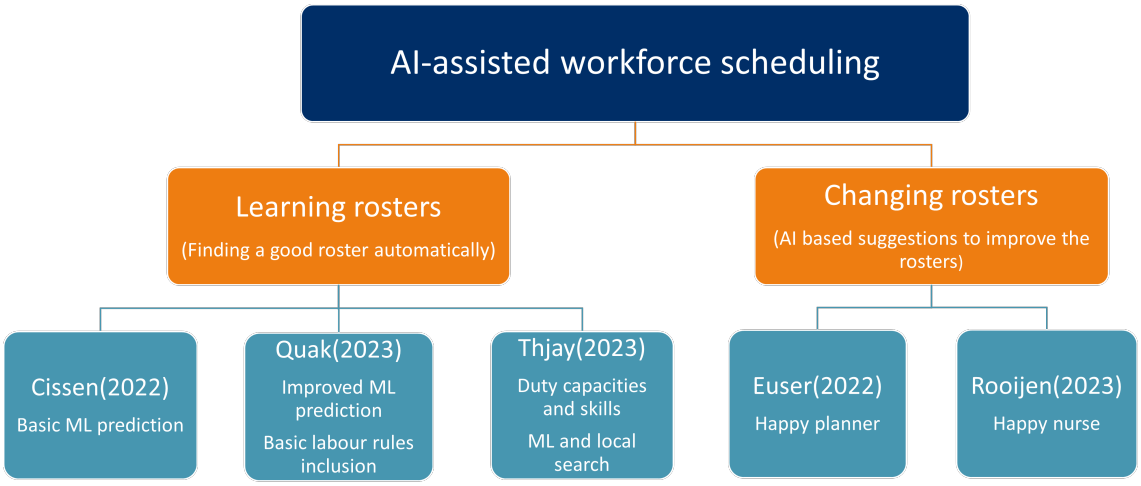



FIGURE 1.2: AI-assisted workforce schedule

## 1.4 Identification of the research problem

Human planners in the hospital make the planning themselves by trying to adhere as much as possible to the hard and soft constraints in their limited planning time. This may lead to sub-optimal schedules, which may overstress the need for nurses more than needed. ORTEC wants to improve the quality of the optimiser in the OWS software for the hospital customers that have bought the OWS license. The Genetic Algorithm (GA)

currently used by the optimiser still has room for improvement and does not perform well on the most used Nurse Scheduling Problem (NSP) benchmark instances (Hassan, 2022). These problem instances vary in problem size, time, number of employees, shift types and constraints.

ORTEC has tried numerous times to find improvements for the optimiser for hospitals within several projects, but without much success. The literature on the NSP shows that the most successful real-life implementations consist of a combination of AI and Operations Research (OR) techniques, including problem-specific information about where the solution is implemented (Burke et al., 2004).

Given the problem of ORTEC, this research focuses on developing a Machine Learning tool that helps nurse planners by prescribing when a nurse is going to work based on the learned planning rosters in the past. The prescriptions are based on the ML prediction of when a nurse will work in the future. Since ORTEC is not allowed to publish any schedules that violate labour rules or hard constraints, this research also investigates how the created schedule can be made feasible through metaheuristic techniques.

The main research question is defined as follows:

> *'How can a combination of ML and improvement heuristics create a feasible schedule that adheres to nurse preferences for hospital X?'*

We designed five research questions to answer the main research questions:

1. What approaches for the NSP has been applied in literature?
   (a) What approaches have been tried for the NSP and scheduling problems in general?
   (b) What factors can make the solution approach for the NSP successful?
   (c) What approach is most appropriate for the problem context of this research?

   First, a literature study on the applications for scheduling problems, and more specifically, the NSP is investigated in the first research question. Although the body of literature on scheduling problems is expansive, the applications of the developed solution methodologies for nurse scheduling are not used in real-life (Ernst et al., 2004). Since the goal of the research is to make a tool that is applied successfully in real-life. The second sub-research question investigates the factors that make the solution approach successful for real-life applications. Identifying the barriers to adoption may help to improve the applicability of the designed solution methodology. The last sub-research question answers what approach seems most suitable for making the prediction of nurse rosters in hospital X.

2. How is the planning of the nurses organised in hospital X?
   (a) What structure is used by the planners to make the planning for the nurses?
   (b) What are essential characteristics of a roster that impact its quality?
   (c) What are important aspects that can make a schedule infeasible?
   (d) What data can be used for this research?

   Now that the general approach is clear, we focus on the context of the problem. The nurse planners have designed a specific structure to plan the nurses by considering the wishes of the nurses and the requirements of the departments as well as possible.

We visited the hospital X once during the thesis project and talked with some nurse planners to understand the planning process better. As explained in Section 1.3.4, a branch within the project of the AI-assisted planning tool is to understand the features in the rosters that impact the quality of the roster made. The available results of the research of (Euser, 2022) from the branch 'Changing rosters' are used to create useful features for the ML model. An essential aspect of the solution is that the solution should not contain any infeasibilities. There are more than a hundred hard constraints used in practice for hospital X. The most important constraints are selected in the third sub-research question. The last sub-question performs the Explorative Data Analysis and checks what data can be used for the prediction model, given the answers to the previous sub-research questions.

3. How can the design of ML models be optimised for schedule prediction?

   (a) What features are most relevant for the schedule predictions from the available data?

   (b) Which ML models are most suitable for the problem context?

   (c) Which hyperparameters result in the best performance?

   The third research question focuses on the design of the ML model. The selected format of data and the features are selected for the ML model in this research question. Also, the feature engineering steps to create more powerful features are described in this first sub-question. The next sub-research question focuses on what ML models best suit the data and the problem context. The third sub-research question discusses the hyperparameter tuning of the ML models for time series data.

4. How can metaheuristic methods make the solution feasible?

   (a) What metaheuristic technique is appropriate for improving the solution quality in this thesis?

   (b) What should the design of the metaheuristic look like?

   (c) What is the objective function of the metaheuristic procedure?

   (d) How should the neighbourhood of the solution be defined?

   Now that the ML model design is ready, we need to ensure that the solution does not violate any hard constraints while the solution value stays as close as possible to the predicted schedule. This research implements a search strategy that efficiently searches the neighbourhood solutions for a solution as similar as possible to the predicted schedule but meets the hard constraint rules. Research question 4a deals with the metaheuristic approach to finding a feasible solution. The framework design of the metaheuristic is investigated in research question 4b. Research question 4c focuses on the objective function formulation for the metaheuristic. Lastly, the operators for the metaheuristic are investigated.

5. What is the performance of the selected solution methodology?

   (a) How can the performance of the ML model be evaluated?

   (b) What is a suitable method to benchmark the results?

   (c) What is the performance of the ML models?

   (d) What are the results of the metaheuristic?

(e) How can the proposed application methodology aid in hospital planners' schedule-making?

Now that the design of the solution methodology for the NSP is defined, the results need to be evaluated. The first research question answers how the performance of the ML prediction models can be evaluated robustly. Sub-research question 5b investigates the design of a benchmark tool that evaluates the performance of the ML techniques against a naive method. The performance of the ML models based on the selected evaluation methodology is described in the third sub-research question. Based on the metaheuristic method described in the third research question, the metaheuristic results are shown. The last research question investigates how the results can be integrated into the nurse planners' planning process.

Figure 1.3 shows the research design of the study.

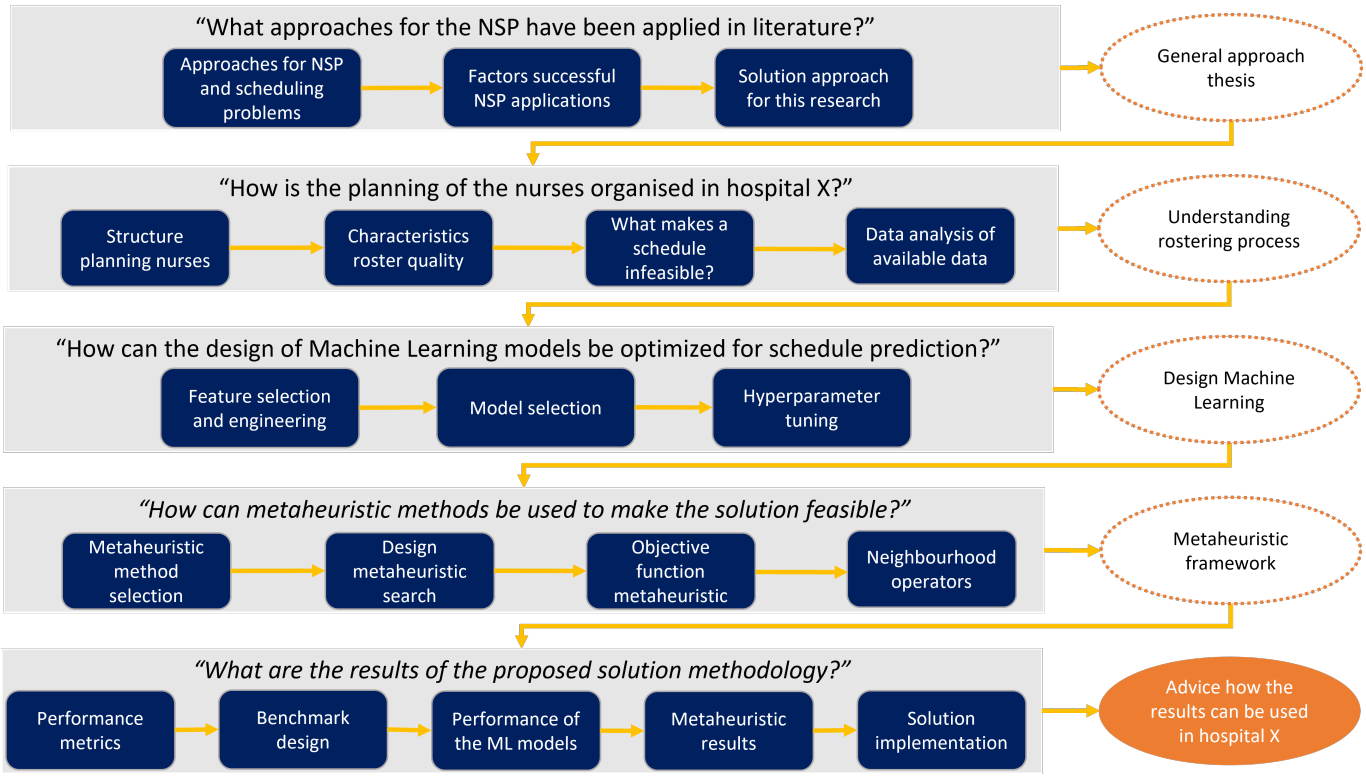



FIGURE 1.3: Research question design of this project

# Chapter 2

# Literature search

This chapter addresses a review of the relevant literature for this research. First, Section 2.1 describes the hierarchical levels of the management of the nurse scheduling process to give an overview of how the nurse scheduling is organised. Section 2.2 then gives an introduction to the situation of nurse scheduling in general and gives a definition of the NSP. Section 2.3 focuses on the solution approaches used in literature for the NSP, and lastly, Section 2.4 investigates the research gap for the NSP that this research aims to fill.

## 2.1 Hierarchical nurse scheduling management

In this section, the steps of nurse scheduling are described on different levels of management. This description is inspired by the taxonomic classification of planning decisions by Hulshof et al. (2012). This research gives an overview of resource capacity planning and control in healthcare based on relevant articles. Hulshof et al. (2012) distinguish between four hierarchical levels of planning: strategic, tactical, offline operational and online operational. In a healthcare facility, many different types of planning decisions need to be made. For this research, we will only be focusing on the aspects directly relevant to the management of planning of the nurse workforce. However, it's important to note that this is just one aspect of the broader planning decisions that are made in a healthcare facility. This discussion on the management structure covers various levels of hierarchy, starting from the highest to the lowest level. These levels include the strategic level, the tactical level, and the operational level.

### 2.1.1 Strategic level

The strategic level of nurse planning aligns the long-term plans for the nurse planning with the objectives and direction of the hospital for health care delivery. The strategic level for nurse planning mainly deals with the number of nurses expressed in full-time equivalents with the necessary skills needed to meet the management's goals for healthcare delivery (Rönnberg et al., 2013; Wright & Mahar, 2013). The strategic decision maker should determine the required number of personnel per department while accounting for non-working activities such as personnel training, sick leave and holidays. A flexible work pool can help to deal with seasonal demand fluctuations by lending a nurse from a department for which the work pressure is high to a department that needs extra staffing. Nurses in the flexible work pool need the training to work in more than one department or learn new specialties, which takes time and training. Still, the flexible work pool improves the departments' resilience to deal with demand fluctuations (Burke et al., 2004).

### 2.1.2 Tactical level

On a tactical level, the nurse scheduling process deals with shift scheduling. The tactical decision maker calculates the required shifts for the departments and the flexible work pool with their respective number of workers needed from each speciality and sets the shifts' beginning and end times (Hulshof et al., 2012). In the case of hospital X, this process is done every three months.

The first step in shift scheduling is modelling the patients' demand (Ernst et al., 2004). The way that the demand is modelled differs strongly in research papers. For some business areas, the incident rate for personnel is already known (partly) in advance, such as airline maintenance and can therefore be used directly as demand in the model. In hospitals, the number of nurses is calculated by a set of regulations that indicate the number of nurses needed per speciality per patient (Griffiths et al., 2005). Unfortunately, most hospitals do not know the best combination of nurse skills to schedule for the patients' demand (Aiken et al., 2002). Often a relatively simple technique is used for demand forecastings, such as regression, exponential smoothing, or seasonal moving averages. Research in the rostering literature often assumes that demand is already known beforehand or simple methods can be used to model the demand. The demand, however, determines the required shifts and tasks that need to be performed; therefore, a reasonable demand prediction model is essential for a good predictor model. This is especially the case for some of the departments in The Netherlands for which the demand of the workload can fluctuate strongly, such as the IC department, because of the unpredictable timing of critical illnesses (Jenkins et al., 2006). When the demand is known, the bed occupancy rate of the departments and the medical needs are calculated. From there, the planner should translate this information to shifts and the number of nurses needed per shift (Hulshof et al., 2012).

### 2.1.3 Operational level

On the operational level, short-term decision-making for healthcare delivery is performed. The elective demand is known entirely now, and only the emergency demand is not known yet since this emerges spontaneously. We distinguish between the offline operational level and the online operational level. Offline operational deals with the detailed coordination of the planning activities based on the elective demand. At the offline operational level, the employees are assigned to shifts based on the shift requirements, which were determined at the tactical level. The scheduling of the nurses may be organised in different ways. Section 2.2.3 describes the main scheduling forms used in hospitals.

The lowest organisational level is the online operational level which deals with the monitoring and reaction to unexpected events, such as an employee calling in sick or an emergency patient that needs to be treated, which was not foreseen. Depending on the situation, the staff may be rescheduled before the start of their shifts. A summary of the organisation of nurse scheduling is depicted in Figure 2.1. As the level of management involved in the planning process increases, there is greater flexibility, but also longer time horizons and increased uncertainty. The main activities per management level are shown in the figure along the planning horizon in hospitals in practice. This research focuses on the shift assignment of the nurses in the hospital. While the shift assignment is normally done one week in advance for hospitals, this is done three months in advance for hospital X and starts directly after the number of shifts has been determined at the tactical level. The planning level that is the focus of this research is highlighted green in Figure 2.1

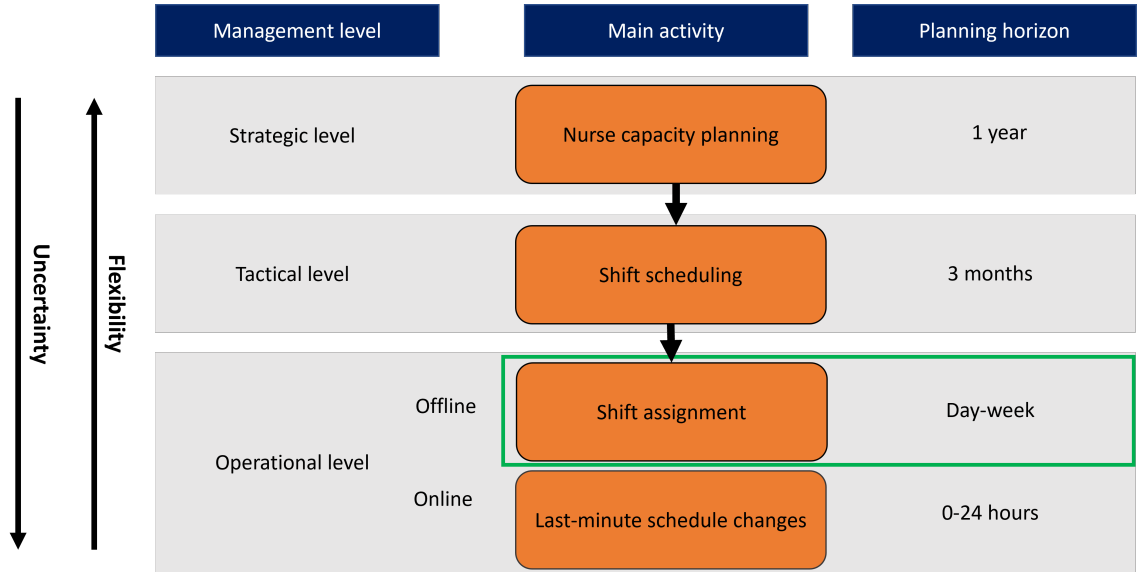



FIGURE 2.1: Management structure of nurse scheduling

## 2.2 Introduction NSP

Nurse scheduling encompasses the management of shifts, vacations and all the aspects involved in the planning of the nurses in the daily operation of the hospitals (Tsai & Li, 2009). The NSP is an Operations Research (OR) problem aiming to assign nurses to the available shifts while complying with the hard and soft constraints as much as possible (Solos et al., 2013). The NSP is applied to constraint planning problems in other business areas, but we focus primarily on the application of the NSP in hospital settings because this fits the focus of the research the best. This section gives a general introduction to the NSP by explaining the relevant constraints in Section 2.2.1, describing the schedule objective in Section 2.2.2 and explaining how team-rostering is organised in literature in Section 2.2.3, which is the recently adopted approach in hospital X for making the nurse schedules.

### 2.2.1 Constrained solution space

The NSP often consists of multiple and conflicting objectives such as cost minimisation and maximising the workforce's satisfaction (Legrain et al., 2014). The satisfaction of nurses is related to the workload, working conditions and personal preferences (Wright & Mahar, 2013). It is often challenging, if not impossible, to comply with all the labour rules because of the demanding healthcare situation. Optimisation of the workforce allocation is especially challenging and essential for the operating department. The revenue for this department is increasing steeply with the ageing population, while the work environment demands more restricted schedule-making. Still, adhering to the preferences of the nurses improves the work satisfaction of the nurses. Increased job satisfaction leads to reduced nurse turnover and improved quality of care (Bester et al., 2007; Blythe et al., 2005; Hall, 2012). This can alleviate some of the pressure on the hospital system and reduce the shortage of nurses. Therefore, it is crucial that hospital administrators optimise the scheduling of nurses while complying with the labour rules and the nurses' preferences.

The NSP is often modelled as an optimisation problem. In an optimisation problem, rules that should be complied with at all times are called hard constraints. The rules that

may be violated to some extent but lead to penalty costs to discourage the decision-maker from doing so are called soft constraints. In the case of the NSP, the labour rules are often modelled as hard constraints, while the preferences can be seen as soft constraints.

### 2.2.2 Schedule objectives

As noted in the research by Euser (2022), the schedule quality from the planner's perspective is a complex composition of the score on multiple schedule qualities. These qualities can be grouped into health score, roster policy and preference adherence. The health score relates to the employee's working hours, while the roster policy dictates the fairness of the workload distribution. Lastly, the compliance of the nurse preferences is ranked by preference adherence. These qualities are based on a hospital X that provided the data for the research. The three important individual characteristics for the quality of the planning, according to the planners, were shown to be a violation of forward rotation (less than 24 hours between start times of consecutive shifts), the ratio between shifts worked at the weekend compared to the total weekend work days, and the ratio between the violation of preferences and the total number of preferences. Euser (2022) noted from an interview that the qualities might differ by the planning strategy per hospital and thus that an optimal schedule may be defined differently per hospital.

Uhde et al. (2020) explain the concept of fairness. Fairness is a broad term and consists of three components: equality, equity and need. Equality means the fair division of the shift among the employees. Equity is the equal distribution of less-attractive and attractive shifts among employees. Nurses may be reluctant to work a New Year's eve shift and expect to be compensated with a preferable shift in the future. Lastly, need refers to the expected required flexibility based on personal circumstances. A single parent, for example, may require more flexibility in scheduling than a nurse without children living close to the work facility.

### 2.2.3 Team rostering

Most hospitals worldwide have moved from a traditional centralised scheduling approach to a more decentralised approach in which nurses are given more autonomy in how the schedule looks like. The rationale for this change is that the resulting schedules better fit the needs of the nurses and increase the feelings of autonomy, increases satisfaction among the nurses and increase the overall productivity (Bailyn et al., 2007; Silvestro & Silvestro, 2000). Also, hospital X has recently changed from a centralised planning approach to a less decentralised rostering process called team rostering. Team rostering is a planning approach that provides more autonomy to the nurses than departmental rostering but does not give the full individual schedule flexibility that self-rostering provides. Figure 2.2 shows the placement of team rostering in between self-rostering and departmental rostering. A description of the three scheduling approaches and the considerations for the implementation are given below:

- The first trials for self-rostering in hospitals started in the 60s, but it is until recently that a considerable number of hospitals have implemented some sort of self-rostering in their hospital departments (Hung, 2002; Silvestro & Silvestro, 2000). The nurses have almost complete autonomy in a self-rostering approach; a senior manager often only finalises and approves the final schedule. Self-rostering has shown to be beneficial by decreasing sickness among nurses and increasing satisfaction and perceived ability of roster control (Garde et al., 2012; Silvestro & Silvestro, 2000). However, the success

of self-rostering depends on the investment in training every nurse to understand the rostering problem and the implication of individual scheduling decisions on the department roster (Silvestro & Silvestro, 2000). Case studies have shown that clear benefits can be attained on the condition that the nurses adhere to the predefined scheduling rules and do not see self-rostering as personal entitlement (Bailyn et al., 2007). The possibility of a successful implementation can be increased when the nurses are included in all stages of program development (Russell et al., 2012).

- On the other hand of the scheduling spectrum is departmental rostering. In departmental rostering, a single senior nurse is responsible for the planning while trying to adhere to the requirements of the scheduling unit and the involved nurses as well as possible. Departmental rostering has the advantage that greater control can be exercised over the roster when only one person is responsible for the roster making. Also, it is easier to deal with unforeseen problems, and the time to make the rosters is significantly less than for self-rostering or team-rostering (Silvestro & Silvestro, 2000). This is also noticed by the planners from hospital X. While centralised control of the planning system can have tangible benefits, success relies on the interpersonal skills of the planner responsible for the planning.

- Hospital X uses a form of scheduling which can be placed in between departmental control and self-scheduling called team rostering. The precise team rostering process at hospital X was explained in Section 1.3.3. Nurses can indicate the shifts that they want to work based on the set of shifts that they are allowed to work on and need to collaborate with each other to construct a final schedule. One nurse planner oversees the team rostering process and ensures that all shifts are filled. Continuous support of the nurse planner and clear guidelines for the nurses are required to enable team rostering to work optimally (Wynendaele et al., 2021). Barriers to the adoption of team rostering include competition between the participants and favouritism by the head planners.

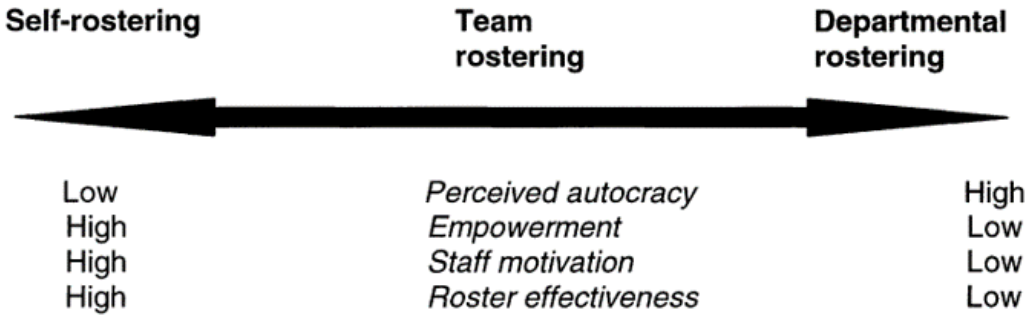



FIGURE 2.2: Placement of team rostering from Silvestro and Silvestro (2000)

## 2.3 NSP solution methods

The NSP has been studied extensively in the literature, with various solving techniques used to optimise the planning. Because of the combinatorial explosion of possibilities with an increased shift length, it is challenging to find good solutions while complying with the highly constrained environment and even more so to find the optimal solution

(Morgado & Martins, 1993; Petrovic & Vanden Berghe, 2012). The NSP is known to be Non-deterministic Polynomial time (NP)-hard complex (Solos et al., 2013).

Nowadays, planners often require decision support to help make scheduling decisions in which the right workers are assigned to the right shift. A decision support system often consists of spreadsheets, database tools and roster optimisers based on mathematical algorithms (Ernst et al., 2004). Still, hospitals often do not use the automatic scheduling systems they possess and mostly use computer software to edit the schedule instead of schedule generation (Burke et al., 2004). Therefore, the scheduling is often carried out by human expert planners who have acquired the knowledge to schedule through experience (Aickelin & Li, 2007). However, some tangible benefits can be obtained by adopting automatic scheduling: increased schedule generation speed, improved schedule fairness for the employees and better and faster adaptations to new situations (Morgado & Martins, 1993).

Although the body of literature on NSP is extensive, the implementation of the methods still falls short (Legrain et al., 2014; Van den Bergh et al., 2013). There are multiple reasons that hospital administrators rarely use decision support software to the full extent. Reasons such as a lack of trust in 'black-box' systems, the narrow focus of scheduling problems in academic articles, the focus on new techniques to stylised unrealistic NSP problems rather than applicability of the solution method, lack of customer support for the developed software and the current organisation of self-scheduling are named as a hindrance to the acceptance of automatic scheduling procedures (Kellogg & Walczak, 2007; Petrovic & Vanden Berghe, 2012).

The remainder of this section is structured as follows: first, the modelling steps for the NSP are discussed, providing a high-level overview of the three steps in creating the mathematical model for the NSP. Section 2.3.2 then describes the solution methods used for the NSP for the second mathematical model step. Also, the barriers to the adoption of the solution methodologies are explained. Since a combination of ML and metaheuristic techniques are applied in this research, Section 2.3.3 and 2.3.4 delve deeper into the literature on this subject.

### 2.3.1 Mathematical model steps

This section gives a literature review of the solution methodologies applied to the NSP. The mathematical model should include three steps in general (Ernst et al., 2004):

1. Model of demand that can forecast based on historical demand data. This demand should then be translated to the required staffing levels to meet the demand adequately.

2. Selection of a solution method that optimises the scheduling for the problem in the constrained solution space.

3. A reporting tool that presents the solutions to the planners and possibly employees.

The first step in solving the NSP is often the modelling of demand. This step determines the hours and tasks required from the personnel and is, therefore, an important step in the NSP. The way that the demand is modelled differs strongly in research papers. This first step is not done by the optimiser and is done by the hospitals themselves instead. The planners make use of forecasting models for the expected demand for healthcare and discuss with the team lead of the nurse groups the exact nurse requirements and the skills needed. Hospitals that use the optimisers give the demand and shifts as input to the roster.

The solution technique should satisfy the hard problem constraints, such as the labour regulations and meet the planning objectives as well as possible. Ideally, the solution method should be able to handle multiple conflicting objectives, such as cost minimisation and maximisation of employee satisfaction. There exists a large number of planning support software packages. Software targeted to optimise a specific application area is challenging to transfer to other scheduling fields. In contrast, less problem-specific planning software that can be used more broadly often focuses more on editing functions and elaborate reporting tools but has limited capacities to generate adequate schedules automatically (Ernst et al., 2004).

### 2.3.2 Solution methodologies in literature

Since the first step in the mathematical model to forecast the demand and find the appropriate expected need for resources and employees is given as input in the optimiser and done by the hospital planners themselves, we now focus on the solution methodologies described in the literature. The literature on solution methods for personnel scheduling is expansive, with a wide range of solution methods, from integer programming to simulation (Rais & Viana, 2011; Van den Bergh et al., 2013). The solving techniques are mostly skewed to mathematical programming and metaheuristics for workforce scheduling and less towards methods such as Constraint Programming and ML for staff scheduling. While AI can provide a richer and more flexible representation than OR methods, the OR method's strength lies in finding the (local)-optimum of well-defined problem spaces (Gomes, 2000). Mathematical programming models the NSP as either a linear, integer or mixed integer program. A metaheuristic is a mathematical procedure that can find an approximate solution to an optimisation problem by making an intelligent trade-off between metaheuristics and local search exploration (Bianchi et al., 2009; Gandomi et al., 2013).

The advantage of metaheuristics over linear programming and other exact approaches is that a good feasible solution can be found in a shorter computing time while sacrificing the quality of the solution value (Van den Bergh et al., 2013). Metaheuristics are useful when solving the problem with exact methods is too challenging or time-consuming, and the problem owner is searching for satisfactory solutions instead of the optimal one. OWS also uses a combination of metaheuristics techniques to find the optimal solution. An important method within metaheuristics is Genetic Algorithms, which has been studied intensely in the literature for the NSP and is also the first step in the optimiser of OWS (Aickelin & Li, 2007; Laurens et al., 2006). Although Mathematical programming techniques are most often applied, these methods struggle with applying domain-specific constraints into the problem formulation (Abdennadher & Schlenker, 1999).

Most researchers still use the set covering formulation first introduced by Dantzig (1954). This method makes it possible for the problem definers to formulate the constraints for the scheduling problem as required. Because of the large number of constraints that the problem may have, researchers often use heuristics or decomposition techniques (Cheang et al., 2003). Decomposition techniques break up the problem into easy constraints and challenging constraints. The cut generation scheme leads to a problem with only the easy constraints that functions as the heuristic solution, while the hard constraints are included in the other cut, which solves the problem exactly (Detienne et al., 2009).

Van den Bergh et al. (2013) did an extensive study on the methods applied to the NSP in literature. They note that the NSP has been studied in depth in the literature but often focuses only on scheduling personnel with fixed inputs. The solution methodologies are often tested on a standardised set of twenty-four test instances for which the best solution values are known ("Nurse Rostering Benchmark Instances", 2023). Scheduling of nurses

often includes more decisions such as demand forecasting, hiring and firing. Because many elements of Nurse Scheduling are often neglected, the solution methodologies for the NSP from the literature are not often put into practice. The more real-life planning elements are integrated into the NSP approach, such as flexible contracts, skillsets required for the jobs etc., the better the solution approach can be applied in real-life settings. In addition, Kumar et al. (2019) note that as the number of constraints and workforce increase, manual scheduling becomes impractical. An OR model may be used to find an appropriate solution for large problem instances, but modelling the constraints is also a very challenging task because of the number of constraints involved and the subtle nuances between a good and bad schedule (Euser, 2022; Kumar et al., 2019). A domain and modelling expert may help to implement the constraints, but this can be laboursome and costly (Burke et al., 2004).

Van den Bergh et al. (2013) noticed that the approaches for NSP allow for more flexibility recently than before but that the models still cannot take the complex preference of the nurses into account for doing the planning. Also, including uncertain parameters, such as the demand, could help adopt the solution methodologies in real life. Most studies only consider static real-life data to make and test the model. Still, the situation of hospitals is dynamic, and therefore a good planning solution should take new inputs into account for generating the schedule with the solution methodology.

Papers have been published on computer-assisted healthcare planning for over sixty years. Still, only a handful of papers describe the application to real-life instances with traditional (linear)-models for hospital planning (Legrain et al., 2014). The NSP is regarded as even more difficult to solve to optimality than the Travelling Salesman Problem (TSP), for which the best-known quantum exact algorithm is $O(1.728^n)$ (Tien & Kamiyama, 1982). Since the larger departments of hospitals can have more than 100 nurses at a time, solving the NSP to optimality requires an exceedingly infeasible amount of computing power.

Since the NSP has been seen as a pure mathematical program in the academic world, the focus for solution methods is based on OR to solve the NSP. Since real-life instances of the NSP are infeasible to solve, researchers have made simplifying assumptions to reduce the complexity of the problem instances, such as considering the number of resources and shift to be smaller and not considering all the constraints (Wiers, 1997). Therefore, the applicability of the solution methods to real-life problems is significantly reduced.

Due to the limited practicality of the traditional OR solutions and partly because of the emergence of AI techniques, more NSP research has been focused on applying AI models. The success of AI models is attributed to the ability to deal with complex problems and its ability to deal with intangible decision-making considerations. The integration of ML in a metaheuristic approach leads to an efficient, effective and robust search and can lead to superior performance, convergence speed and robustness of the solution methodology (Karimi-Mamaghan et al., 2022).

The following subsection focuses more on the AI approaches available in the literature for the NSP.

### 2.3.3 AI approaches for NSP in literature

ML techniques can be used for a range of workforce scheduling operations. Serengil and Ozpinar (2017) successfully implements a neural network first to predict the workload size and then optimise the workforce planning by clustering the employees based on their performance and their skills with unsupervised k-means. This hybrid multi-level scheduling proved to predict workload better than conventional time-series forecasting techniques such as exponential smoothing methods.

An increasing number of researchers are developing hybrid methods for the NSP by

combining exact OR approaches with the learning capabilities of AI approaches. Exact methods may be able to find optimal solutions but are limited by the size of the instance and are infeasible to solve for large real-life instances. Heuristic approaches can find feasible high-quality solutions but do not attain the accuracy of exact methods. Therefore, studies are started to use learning mechanisms for discrete optimisation problems. Most studies applied ML techniques in the field of the Travelling Salesman Problem and Vehicle Routing Problem for discrete optimisation problems, but Chen et al. (2022) suggest that these approaches may also be useful for the NSP.

Several authors have designed hybrid methods that use ML techniques that optimise the performance of their exact methods (Chen et al., 2020, 2022; Khalil et al., 2016). For example, Chen et al. (2020) uses a Deep Neural Network-assisted tree search to find the distance between the solution and the best possible solution and uses this information to steer the search direction of their Mixed Integer Problem.

Li and Aickelin (2003) proposed a Bayesian optimisation algorithm that mimics the way schedulers make scheduling decisions by using a set of suitable rules for each of the nurses. The Bayesian optimisation algorithm is able to learn good partial solutions and finalise the schedule by constructing a Bayesian network of the joint distribution of solutions.

### 2.3.4 Metaheuristic approaches in NSP literature

Metaheuristics are a commonly used method to obtain near-optimal solutions in a reasonable time when exact methods are not able to do so (Voß, 2001). The efficiency of metaheuristics for the problem context is essential in the case that the problem owner not only wants to find a feasible solution but also the quality of the feasible solution given an objective function is essential.

As noted in Section 2.3.2, metaheuristics form an important group in solving the NSP in literature since these can be designed to tackle complex optimisation problems where other solution methods failed to obtain a good feasible solution efficiently or even a solution at all (Van den Bergh et al., 2013). The practical benefit of using the metaheuristic approach is the effectiveness in searching the solution space and the generality of use for other application areas. Many forms of metaheuristics exist, and they have been applied in the literature. Examples include Simulated Annealing (SA) (Turhan & Bilgen, 2020), Genetic Algorithms (Aickelin & Dowsland, 2004) and Particle Swarm optimisation (Günther & Nissen, 2010), Variable Neighbourhood Search (Stølevik et al., 2011) and tailor-made heuristics (Lü & Hao, 2012). Especially Simulated Annealing, Genetic Algorithms and Tabu Search are often applied in literature for scheduling problems.

A recently upcoming promising area of research is the use of hybrid methods to solve the NSP. Metaheuristic approaches can be combined with an exact solving method in a hybrid method, such as Integer Programming (IP) techniques, to obtain a more robust method. An example is the paper of Turhan and Bilgen (2020) that uses a combination of a technique called Fixed and Optimise, and Simulated annealing. First, the MIP of the NSP is solved as a relaxed MIP problem, and a good initial solution is found. They then use a Simulated Annealing approach to search the solution space for better solutions. After a constant number of iterations, the Simulated Annealing is stopped, and the Fixed and Optimise method is started. This method decomposes the schedule into weeks and fixes the best-performing weeks for the objective value. The rest of the weeks are individually solved by the MIP, and the week schedules are then combined to get a schedule for the whole planning horizon again. In this way, the Simulated Annealing can better perform diversification of the search space.

## 2.4 Gap in literature

In this research, we propose a hybrid method to aid in the nurse planning for hospital X. ML techniques are used to make an initial schedule for the nurses based on past realisations of schedules. These ML models mimic the past schedule-making process of human planners given the relevant prediction features. Since the schedule that is published by ORTEC can not contain any violations of hard constraint and we require a minimum number of persons to be working every day based on the duties that need to be performed every day, we use a metaheuristic approach to search the solution space for solutions that comply with these hard rules, but still, stay as close as possible to the predicted schedule. In this way, the final created schedule resembles the human schedulers' scheduling-making process and still leads to feasible schedules.

To the best of the author's knowledge, no hybrid models have been implemented that predict a roster based on past realisations and use metaheuristics to improve on this ML solutions. Many of the proposed solutions are not used in real-life, mostly because of the flexibility that is not present in the solution approach (Ernst et al., 2004). Although the NSP is a long-known problem and has been researched extensively, the research mainly focuses on a selective set of NSP examples that do not include nurse preferences. This research aims to partly fill the gap by approaching the problem in a different light and changing the perspective of the NSP from a pure optimisation problem to a prediction problem that combines elements of optimisation problems to ensure a feasible schedule is created. With this new approach, the nurses' preferences are better cared for, and more flexibility can be provided. These are identified as the most critical hindrance to adoption.

# Chapter 3

# Data preparation

This chapter lists the data preparation steps for this research based on the provided database. The first section describes the data provided to us. Next, data analysis is performed, which serves as the starting point for the feature engineering section. Section 3.3 describes the train-test split used to tune the parameter for the ML models. Lastly, the data transformation steps are discussed. Figure 3.1 summarises the data approach steps.

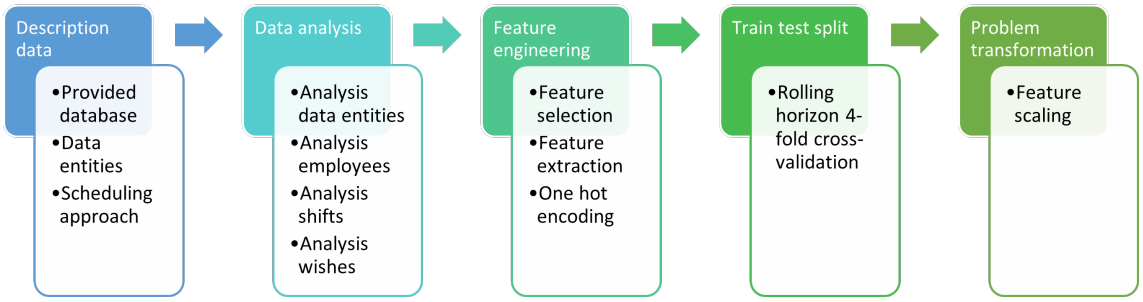



FIGURE 3.1: Visualisation of the data approach steps

## 3.1 Data analysis

An ORTEC Workforce Scheduling database for a large hospital in The Netherlands was used for this research. The data contained information about 600 departments from hospital X spanning from the 1st of April 2016 to the 1st of September 2022. The databases were anonymised according to the GDPR to ensure the privacy of the hospital employees. Some of the data that would be useful for this research was not yet available in the database, such as the seniority of the employees and the skills they can perform, which describe the activities that the nurses can perform and help to better assess the feasibility of the schedule. Although employees of ORTEC are working on extracting this useful information, it cannot yet be used for this research.

This section describes the data contents of the provided database and sets the direction for the selected features in Section 3.2. The information was retrieved from the database by using SQL Server Management Studio. This section is structured in the following way: Section 3.1.1 describes the main data entities of interest for this research. Then, Section 3.1.2 summarises the data entities for all the departments, Section 3.1.3 provides information about the nurses in a department, Section 3.1.4 describes the number of shifts over time, and lastly, Section 3.1.5 analyses the type and number of wishes in the database.

### 3.1.1 Data entities

This section describes the data entities selected from the database for this research. These data entities are derived from the planning concepts utilised in OWS planning software. This section clarifies the different concepts used by the software for the planning elements. These are department, duty, shift and resource and wishes.

**Department** The managerial division of the care delivery for the patients. An example of a department is the radiology department.

**Duty** A set of activities performed by the employees, such as performing blood tests without specifying the time and date that this has to be performed.

**Shift** Shifts are determined in the tactical management phase (see Section 1.3.3) based on the expected number of duties that have to be performed for the delivery of care. Skills are the realisation of duties in the planning horizon with an assigned time window and assigned resources.

**Resource** Resources are the nurses or hospital employees assigned to the shifts to perform healthcare-related tasks. The resources can indicate their wishes before every schedule cycle, and most resources only work for a specific department at a time.

**Wishes** The wishes are the preferences that the nurses provide. There are multiple options for wishes. For example, a person can ask not to work on Wednesday evenings for the coming eight weeks. They can also give preference for a specific duty they want to perform on a certain date. A distinction can be made between required wishes and not required wishes. Some wishes should at all times be complied with, such as maternity leave or the use of specific holiday allowance days. Other preferences, such as working on a Monday night shift, are less critical.

### 3.1.2 Analysis data entities

Table 3.1 shows the number of observations for the different data entities. In the table, the departments that did not include any shift data in the database were not included in the column 'Number of entries'. The number of entries for the resources is the number of employees employed at the latest time that shift data was recorded in the database (01-09-2022).

The size of the departments varies greatly: a large department, such as the haematology department, has 165 active employees working at the beginning of 2022, while some small departments may only have one active employee working at the start of 2022. The number of duties may be misleading since only a fraction of the duties is used frequently, while often, a duty is created to meet the requirements of the specific case at hand. Only 17.893 duties were used more than ten times over all the departments, while the hundred most occurring duties account for almost 42% of all shifts. Because of the difference in the amount of data we have for each department, choosing what departments are selected for this research is not trivial. The following departments were selected because of their considerable size and because the nurse planners connected to this research are responsible for planning for these departments: haematology, radiology, and Intensive Care (IC).

| Entities | Number of entries |
|---|---|
| Departments | 600 |
| Duties | 4.077.349 |
| Shifts | 12.378.864 |
| Resources | 8.539 |
| Wishes | 655.648 |

TABLE 3.1: Analysis data entities

### 3.1.3 Analysis employees

This section describes the data for the workforce of the selected departments. Figure 3.2 shows graphically how the number of employees fluctuates over time. The number of employees for the departments is relatively stable except for the haematology department, which has a large number of employees at the start of 2020.

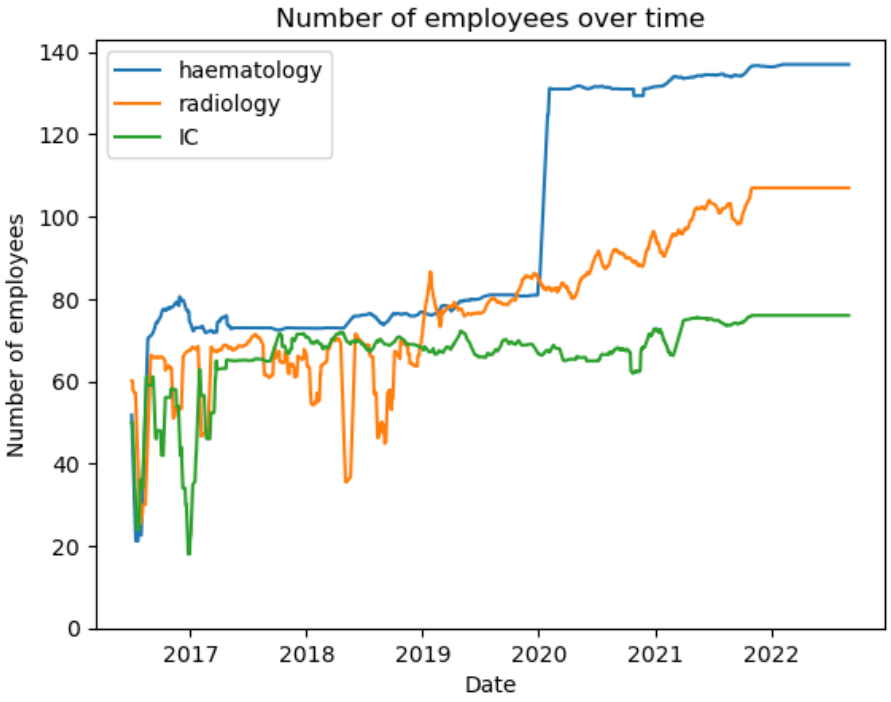



FIGURE 3.2: Number of employees over time

Table 3.2 shows the total number of resources that worked at the department, the number of employees active on 1-1-2022, the number of people added in the first three months of the planning horizon, the average number of days that a nurse is employed at the department in the department and the average number of shifts done. Note that the actual employment and number of shifts done can be higher than stated in the table since only the nurses' employment details were available from 01-04-2016 to 01-09-2022. Table 3.2 shows that the number of nurses is relatively stable over time and that there is a substantial number of shift data available on average for the nurses in the departments. There are, in total, 2344 days between the first and last observation day. The average employment number also shows that the length that the employees work for a certain

department is relatively stable and that the selected nurses do not change departments or jobs often during the time period.

| Specification | IC | haematology | radiology |
|---|---|---|---|
| Total nurses | 163 | 257 | 224 |
| Active contract on 1-1-2022 | 78 | 138 | 107 |
| Persons added between 1-1-2022 and 1-4-2022 | 5 | 0 | 5 |
| Average employment | 1721 | 1523 | 1245 |
| Average number of shifts done | 780 | 700 | 533 |

TABLE 3.2: Analysis of employees for selected departments.

### 3.1.4 Analysis shifts

This section focuses on analysing the shifts of the departments over time. Figure 3.3 shows the average number of shifts for a rolling time window of 30 days for the selected departments. The figure shows that the number of shifts every day is relatively stable. The spike in employees at the beginning of 2020 is also observed in the number of shifts done for the haematology departments, suggesting an expansion or a merger with a similar department. The data indicate seasonality in the number of shifts performed; therefore, the average number of shifts per day of the week and the average number of shifts per month are investigated in Figures 3.4a and 3.4b, respectively.

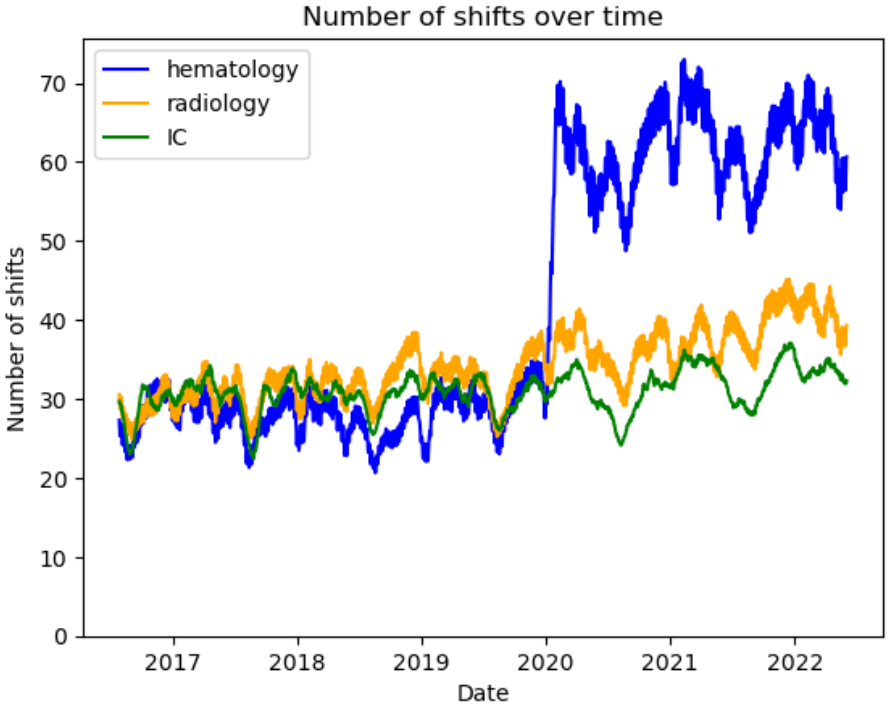
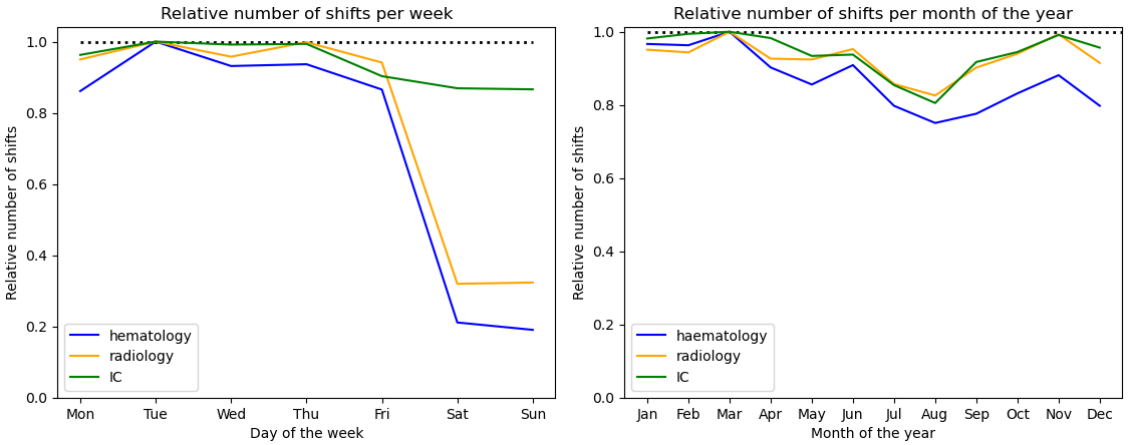



FIGURE 3.3: Number of shifts over time

Figure 3.4a shows the average number of shifts performed compared to the day with the highest average shifts performed per day. The data shows that, on average, fewer shifts are performed during the weekend, especially for haematology and radiology. The day of

the week is valuable information for the predictions and is there for selected as a feature in Section 3.2.

Lastly, the number of shifts over the year is investigated. Also, a seasonality trend is observed in this graph: fewer shifts are performed during the summer months than in other months. The nurse planners also confirmed this during the interview. Nurses use many of their holiday allowances during the summer months, and to compensate for the reduced number of shifts that are done in summer months, some extra shifts are performed in other months, especially the first months of the year.




(A) Relative number of shifts per day of the week

(B) Relative number of shifts per month of the year

FIGURE 3.4: Relative number of shifts per time unit

### 3.1.5 Analysis of wishes

The provided database contains all the wishes and the realised shift information from 1-04-2016 to 1-09-2022. Nurses provide their wishes in advance to the nurse planners. These wishes may reflect how the nurses want to be scheduled over time. The database stores five kinds of wishes: duty wish, work in interval wish, leave wish, recurring duty wish and recurring wish. The planning software also makes a distinction between required and not required wishes. Required wishes are wishes that the planners should always try to comply with, such as accepted holiday days or maternal leave.

If an employee wants to work a specific duty on a day, this wish is stored as a duty wish in the system. When this wish recurs, then this wish is stored as a recurring duty wish. Leave wishes are all required, such as the maternal leave wishes and the accepted holiday days. Work in interval wishes are wishes to work during a specified period of a day. Recurring wishes are wishes to work or not work on a day for a given time, length and frequency.

Figure 3.5 shows the number of wishes for the three selected departments along with the percentual share of the wish type compared to the total wishes. The duty wish is the most requested wish, with 65.7% of all wishes being duty wishes. The only required wish is the leave wish which only accounts for 2.1% of the total number of wishes.

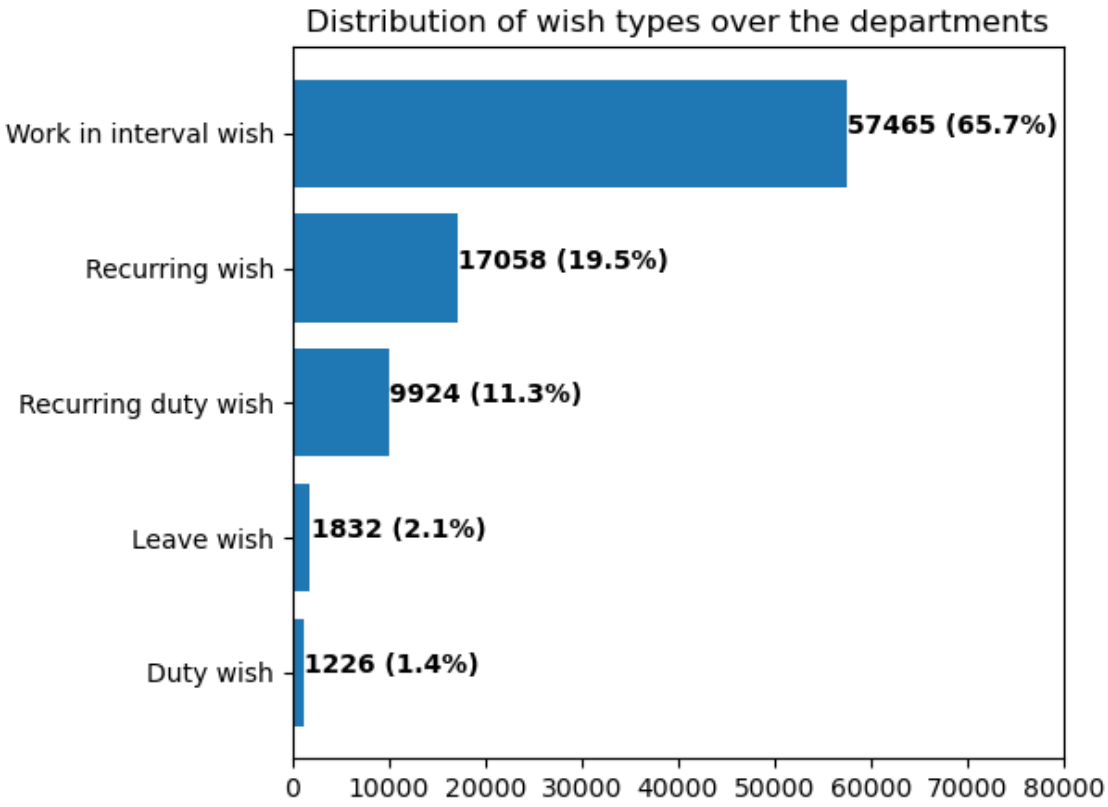



FIGURE 3.5: Distribution of the wishes in the database for the four selected departments.

## 3.2 Feature engineering

This section focuses on selecting and engineering the right features from the database to predict the nurse schedules. A feature describes some aspect of the studied data and is synonymously used as an attribute or variable in the ML domain (Dong & Liu, 2018). A feature may be the weekly availability of the nurse in the case of the NSP. Features form the foundation for data analytics by representing the values of the data objects in a feature-vectored space (Dong & Liu, 2018). There are three different feature engineering steps (Géron, 2019):

- Feature selection: Selecting the most relevant and useful features from available features in the dataset.
- Feature extraction: Producing more powerful features from the available features.
- Creation of new features by gathering new data.

This research focuses on obtaining the best set of features from the available data points; thus, the first two steps are the most relevant. A common saying in the computer science and mathematics community is "garbage in, garbage out". This phrase means that the input's quality determines the solution's quality. An ML model is unlikely to produce robust good results on the training set if the selected features are not relevant to the learning process (Géron, 2019). Also, the ML models are likely to underfit the problem if the representability of the features is not good enough. An important dilemma in selecting

the features for the ML model is the trade-off between the interpretability and accuracy of the results. Adding more features to an inherently non-linear model may improve the model's accuracy but may sacrifice simplicity and lead to overfitting. Kuhn and Johnson (2019) note that accuracy should never be severely sacrificed for a simpler model.

First, Section 3.2.1 describes the feature selection process. Then, Section 3.2.2 describes the feature extraction steps taken to create more powerful features. Lastly, Section 3.2.3 describes which variables are one-hot encoded to deal with categorical data.

### 3.2.1 Feature selection

The databases provided for this research contain much information, but not all the information is helpful for our prediction model. Based on the analysis of the data entities in Section 3.1 and the available data, the features are selected for the prediction. The analysis showed that the average number of shifts performed each day fluctuates over the weekdays and the month of the year; therefore, these features are included in the prediction model. Next, the wishes provide valuable insight into how the employees wish to be scheduled. The leave wishes are mandatory and should at all times be complied with. Other wishes, such as duty wishes, are less restrictive, but complying with non-restrictive wishes improves the schedule quality. Next to the wishes and the type of the day, we also have the weekly availability of the employees over time. The weekly availability of the nurses indicates the average number of shifts that the nurses want to work according to their contract. Figure 3.6 provides a summary of the feature selection. An interesting feature that was not available on time that could help improve the predictions is the skill level of the nurses and is therefore not included in the feature selection.

### 3.2.2 Feature extraction

The feature extraction phase commences after the initial features are selected from the database. The weekly availability of the nurses may provide insight into the number of hours a person wants to work. Still, more powerful features can be designed based on the person's work history over time and weekly availability. The following features are extracted after some feature engineering on the available data:

**Hours worked this week** Employees should not work many more hours than their weekly availability. Therefore a feature was added that stores the number of hours that this person has worked in the week starting from Monday.

**Hours worked last week** Since the hours worked last week may indicate the hours in the upcoming week, the hours worked in the last week is selected as a feature. It may be the case that the person has to compensate for his/her missing hours in the last week in the current week and that, therefore, more hours need to be worked in this current week.

**Hours left from availability this week** If a person has already worked all their hours in the week, the shift is more likely to be assigned to an employee that did not work many hours yet. For each day that the person was employed at the department, the hours left were calculated by subtracting the total number of hours worked to this day in the week from the weekly availability of the employee.

**Percentage worked weekday** This feature calculates how often the person worked on the weekday that we want to predict in the preceding thirteen weeks. If we want

to predict a Monday, this number then counts the number of times that a person worked in the preceding thirteen weeks on Monday. The idea for this feature is that a person that is scheduled often on a Monday has a higher chance of being scheduled on a Monday in the future.

**Lag variables** Lag variables store the information on what days the person worked in the past. We have included fourteen lag variables that store till two weeks back if a person worked or not. Adding more lag variables did not improve the performance. As an example, $T_{-1}$ stores if the person worked yesterday.

Section 3.1.5 showed that there are five types of wishes stored in the database: duty wish, recurring duty wish, leave wishes, work in interval wishes and recurring wishes. These wishes combined convey the preferences of the nurses over time and are stored in the database.

Some transformation was required to make sure that these wishes were stored correctly in the database and that ML models could be applied efficiently. Recurring and one-time wishes are stored differently in the database, but both convey the same message: a person has a wish for a specific day. The recurring wishes and recurring wishes duty wishes are converted to one-time wishes according to the frequency and the time that the wish is valid. If, for example, a nurse has a wish not to work every Wednesday for a period of four weeks, then this recurring wish is transformed into four one-time wishes to not work during the selected time period.

For some of the wishes, it is mandatory to comply with the wish because the planners are required to do so. An example is a wish to not work during maternal leave. We can make a distinction between the following three aspects of the wishes:

1. Wish for work or duty.
2. Wish to do the work or duty on a day or wish not to do it.
3. Requirement of the wish.

Eight possible wish types can be constructed from these aspects. Only no work wishes can be required, and therefore no work duty required, work duty required and work required do not exist. We obtain a list of five different wish types:

**Duty Work wish (DW)** A person prefers to do duty on a specified day.

**Duty No Work wish (DNW)** A person prefers not to do a specific duty on a day.

**No Work Required wish (NWR)** A person does not want to work on a specified day, and this has also been approved. The planners should at all times try to comply with these wishes.

**No Work Not Required (NWNR)** A person does not want to work on a day, but it is not mandatory to meet this wish.

**Work Not Required wish (WNR)** A person wants to work on a day. Work wishes are never required.

### 3.2.3 One-hot encoding

Some ML models may not work well with qualitative predictors selected for the prediction model. For example, the day variable is a categorical variable that is not encoded in a way that most ML models can work with this. One way to transform this variable is to use one-hot encoding (Gareth et al., 2013). One-hot encoding is a technique used in ML and data analysis to represent categorical data as numerical data. One-hot encoding transforms the categorical variable to a set of binary vectors for each possible category in the categorical variable.

One-hot encoding is applied to the features in Figure 3.6 that are underlined. Both these features have a qualitative response of more than two levels, so multiple dummy variables must be made for the encoding. Equation 3.1 shows how the months with 12 possible qualitative response variables can be broken into 12 dummy variables.

$$
\begin{aligned}
\hat{x_1^i} &= \begin{cases} 1, & \text{if Month i is January} \\ 0, & \text{otherwise} \end{cases} \\
\hat{x_2^i} &= \begin{cases} 1, & \text{if Month i is February} \\ 0, & \text{otherwise} \end{cases} \\
&\dots \\
\hat{x_{12}^i} &= \begin{cases} 1, & \text{if Month i is December} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.1}
$$

Figure 3.6 summarises the features that are used for the prediction. The variables that are one-hot encoded are underlined in the figure. The prediction model is trained for each department separately because the content of the features differs by department, and a different set of nurses need to be predicted for each department.

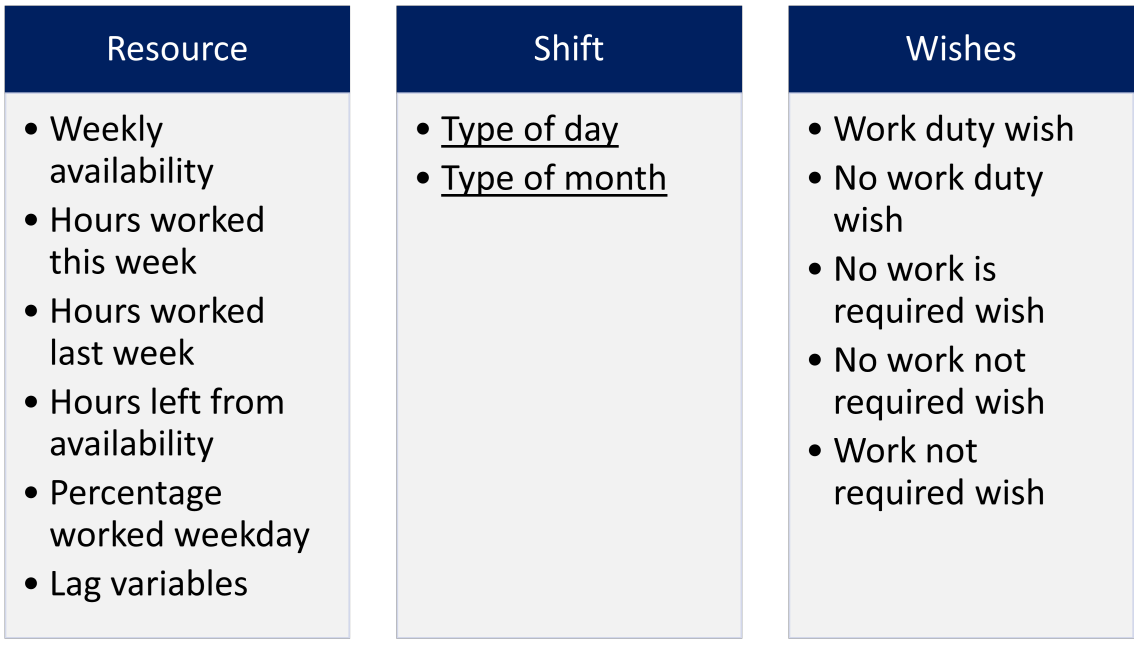



FIGURE 3.6: Feature selection per data entity

## 3.3 Last pre-processing steps

This last section discusses the last pre-processing steps performed on the database. First, the splitting method for the train, test and validation set is discussed in section 3.3.1. This step is required for our time-series data to tune the parameters for the selected ML models and evaluate them correctly. Section 3.3.2 describes the normalisation technique that scales the data to prevent ML models not working properly.

### 3.3.1 Train-test-validation split

It is common practice to split the available data into training and test sets. The purpose of the training set is to train the model, while the test set, also known as the hold-out set, can be used to evaluate the model's performance that the trained model has not seen before. Since the training model has not seen the data in the test set, the performance in the test set indicates how well the model generalises to new unseen data.

The validation set is another set of the total data used to provide an unbiased assessment of the model fit while tuning the model's hyperparameters. The validation and test sets are not part of the training data, but the difference lies in the function of the two sets. The validation set is used to tune the parameters for the ML model on an unknown dataset. The idea for the validation set is to estimate the generalisation performance of the ML model on unseen data. The idea behind holdout and cross-validation is to estimate the generalisation performance of a learning algorithm–that is, the expected performance on unknown/unseen data drawn from the same distribution as the training data. Tuning the hyperparameters on the training set instead of the validation set can lead to good performance on the training set but less ability to generalise to unseen data and, therefore, not work well for new data.

The testing of the performance of the selected models and optimised parameters is then done on the test set that has not been seen during the validation or training phase and therefore provides an unbiased evaluation of the performance of the ML model on unseen data. Validation sets can also help prevent overfitting in some ML models by early stopping, such as Gradient Descent. It is a sign of overfitting when the generalisability error increases in the validation set when tuning the hyperparameters. Stopping the tuning process prevents the model from overfitting the data (Prechelt, 2012).

Multiple ways exist to split the data into a train, test and validation set. Well-known possibilities are the random split, the stratified split in which the proportion of the classes in all the sets is preserved, and the time series split. The time series split is relevant for time series data in which the data is split based on the time of the observation, while the most recent data is put aside for the test set.

We can make use of Cross-validation (CV) to obtain more stable results. CV is a resampling procedure which is one of the most used methods to evaluate the generalisability of ML models (Arlot & Celisse, 2010; Stone, 1974). There are several ways to do cross-validation, but the most used is the k-fold cross-validation method. The data is split into k-folds, and then the model is evaluated to find the best generalisable parameters for k-splits. For every split, the best validation set is a different fold of the training data.

The k-fold method is challenging to implement for time-series data because of the inherent serial correlation present in time-series data (Bergmeir et al., 2018). A possibility is to use cross-validation on a rolling base for time series data (Hastie et al., 2009) or use k-fold blocked cross-validation. The available data contains information for 77 months, starting from 01-04-2016 to 01-09-2022. Figure 3.7 how the data is split in this research. Four splits were selected with a test set equal to three months for the evaluation of the

models. In this way, the model performance is tested on the most recent one year of information. The validation set is set to be equal to the length of the test sets in the splits. The training set is all the months starting from the first observation month to month 62 and also includes months between 66 and 74, depending on the split. The IC and haematology departments underwent significant changes during the pandemic, as illustrated in Figure 3.2. For both these departments, the training data started at month 45, corresponding to 01-01-2020, with all the rest kept the same. Experimentation showed that this change indeed enhanced the model performance, as the training data better matched the current situation of the departments.

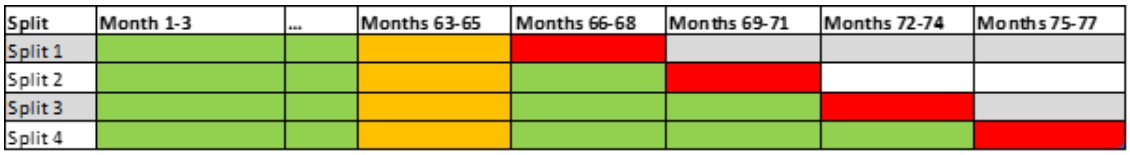



FIGURE 3.7: Rolling time horizon as cross validation

### 3.3.2 Feature scaling

One of the most important transformations in data pre-processing is data scaling. Since the range of values differs per feature, an ML model can unfairly give more weight to a feature that contains, on average higher values than a feature with lower values. Feature scaling can transform the values in such a way that this is prevented.

The two primary feature scaling techniques are min-max scaling and standardisation (Géron, 2019). Min-max maps the values on a zero-to-one scale, while standardisation maps the values on a unit scale. Standardisation is less sensitive to outliers but does not scale well with neural networks. Since the data does not contain many outliers or these have been removed during the pre-processing steps, the min-max transformation has been selected as the choice for the feature scaling. Performance testing also showed that this method resulted in better performances than min-max transformations.

## 3.4 Conclusion

This chapter performed the pre-processing steps to enable the ML models to work. Based on the analysis of the available data, the departments and the features for this research were selected and extracted. These features are based on the working history of the nurse, wishes, type of day and the weekly availability of the nurse. A 4-fold cross-validation method was selected for the evaluation of the performance and a min-max transformation for the feature scaling.

# Chapter 4

# Solution approach

This chapter describes the solution approach for the classification problem presented in Chapter 3. The approach to creating a feasible solution based on historical planning data is two-fold. First, Section 4.1 describes the type of classification and the simplifications that we have taken. Then, Section 4.2 presents the selected ML models fitted on the pre-processed data to make predictions for the test set. A description of the selected models and an explanation of why the models have been selected based on the criteria in 4.2 are presented in Section 4.3. The performance testing of the ML model is discussed in Section 4.4. Section 4.5 describes the improvement heuristic design and experimentation. The findings of this chapter are summarised in Section 4.6.

## 4.1 Classification approach

This first section describes what kind of task we aim to perform with the ML models. Section 4.1.1 describes the type of classification that we use for this research. Then, 4.1.2 gives the description of the prediction type and the simplifications taken.

### 4.1.1 Classification format

There exist three types of classification problems: binary classification, multi-label classification and multi-class classification. In a binary classification problem, an instance belongs to one class, and every instance should be predicted to be one and only one label in the set of possible labels. Binary classification is the process of classifying data points in either one of the two possible predetermined classes (Kumari & Srivastava, 2017). The multi-class classification problem is similar to binary classification, with the only difference being that more than one possible label exists. Lastly, multi-label classification involves the job of classifying an observation into a subset of possible labels.

The prediction model should predict whether a person will be working on a day or not work on a day. The classification can both be seen as a single-label classification problem for which every person is considered individually on a day and predicted if this person is going to work or not, or as a multi-label problem in which for every day, we want to predict what subset of nurses is going to work. Initial testing showed that single-label classification has superior performance to multi-label classification; therefore, the problem is considered a single-label problem.

### 4.1.2 Scheduling approach and simplifications

The dynamic nature of hospitals and the subtle preferences of the nurses makes the planning process difficult. This section describes the scheduling approach settings based on the actual planning process of hospital X and the available data we have. Some simplifications and assumptions are made to ensure that a roster can be predicted correctly for the unknown future.

The planning for the nurses is made in advance for every quartile. The prediction time horizon is, therefore, also set to be three months, starting from the first day of the quartile. Some simplifications are taken because the ML models may not be able to deal with unforeseen events. For every prediction, only the nurses who had a valid contract during the prediction time horizon are considered. Also, to ensure enough data to predict the nurses, the nurses employed for less than 90 days or had done less than 30 shifts in total are excluded from the dataset. In this research, only the employees who work for the selected department are included for every department where ML is applied. A small subset of the nurses that work for a department may come from a flex pool of nurses that exists to fill up gaps in the capacity for the shifts in a department when needed. Lastly, some nurses, such as nurse students, are scheduled above the capacity of the day. These students are scheduled on the shifts but are not required for the successful operation of the nurse activities and are only employed for a short time frame at each department.

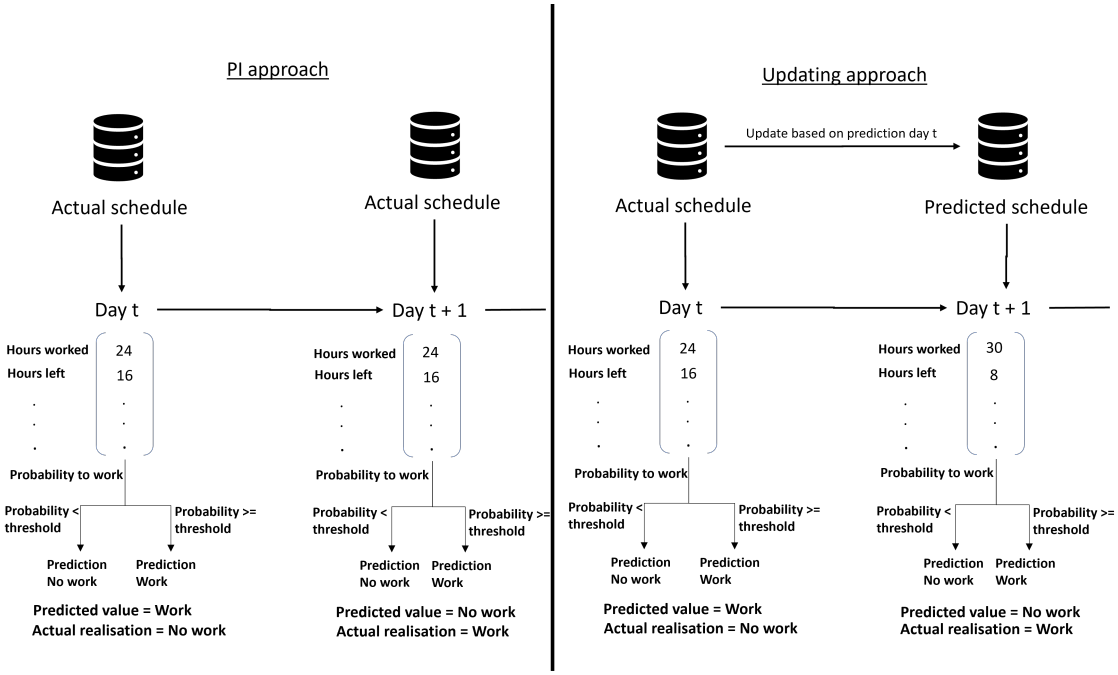



FIGURE 4.1: Difference between PI and updating approach

The prediction model classifies over the prediction horizon if the person is working. We consider two different prediction methods: Perfect Information (PI) and updating approach. Both methods predict for the selected set of nurses if they are either working on a day or not based on the probability output of the Machine Learning model. To determine which nurses will be selected, a threshold value is calculated based on the expected number of nurses needed for that day. The threshold is set to select the nurses with the highest probability, such that the total number of nurses predicted to work on that day matches the expected demand.

Figure 4.1 shows the difference between the PI and the updating approach for a single nurse. This figure shows two of the total prediction features that were used. In this visualisation, the day $t$ corresponds to the first day of the prediction horizon. The first day is the same for both approaches, and they both use the actual realised values to make a prediction. The difference lies in the feature values that are used after the first prediction. In the case of the PI approach, the features for $t + 1$ are based on the actual realised values from the actual schedule realisations, while the values are updated in the updating approach based on the prediction made previously. In the toy example in figure 4.1, the feature values for hours worked and hours left are not updated in the PI approach because the actual realisation for day t was that this person did not work. Since the ML model predicted that the person would work, the values are updated accordingly for the day $t+1$.

The nurse planners know before the planning is made what the number of required personnel is on a day. In Section 4.3, various Machine Learning models are chosen, which generate the probability of an individual working based on the given feature values. We utilize the daily capacity information by identifying and selecting the group of individuals with the highest probability of working to fulfil the required number of personnel. To illustrate, suppose we have the knowledge that forty nurses are needed to work on a particular day. Based on the outcome of the ML model of the probability to work, the forty nurses that had the highest probability of work are then predicted to work.

## 4.2 ML model selection process

This section describes how the ML models are chosen for this research. First, Section 4.2.1 describes the No Free Lunch theorem that states that it is impossible to know the best model without any knowledge of the data and problem context. Next, in Section 4.2.2, the considerations for the model selection in Section 4.3 are listed.

### 4.2.1 No Free Lunch

The No Free Lunch theorem, first coined by Wolpert and Macready (1997), purports that without the knowledge of the problem description or the data at hand, it is impossible to tell what predictive model performs the best. Furthermore, a model optimised for collinear predictors is not necessarily better on data with collinear predictors than other models if the data is constrained to linearity and if the model is sensitive to missing data values (Kuhn & Johnson, 2019). Research has been done on which models perform on average better than others, most notably Demšar (2006) for classification and Fernández-Delgado et al. (2019) for regression problems. Although some model types are more likely to produce better results than others, the difference is not enough to always resort to these models. Practice shows that the best approach for model selection is to use dissimilar models on the data and check by trial and error which model performs the best on the dataset at hand (Kuhn & Johnson, 2019).

### 4.2.2 Model considerations

The NFL theorem states that it is impossible to tell which model best suits the problem description if the data and information about the problem are missing. Depending on the problem task at hand and the data structure, we can select which models may be most appropriate for this research. The following considerations are important when selecting the set of suitable models for our ML models (Kinha, 2022):

- Data size: More data can generally lead to more reliable predictions. In the case that the data has more features than observations, which can be the case for tasks such as text classification, the model to be selected should have a high bias-to-variance ratio such as linear regression or Naive Bayes. In contrast, if the training data is sufficiently large, which is the case for this problem, ML models with a low bias to variance are more appropriate such as KNN, decision trees and kernel SVM.

- Accuracy versus the interpretability of the outcomes: Some models inherently have more interpretability than other models because of the approach to classifying the objects. In general, the higher the interpretability of the model, the worse the model's accuracy.

- Required speed and accuracy: Models that do not require much tuning of the training data are generally easy to implement and can lead to faster algorithm building and run speed. Some models require more tuning of parameters, such as the SVM method or have high convergence time. Simpler models are preferred for the prediction since the situations in the hospital may change quickly, and the best-found solution found may thus also change. However, because of the complexity of the problem, it may be necessary to implement more advanced models.

- Linearity of the model: a significant subset of the existing ML models is built upon the assumption that a straight line can separate classes. These models perform naturally well for linear data. Linearity can be checked by applying and evaluating the performance of a model such as SVM. If the SVM model performs well, then there is probably a linear trend in the data and other models that assume linearity may also perform well.

- Ease of obtaining predicted probabilities that a class belongs to an instance. The ability to obtain predicted probabilities that a given instance belongs to a particular class is an important consideration when selecting an ML model. While many models can output such probabilities, not all of them have this capability, or it may be too time-consuming to retrieve the probabilities. It is required that the selected models can easily output the predicted probabilities since we use the probabilities of Machine Learning to predict nurses based on the required capacity per day.

## 4.3 Model selection

This section describes the selection for the model approach. Five models have been selected that suit the problem description and type of data: Logistic Regression (LR), Gradient Boosting (GB), K-Nearest Neighbours (KNN), Random Forest (RF) and Artificial Neural Network (ANN) are explained in that order in this section. The reason for inclusion based on the available data and the model considerations in 4.2.2 are explained in each section. The parameters for the models are selected based on a grid search. The reader is referred to Appendix A for all hyperparameter tuning results. Table 4.1 shows the used Python packages and hyperparameters used for the selected models.

| Model | Package | Hyperparameters |
|---|---|---|
| LR | Sk.learn.ensemble LogisticRegression | solver = 'saga', penalty = 'l2' , C = 0.1 |
| GB | Sklearn.ensemble GradientBoostingClassifier | n_estimators= 3000, learning_rate = 0.1<br>max_depth = 4, min_samples_leaf=0.1 |
| RF | Sklearn.ensemble RandomForestClassifier | n_estimators= 2000, max_depth= 5<br>min_samples_leaf= 0.1, class_weight='balanced subsample' |
| KNN | Sklearn.neighbors KNeighborsClassifier | n_neighbors=20, weights='uniform' |
| ANN | Keras | epochs=10, batch_size=10<br>neurons input layer = 25, neurons hidden layer = 100<br>dropout rate = 0.2, optimiser = Adamax<br>batchsize = 10, epochs = 100 |

TABLE 4.1: Selected hyperparameters models

### 4.3.1 KNN

KNN is a simple non-parametric, powerful way to classify data points based on the distance between the observations in the feature space. KNN is one of the most broadly used ML classifier models attributing to the simplicity and intuitiveness and that no assumption is required about the underlying data distribution (Imandoust & Bolandraftar, 2013). KNN is relatively slow for predictions but very fast in training on the train data. Since KNN is a non-parametric model, it can deal with complex relationships between the in- and output of the variables. The KNN algorithm classifies new observations by generally taking the mode of the K number labels of the closest to this new observation without using distance-based weighting (Guo et al., 2003). KNN is a lazy learner, meaning that the calculation for the assigned label is deferred until the classification. Since the cost of making new classifications is high, the KNN is useful for applications where accuracy is important, and new predictions don't have to be made generated (Guo et al., 2003; Zhu et al., 2014). KNN is susceptible to differences in feature scales. Normalisation can therefore increase the performance significantly. In general, the min-max normalisation gives better results for the KNN than the Z-score transformation (Henderi et al., 2021; Pandey & Jain, 2017).

A mathematical description of the working of KNN is given in Equation 4.3.1. We denote the $x$ as the feature and y as the label. The goal for the KNN is to find a function $f : X \to Y$ that links the correct label y to the new observation x. There are plenty of options for selecting the distance metric, but the Euclidian distance is used the most. The distance d between data points $x$ and $x'$ with n features based on the Euclidian distance is shown in Equation 4.3.1 below with $x_n$ being the value of the $n^{th}$ feature of $x$.

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2} \tag{4.1}$$

The label is then selected based on the most occurring label in the set of K closest labels to the new observation. When K is initialised to 1, new observations are labelled the same as the first closest observation point. Figure 4.2 gives a graphical representation of how KNN works using Euclidian distance. In this example, the green checkmarks correspond to a person working and the red crosses to not work. The question mark indicates the observation that we want to classify based on the hours worked this week and the weekly availability. If K is equal to three, the KNN classifies the new observation as work since two of the three closest neighbours are of this class. Things change when we set K equal to six since this would label the new instance as no work since four of the seven closest neighbours are red crosses.

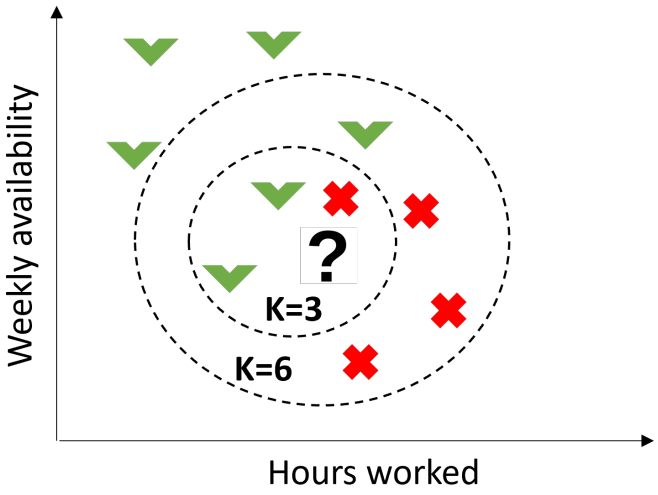



FIGURE 4.2: The KNN classification based on k=3 and k=6

The value selected for K is not trivial and impacts the model's reliability. There are many ways to evaluate a good value for K, but the simplest is to do a trial and error for different values of K and select the one that performs the best. K = 1 can result in overfitting and result in high sensitivity for noise in the data. On the contrary, setting the value to a large value can increase the robustness of the model but also comes with an increase in the computer power required.

### 4.3.2 Logistic regression

Logistic regression can deal with large datasets but may have difficulty working with high-dimensional data. Logistic regression has a high interpretability since the output of the predictor is the probability that a class belongs to an instance. Using the logit transform, the partial effect of changing the value of a variable on the outcome can be analysed. Logistic regression is also fast to train but may not capture complex relationships in the data since it is a linear model.

Logistic regression, also known as logit regression, is a method that can be used for classification purposes. Logistic regression uses a linear combination of predictor variables to approximate the probability that an instance belongs to a class. Logistic regression has similar components to linear regression but is appropriate for problems where the dependent variable is dichotomous (Kurt et al., 2008).

Logistic regression outputs a probability that data points belong to a binary class. Let us denote $\hat{p}$ as the probability of the result of the classification model in LR. The probability is a sigmoid function of the weighted sum of the input features and a bias term. Equation 4.2 shows the formula in a vectorised form that classifies observations, in this case, as a 1 or 0 based on the values of the input features.

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{p} = \sigma(\theta^\intercal \mathbf{x}) \geq 0.5, \\ 0, & \text{if } \hat{p} = \sigma(\theta^\intercal \mathbf{x}) < 0.5 \end{cases} \tag{4.2}$$

Figure 4.3 provides an example of both a linear regression model and a logistic regression model that are fit on a sample nurse rostering problem based on only one feature: hours left in week from weekly availability. The figure shows that the fit of the linear regression is worse than the LR fit since the linear regression only classifies a person to be working

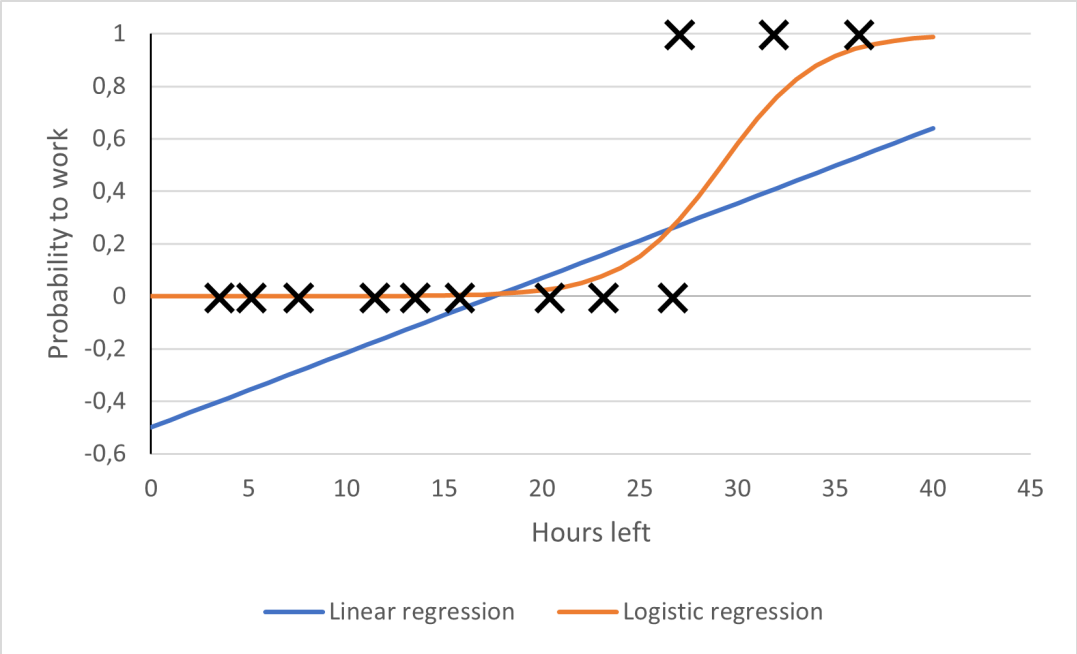



FIGURE 4.3: Comparison between a linear regression model and a logistic function.

when the person has at least 35 hours left. The logistic regression, however, better fits the data since it classifies more points in the training set correctly.

This sigmoid function in 4.2 ensures that the probability is S-shaped and bounded between 0 and 1. The $t$ in the Equation is called the logit and is the log of the ratio between the estimated probability for the positive class and the negative class estimated probability. The name logistic regression is derived from this applied logit function.

The logistic regression model is fitted on the data using a log loss function. The log loss function measures the sum of the costs function for all the individual training instances. The cost function for a single observation has the following form:

$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1 \\ -\log(1-\hat{p}) & \text{if } y = 0 \end{cases} \tag{4.3}$$

The cost function can be interpreted as follows: the cost is high when a low probability is assigned to a positive class and relatively low when the assigned probability is close to 1. The opposite holds for classifying negative classes. The log loss minimises the average of these costs over all training instances. Formula 4.4 shows the calculation for the log loss $L$. The log loss uses the cost function from the equation and adds all costs from the m observations. It then takes the negative average of this sum since we want to find the value for $\theta$ that minimises the error over all the observations.

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \log(\hat{p}^i) + (1-y^i)\log(1-\hat{p}^i)\right] \tag{4.4}$$

There is no closed-form equation that finds the optimal value of $\theta$, such as the normal equation for linear regression. It is, however, possible to find the global minimum of the function with any optimisation algorithm with the right learning rate in the finite time since the cost function is convex.

### 4.3.3 Random Forest

Random Forest (RF) models can handle large datasets, although they are not the fastest to train. A helpful feature of the Random Forest algorithm is that it can output the feature importance of the features that they are trained on that can be useful for interpretation. Random forests are prised for their high accuracy and can also capture the complex relationship that may be present in the provided data. Lastly, Random Forest is mostly equipped for complex databases and nonlinear data.

Random Forest is an ensemble of decision trees. First, the notion of ensemble learning is explained, and hereafter, a description is given of the working of a decision tree. Ensemble methods are learning mechanisms that use a weighted voting system for the outcome of a set of classifiers to classify new data points (Dietterich, 2000). Often, the ensemble method's aggregated solution is better than the best single classifier in the ensemble. The aggregation of multiple predictors is known as an ensemble, hence the name ensemble learning.

The best performance for ensemble methods is achieved when the individual predictors are as independent of each other as possible. An ensemble of weak learners (predictors that only classify slightly better than random guessing) can still lead to a strong aggregated learner, given that the sufficient number of weak learners is diverse (Géron, 2019). There exist many options for ensemble learning. Two well-known options for ensemble learning are the soft-voting and the hard-voting methods. Soft voting often leads to more reliable results, but it requires that the individual learners can give a class probability output.

We can also distinguish between the sampling method for the individual predictors. Predictors are often trained on random subsets of the data. When this sampling is done with replacement, the ensemble method uses a bagging approach, while sampling without replacement is called pasting. Bagging often leads to better results since the predictors are less correlated. Random forest often uses a bagging method. If the dataset contains many features, one can use the random subspaces method, which results in more predictor diversity (Ho, 1998). In this way, the predictors are given a subset of the features for the selected random sample, which increases the bias but lowers the variance.

A decision tree is an ML model that can tackle both classification and regression problems and can deal with complex datasets. Furthermore, decision trees are not susceptible to unscaled data and require little pre-processing. A decision tree classifies into branch-like segments, with the top being the root node. Figure 4.4 shows how a decision tree for the NSP. In this oversimplified example, the prediction that someone is going to work is based on the values of two features. The root node splits the data up if the person wishes to not work on a day. If the person had a wish to not work, the person is predicted to not work on that day. If the person did not have a no work wish but already worked for at least forty hours during the week, the person is also predicted to not work.

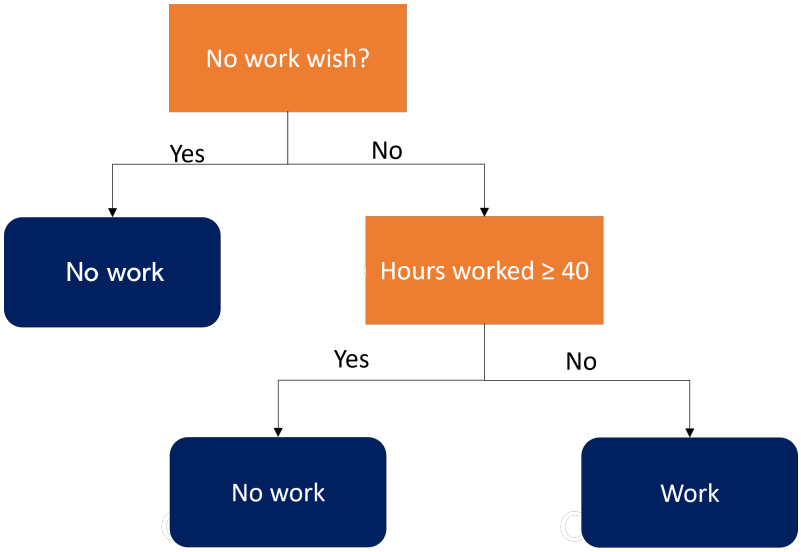



FIGURE 4.4: Example of a Decision Tree for the NSP.

As can be seen in Figure 4.4, decision trees have high interpretability, and a new sample can even be classified manually when the decision rules from the decision tree are known. There are various ways to calculate how the decision should split the nodes, but the performance is for these methods often very similar (Géron, 2019). The most used technique is based on minimising the impurity of the child nodes.

The decision tree algorithm tries to split the training set into two subsets based on feature $k$ and threshold $t_k$ such that the impurity of the child nodes is minimised. The split that minimises the sum of the left and right child nodes is selected as a split by the decision tree. Equation 4.3.3 shows the split process based on minimising impurity.

$$\min I(k, t_k) = \frac{n_{\text{left}}}{n} G_{\text{left}} + \frac{n_{\text{right}}}{n} G_{\text{right}}, \tag{4.5}$$

where:
$G$ = the Gini index
$n$ = the number of observations

### 4.3.4 Gradient Boosting

Like Random Forest, GB is an ensemble method that can deal with large datasets and is effective on high-dimensional data. GB has proven to be successful in many domains. GB is another ensemble method that trains on decision trees. can usually outperform Random forests (Hastie et al., 2009). However, some interpretability is sacrificed for accuracy, making it more difficult to interpret the decision-making process. This non-linear model does not make assumptions about the dataset and can understand and capture complex relations in the data, although training with GB can be computationally expensive. Lastly, GB also outputs the predicted class probability.

GB is a popular ML technique that can perform both regression and classification problems. While RF builds an ensemble based on independent deep trees, GB is built on shallow weak trees, with each tree improving on the errors of the previous trees. GB originated from the idea of creating an efficient algorithm to create a strong learner from weak learners (Kearns, 1988).

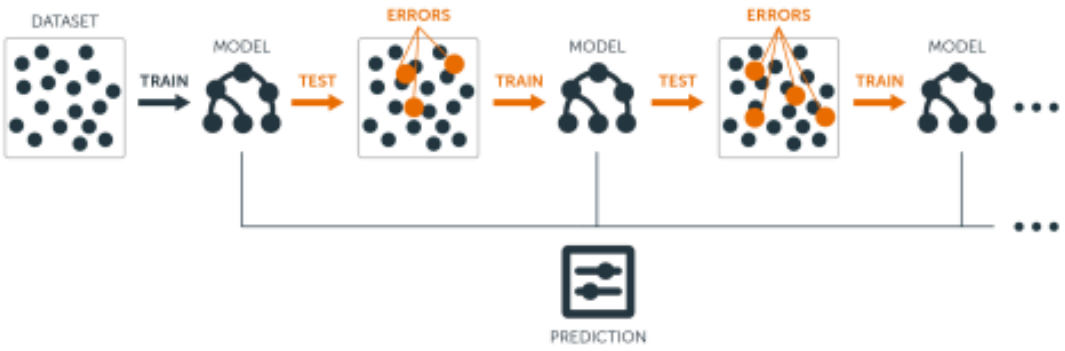



FIGURE 4.5: Example of GB from "Gradient Boosting Machines" (2023)

The trees are constructed in a greedy way and are split on purity scores such as the Gini index, as shown in Equation 4.3.3. The objective of the GB is to minimise this loss function for the overall model. It does so by adding weak learners in an iterative way.

GB is a popular machine-learning technique used for solving regression and classification problems. It involves the creation of a strong learner by combining multiple weak learners. Each weak learner is trained on a subset of the data and a specific loss function, typically a decision tree. The objective of GB is to minimise the loss function of the overall model by iteratively adding weak learners to the model. Each sub-subsequent learner is trained on the residual error of the previous learner (Géron, 2019). GB stops when the training error is minimised, or a stopping criterion is reached.

Figure 4.5 shows how GB works. The decision tree in the first iteration is based on the training data. Then the next decision tree is fitted based on the residual errors from the first decision tree and so forth. The output of the GB is the ensemble of all the decision trees together.

### 4.3.5 Neural Networks

Neural Network (NN), also known as Artificial Neural Networks (ANN), is an ML technique for which human neurons inspire the name and structure. ANN can be used for various tasks, including binary classification. ANN are powerful models that can learn and model relations between input and output data that are nonlinear and complex. They work specifically well on complex data and can also deal with unstructured data such as text or image data. Increasing the size and complexity of the network architecture may lead to better performance for complex and large datasets but may lead to overfitting and leads to longer training times.

Neural networks consist of layers of interconnected nodes knowns as neurons. We can make the distinction between the input layer, a set of hidden layers and an output layer. The input layer sends the input information to the hidden layers that use weights and biases to transform the data to fit the output train data. Nodes work similarly to a linear regression model: every node consists of an input weight, bias and output. Equation 4.6 shows the formula for the calculation of the output, while the output of the node is given in 4.7.

$$\sum_{i=1}^{n} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + bias \tag{4.6}$$

$$f(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i + bias \geq 0 \\ 0 & \text{if } \sum_{i=1}^{n} w_i x_i + bias < 0 \end{cases} \tag{4.7}$$

ANN generally consists of two phases: forward propagation and backward propagation. Input data is fed through the network during forward propagation, and the output of the nodes is calculated. Then, in the backward propagation phase, the weights and biases are changed to minimise the difference between the predicted output of the forward propagation phase and the actual labels. This process is repeated until a stopping criterion is reached.

There exist many forms of NN that each have their neural network design and use case. In this research, the basic ANN model is considered that only comprises input, output and hidden layers.

## 4.4 Performance testing

The performance of classification models can be tested in several ways, but the applicability of the evaluation metric depends on the problem-specific context (Entezari-Maleki et al., 2009). Popular evaluation metrics for a single-label classification problem include accuracy, precision, recall and f1-measure (Sebastiani, 2002). Also, the Receiver Operator Characteristic curve (ROC) is a graphical plot showing a binary classifier's diagnostic ability while varying the discrimination threshold. The ROC curve plots the True Positive Rate (TPR) against different False Positive Rate (FPR) for different classification thresholds (Mandrekar, 2010). Confusion matrices plot the classification performance of all labels in one table.

Section 4.4.1 explains the confusion matrix, which is a table that summarises the classification performance of all labels and explains the concepts of recall, precision, accuracy and f1-score. In Section 4.4.2, we describe the ROC curve. Hereafter, more problem-specific metrics that are designed for this research are discussed. The wish score that measures the adherence of the schedule to the known wish types is discussed in Section 4.4.3. We also evaluate the predicted schedule to ensure it meets the required capacity every day. We discuss the evaluation of missing capacity in Section 4.4.4. Lastly, Section 4.4.5 shows the hard constraint violations.

### 4.4.1 Confusion matrix

A confusion matrix summarises the prediction performance of a classification model by comparing the predicted results with the actual labels. In the case of binary classification, the matrix is a two-by-two table that contains the following four categories: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). TP represents the number of instances that the observation was classified as positive when the actual label was positive. FP represents the number of instances for which the observation was classified as positive when the actual label was negative. FN represents the number of instances for which the observation was classified as negative when the actual label was positive. Lastly, TN represents the number of instances for which the observation was classified as negative when the actual label was negative. Figure 4.2 shows the framework for the confusion matrix, where $p$ stands for work and $n$ for the not working label.

Popular evaluation metrics for a single-label classification problem include accuracy, precision, recall and F-measure (Sebastiani, 2002). These can be constructed from the performance of the categories in the confusion matrix.

| | | **Prediction outcome** | | |
|---|---|---|---|---|
| | | **p** | **n** | **Total** |
| **Actual value** | **p′** | True Positive | False Negative | P′ |
| | **n′** | False Positive | True Negative | N′ |
| | **Total** | P | N | |

TABLE 4.2: Example of Confusion matrix.

1. $Recall = TP/(TP + FP)$: measures the number of positive instances predicted correctly compared to the total number of positive instances.

2. $Precision = TP/(TP + FN)$: measures the number of positive instances predicted correctly compared to the total number of the predicted positive class instances.

3. $Accuracy = (TP+TN)/(TP+FN+FP+TN)$: measures the proportion of correct predictions. The accuracy can be calculated by dividing the total number of correctly predicted labels by the total labels in the test set.

4. $F1 - score = 2TP/(2TP + FN + FP)$: measures the harmonic mean of precision and recall. While the accuracy may give a good indication of the proportion of correctly identified labels, the f1-score can give a more balanced representation of the performance by taking class imbalance into account.

### 4.4.2 ROC curve

The Receiver Operating Characteristics curve (ROC) is a graphical plot showing a binary classifier's diagnostic ability while varying the discrimination threshold. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rates (FPR) for different classification thresholds (Mandrekar, 2010). The ROC curve helps understand the diagnostic ability of a binary classifier when the discrimination threshold is varied. Lowering the discrimination threshold results in more observations being classified as positive, thereby increasing the FPR and TPR.

The Area Under the Curve (AUC) measures the integral under the ROC curve. This measure gives an aggregate evaluation of the performance of the binary classifier for all possible threshold values. The best possible value is 1 for the AUC when the classifier can distinguish objects perfectly, while a random classifier has an AUC of 0.5 and classifiers worse than average have an AUC of less than 0.5. Figure 4.6 provides an example of a ROC curve. A binary classifier that classifies labels randomly converges to the random red classifier for an infinitely large test set. In the case of a perfect classifier, the TPR rate is 1, and the sensitivity is equal to 0, resulting in a line that goes from coordinates (0,0) to (0,1) to (1,1).

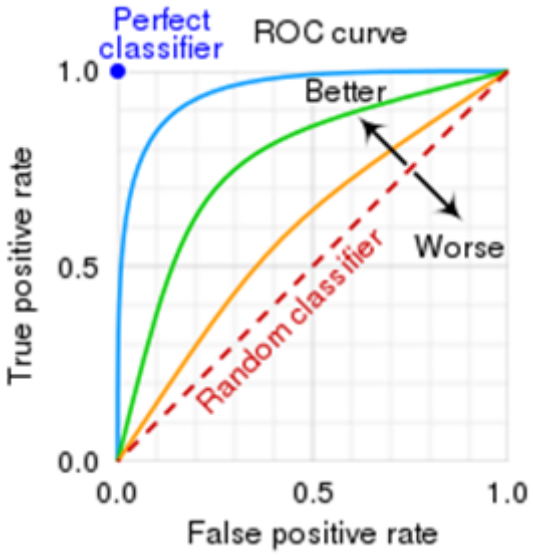



FIGURE 4.6: Example of ROC curves from Wikipedia contributors (n.d.).

### 4.4.3 Wish score

The ML models are also evaluated based on the wish score. The wish score is defined as the percentage of the number of wishes that are adhered in the ML model compared to the total number of wishes that are present in the selected time horizon for a specific wish type. As an example, when we have hundred duty wishes and the ML model schedule adheres to 50 of those duty wishes, then the wish score is equal to 50%. The wish score is evaluated for the five types of wish scores along with the total wish score.

### 4.4.4 Missing capacity

The database contains information about the required number of nurses per duty. The required number capacity per day is approximated by finding the set of duties that need to be performed on a day. The sum of the duties times their respective required capacity results in the minimum number of nurses that are required on a day. We assume that this is the minimum number of nurses that are required to work in a day.

The predicted schedule should, in the optimal case, be as close as possible to the actual schedule while at the same time meeting the minimum required capacity per day. This is also one of the two parts of the objective function discussed in Section 4.5.2.

### 4.4.5 Hard constraint violations

The next important indicator of the schedule's fitness is measuring the hard constraint violations. We consider five important constraints for the feasibility of the schedule. These constraints are valid for hospital X, and variants of these constraints are also used at other hospitals. Each constraint is denoted by $c$ followed by a number that identifies the specific constraint.

**Weekends off constraint** ($c_1$) Nurses should have at least two weekends off per 6 weeks.

**Max shifts in a row violation** ($c_2$) A maximum of six shifts in a row can be performed.

**Max shifts per week** ($c_3$) A maximum of four shifts can be performed by resources with a weekly availability of fewer than 20 hours per week.

**Required no work violation** ($c_4$) A person is not allowed to be scheduled on days that this person has a required no work/leave wish.

**Max hours per four weeks** ($c_5$) A person can not work more than 160 hours every four weeks on average.

## 4.5 Improvement heuristic

Section 2.3.4 describes the choice for Simulated Annealing (SA) as an improvement heuristic for the predicted schedule. SA is a metaheuristic metaheuristic that has shown promising results for the NSP (Kuhn & Johnson, 2019; Turhan & Bilgen, 2020). SA works by randomly perturbing the current solution using some kind of operator and then evaluating if the solution should be accepted based on the current temperature and the difference between the previous and current solutions.

First, 4.5.1 gives the overall framework for the improvement heuristic approach. Next, Section 4.5.2 describes the objective function that determines the value of the Simulated Annealing (SA) solution. Since the starting solution should be a feasible solution, the initial solution is made feasible via an algorithm. This is explained in more detail in Section 4.5.3. The operators to create neighbouring solutions are discussed in Section 4.5.4. Lastly, the experimentation for the improvement heuristic design is discussed in Section 4.5.5.

### 4.5.1 Simulated Annealing framework

The improvement heuristic framework is presented in Figure 4.7. The algorithm starts with a prediction solution from the ML model for our problem. This prediction is first made feasible by the algorithm in Section 4.5.3. The values of the SA are now initialised. The initial values for the SA approach impact the quality of the solution obtained. The experimentation of these values is discussed in Section 4.5.5. Then a neighbour solution is created. Section 4.5.4 discusses the neighbour operators. Since the solution may not violate any constraint, we first check if the operator introduces any violations. The solution is thrown away when any violation of the hard constraints and the process of generating a new neighbour solution is started again. When a feasible solution has been found, the delta objective is calculated. Better solutions are always accepted. When the solution value of the neighbour solution is worse than the current solution, there is still a chance to accept the solution with $1 - e^{\left(\frac{\Delta f}{T}\right)}$ probability. The higher the temperature and the lower the difference between the neighbour solution and the current solution, the higher the probability of being accepted. The temperature is cooled down with factor $\alpha$. The SA stops when the stop temperature is reached, and the best-found solution is returned.

At the start of the algorithm, the temperature should be relatively high, such that the worse solutions can be accepted, and the algorithm performs enough diversification to prevent getting stuck in a local optimum. The temperature wears off after every iteration, lowering the probability of accepting a worse solution than neighbouring solutions.

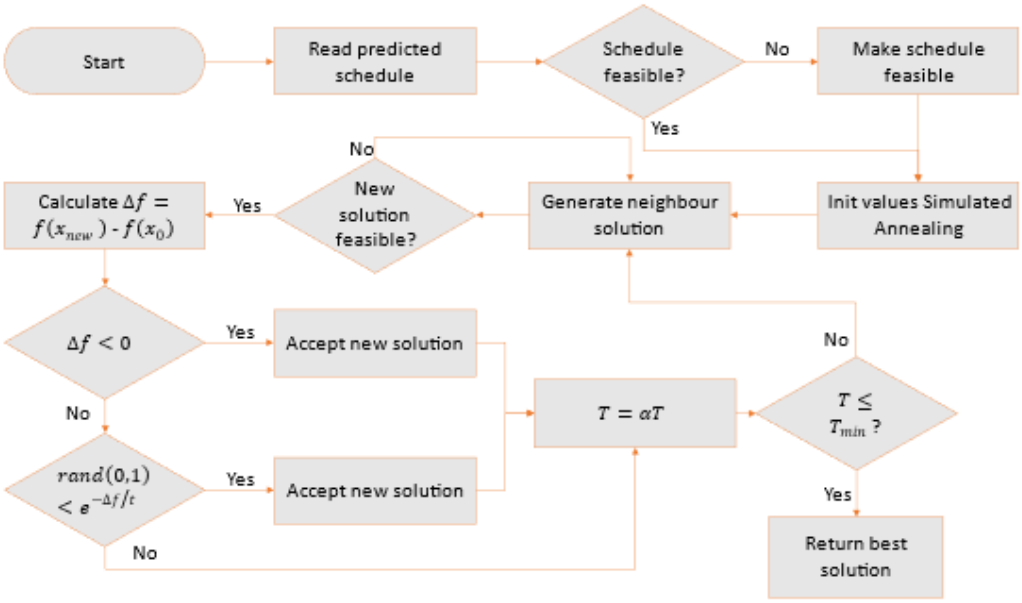



FIGURE 4.7: Flowchart of the SA approach.

### 4.5.2 Objective function

This section gives the objective function of the SA approach. The objective has two parts: the first part of the objective function calculates the sum of missing nurses each day. Hospital X uses a flex pool of nurses to meet the missing demand for the departments. While missing one person can most often be absorbed by the flex pool, missing more nurses on a day becomes increasingly problematic. Therefore the missing capacity objective part is made quadratic. The second part of the objective function describes the difference between the predicted schedule and the current schedule. The sum of the differences measures the distance from the current solution to the predicted schedule.

The constraints are the same as used in Section 4.4.5 to make the schedule feasible. These constraints are based on the most important constraints that can be checked for the kind of prediction done in this research. The objective function is presented in Equation 4.8.

$$\min \sum_{j \in J} (max(0, (d_j - \sum_{i \in I} X_{i,j}))^2 + \sum_{i \in I} \sum_{j \in J} (|p_{i,j} - X_{i,j}|). \tag{4.8}$$

s.t.

$(c_1) : Weekends\ off\ per\ 6\ weeks \geq 2$

$(c_2) : Maximum\ shifts\ in\ a\ week\ 0.5\ fte \leq 4$

$(c_3) : Maximum\ shifts\ in\ a\ row \leq 6$

$(c_4) : Violation\ of\ required\ leave\ wishes = 0$

$(c_5) : Violation\ max\ hours\ per\ four\ weeks = 0$

$x_{i,j} \in \mathbb{N}^+$

$d_j \subset \mathbb{N}^+$

$p_{i,j} \in (0,1) \ \subset \mathbb{R}^+$

$$X_{i.j} = \text{decision variable}$$
$$d_j = \text{Duty demand on day j}$$
$$p_{i,j} = \text{predicted schedule of person i worked on day j (1= work, 0 = no work)}$$

### 4.5.3 Making the predicted schedule feasible

The predicted schedule by the ML model may not be a feasible starting solution if it violates any hard constraints as formulated in Section 4.3.4. The SA approach, as described in Section 4.5.1, does not accept a solution that violates any of the hard constraints since these violations may not exist in the final schedule. Therefore an algorithm has been designed that repairs the schedule in a minimal interfering way with the predicted schedule by deleting shifts. Appendix B shows the outline of the algorithm to make the schedule feasible. Multiple different feasible schedules can be made by slightly changing the algorithm in Appendix B.

### 4.5.4 Operators

This section describes the operators used to find neighbouring solutions to the current schedule. These operators are based on commonly used operators in literature, such as Knust and Xie (2019), Liu et al. (2018), and Turhan and Bilgen (2020). The probability that an operator is selected is equal for all operators for the whole improvement heuristic duration. While some operators change the schedule slightly, such as the insertion operator, the block exchange can change the schedule more significantly.

- Insertion: A nurse is working on a day this person was not working before.
- Deletion: A nurse is not working on a day on which this person was working before.
- Shift switch: A nurse is not working on a day on which this person was working before and is working on a day on which this person was not working before.
- Two-exchange: Shifts between two nurses are swapped.
- Multi-exchange: Shifts on 3 to 6 random days are swapped between two nurses.
- Block-exchange: Shifts on 3 to 6 consecutive days are swapped between two nurses.

### 4.5.5 Improvement heuristic experimentation

This last section describes the experimentation of the metaheuristic methods. Three methods versions of the SA were proposed:

- SA: The classical SA framework in which infeasible neighbour solutions are not accepted.
- Relaxed optimisation: similar to the SA approach, but now infeasible solutions are accepted sometimes. During the relaxed optimization process, we allow for a hundred evenly spaced intervals where, for a specified number of iterations (e.g., 10), infeasible solutions are allowed. After these iterations, the algorithm repairs the schedule to make it feasible using the same method as the starting algorithm. This process helps to diversify the search space, potentially leading to better solutions.

- Gradient Descent: We do not accept worse solutions with a probability anymore. Therefore the solution value can only increase over time but with a higher risk of staying in a local optimum.

Next to these three proposed methods, experimentation with hyperparameters is conducted. The hyperparameters in the improvement heuristics next to the method are the starting temperature, stop temperature and alpha value. The results for various settings for these hyperparameters are presented for the best-performing ML models for each selected department. The trade-off between the solution quality and required computational power is also made in this section.

## 4.6 Conclusion

This chapter presented the solution approach for this research. First, the scheduling approach was explained in more detail, and the simplifications were described. Then the criteria for the model selection were given in Section 4.2. Five Machine Learning models were selected based on the problem requirements of this research: KNN, LR, ANN, RF and GB. Section 4.4 presented that the main evaluation methods are the confusion matrix, ROC curve, missing capacity and hard constraint violations. Finally, Section 4.5 showed the framework used for the improvement heuristic and the experimentation to find better feasible solutions. Three variations for the improvement heuristic were selected: SA, Gradient Descent and relaxed problem.

# Chapter 5

# Results

This chapter presents the research results. First, the ML results are discussed in Section 5.1. Then, Section 5.2 shows the results of the improvement heuristic for the best-performing models. Section 5.3 compares the ML results before and after the improvement heuristic for all departments. Lastly, the implementation of the results is discussed in Section 5.4.

## 5.1 Results ML

This section presents the results of the ML models. Based on the performance of the 90-day planning horizon starting from 01-09-2021, 01-12-2021, 01-03-2022, and 01-06-2022. A simple heuristic is implemented to compare the results of the ML models to a naive method. This naive method copies the schedule from one year back with the same starting date as the planning horizon. For example, let us assume that we want to forecast a 90-day period starting on 01-09-2021, which was a Wednesday. Using the naive method, we start by copying the schedule from the first day of the previous year, which was also a Wednesday. However, since 01-09-2020 was a Tuesday, we instead copy the schedule starting from the next day, which was 02-09-2020. In this way, we create a schedule, but may, just as in the ML models, result in missing capacity of nurses and violations of hard constraints. Therefore, we also compare evaluate the results of the naive method after the improvement heuristic.

The structure of this chapter is as follows. The f1-scores of the ML models are presented in Section 5.1.1. Section 5.1.2 presents the SHAP value analysis of the best-performing model to get more insight into how the ML models make predictions. Based on the performance of the f1-scores, the best models are selected for further evaluation based on the confusion matrix, ROC curve, and the wish scores. These are discussed in Section 5.1.3, Section 5.1.4, and Section 5.1.5, respectively.

### 5.1.1 F1-score

This section presents the f1-score results for the three selected departments. There exists several metrics for the f1-score calculation but we use the weighted f1-score because of the class imbalance in the data. The departments are shown in the following order: IC, radiology, and haematology. Two illustrations are shown for every department. The first figure plots the min, max, and average scores of the six different models and the two different prediction approaches based on the four sequential time horizons of ninety days. Every model has a separate colour to highlight the differences in the model performance. The vertical line indicates the range between the minimum and maximum values. The

dot indicates the average performance for the four time periods. The black horizontal line in this figure corresponds to the average performance of the naive method for the four time periods and is used as a baseline metric to compare the model performances. The second illustration shows the f1-score per time period for the best-performing models in both approaches and also for the naive method.

**IC department** Figure 5.1 shows the f1-score results for the IC department. The results show that every ML model, on average, performs better than the naive method. The predicted rosters, therefore, resemble the actual realised schedule better than the naive method approach. The results for the ML models are better in the PI approach and also show less performance fluctuation of the f1-score for the four selected time periods.

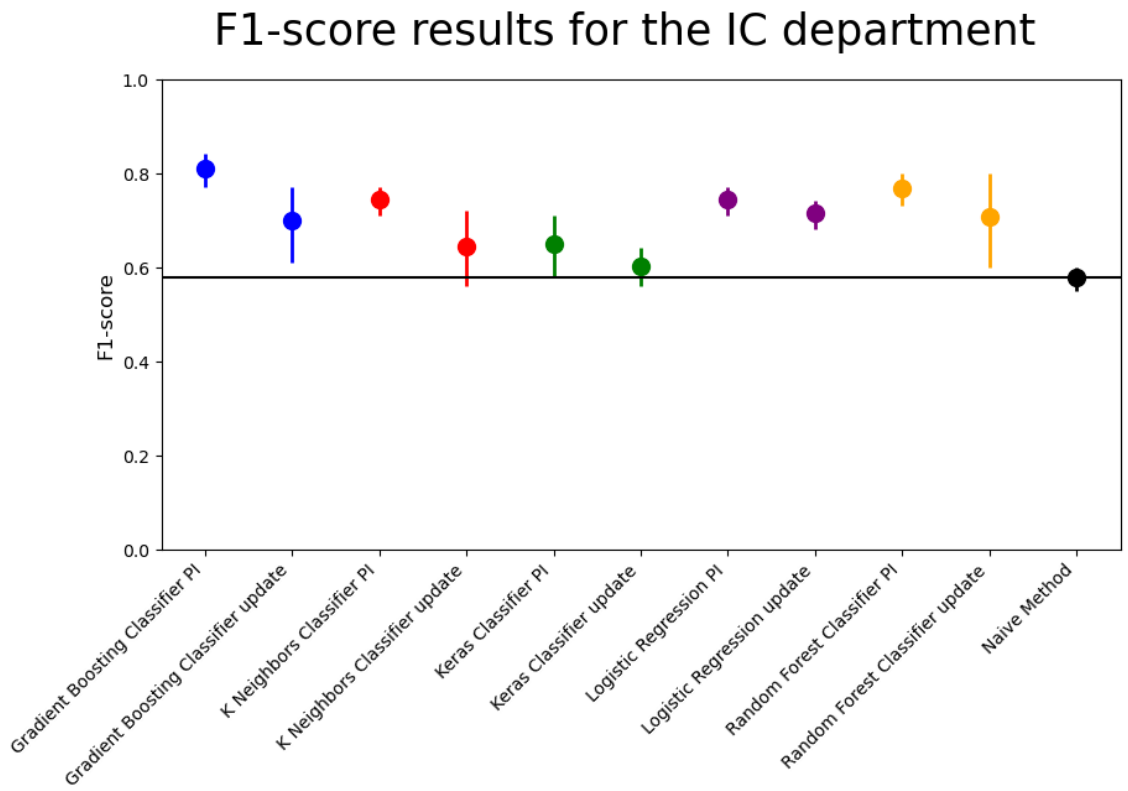



FIGURE 5.1: Min, max and average f1-score results per model and prediction type for the IC department

Table 5.1 examines the performance more closely of the best-performing ML model for the IC department. The table shows the average performance over time per start date of the prediction horizon for the updating and PI approach and also compares the performance to the performance of the naive method per time period.

| F1-score comparison | 1-9-2021 | 1-12-2021 | 1-3-2022 | 1-6-2022 | Average |
|---|---|---|---|---|---|
| Naive method | 0.58 | 0.55 | 0.58 | 0.6 | 0.58 |
| | | | | | |
| Updating approach | 0.68 | 0.7 | 0.74 | 0.74 | 0.72 |
| Improvement over naive method | 0.10 | 0.15 | 0.16 | 0.14 | 0.14 |
| | | | | | |
| PI approach | 0.77 | 0.79 | 0.84 | 0.84 | 0.81 |
| Improvement over naive method | 0.19 | 0.24 | 0.26 | 0.24 | 0.23 |

TABLE 5.1: Comparison of f1-score for the best-performing ML models for the IC department.

The PI and updating approach result in a better performance in the f1-score for all the time periods. GB had the highest f1-score performance in the PI approach with an average of 0.81, while the LR performed the best in the updating approach with an average of 0.72. This is considerably better than the naive method that had a performance of 0.58 for the four evaluated time periods.

**Radiology department** Figure 5.2 shows the f1-score results for the radiology department. The naive method scores better in the radiology department than in the IC department. This is because the recycled schedule from one year back is more similar to the actual schedule. One way to express the similarity of two binary schedules is to use the Jaccard distance. The Jaccard distance measures the number of overlapping instances compared to the total number of instances in the two sets. The average Jaccard distance is 0.32 for IC and 0.39 for Radiology's naive method schedules.

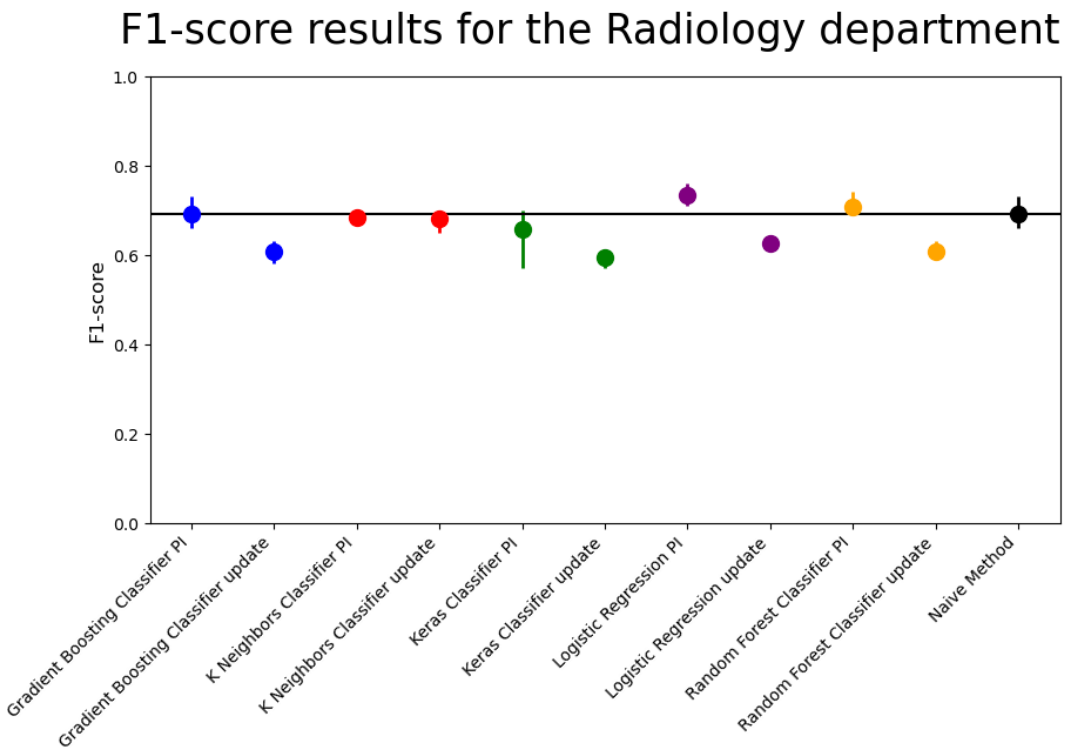



FIGURE 5.2: Min, max and average f1-score results per model and prediction type for the radiology department

Table 5.2 shows the performance of the best models for the updating and PI approach for the radiology department, which are LR and KNN, respectively. The GB model outperforms the naive method by 0.03-0.06 f1-score points in the PI approach. The KNN model performs marginally worse than the naive method, with an average performance of 0.68 over 0.69 for the naive method.

| F1-score comparison | 1-9-2021 | 1-12-2021 | 1-3-2022 | 1-6-2022 | Average |
|---|---|---|---|---|---|
| Naive method | 0.66 | 0.68 | 0.69 | 0.73 | 0.69 |
| | | | | | |
| Updating approach | 0.65 | 0.69 | 0.69 | 0.69 | 0.68 |
| Improvement over naive method | -0.01 | 0.01 | 0.00 | -0.04 | -0.01 |
| | | | | | |
| PI approach | 0.72 | 0.71 | 0.74 | 0.76 | 0.73 |
| Improvement over naive method | 0.06 | 0.03 | 0.05 | 0.03 | 0.04 |

TABLE 5.2: Comparison of f1-score for the best-performing ML models for the radiology department.

**Haematology department** Although that the naive method is comparable in f1-score to Radiology, no ML model performed better on average than the naive method. Figure 3.2 in the data analysis showed that there was a high influx of employees at the start of 2020. Since we did not consider training data before this date, we had fewer data to train on. The combination of significantly fewer wishes than the IC department, the change in department structure in 2020, and the higher performance of the naive method are the main reasons that explain the difference in performance of the ML over the naive method.

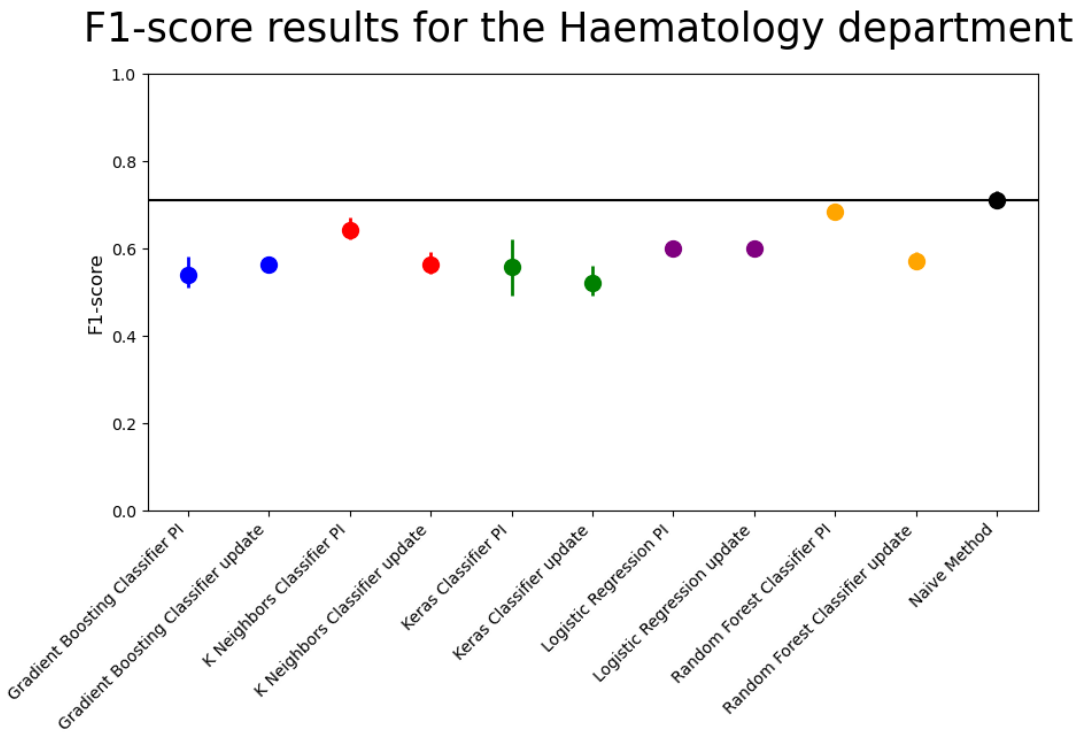



FIGURE 5.3: Min, max and average f1-score results per model and prediction type for the haematology department

Table 5.3 shows the performance of the LR model for the updating approach and the RF for the PI approach. The naive method is 0.11 and 0.03 higher in average f1-score over the best model in the updating approach and PI approach, respectively.

| F1-score comparison | 1-9-2021 | 1-12-2021 | 1-3-2022 | 1-6-2022 | Average |
|---|---|---|---|---|---|
| Naive method | 0.69 | 0.72 | 0.7 | 0.73 | 0.71 |
| | | | | | |
| Updating approach | 0.60 | 0.59 | 0.61 | 0.60 | 0.6 |
| Improvement over naive method | -0.09 | -0.13 | -0.09 | -0.13 | -0.11 |
| | | | | | |
| PI approach | 0.67 | 0.68 | 0.69 | 0.69 | 0.68 |
| Improvement over naive method | -0.02 | -0.04 | -0.01 | -0.04 | -0.03 |

TABLE 5.3: Comparison of f1-score for the best-performing ML models for the haematology department.

### 5.1.2 SHAP value analysis

Shapely Additive exPlanations (SHAP) is a commonly-used method based on game theory to increase the interpretability of ML models.

This section presents the SHAP value results for the LR model in the IC department for the first day in its best test prediction horizon (01-03-2022) to show the feature importance of this model and how the value of the features influences the prediction outcome. The SHAP value analysis for the best-performing models for the other departments and prediction approaches are shown in Appendix D.


FIGURE 5.4: SHAP values for GB for IC department with PI approach

The most important feature is the percentage worked variable, which indicates the number of times that a person worked on a certain day. This is also true for the other departments as shown in Appendix D. The lower the value of the SHAP value for this feature, the lower the probability that a person worked on a day, while the reverse is true for a high SHAP value for this feature. This makes sense since we would also logically expect that when a person does not work, for example, on a Tuesday that the probability that a person wants to work on Tuesday in the future is also small. Other important features are if a person worked yesterday, the duty work wish and how many hours the person has worked already this week. The results for most important features are also similar as seen in Appendix D with the most important features being the percentage worked and lagged variables. The rest of Section 5.1 evaluates the best-performing models based on the average f1-score for each of the selected departments and prediction approaches.

### 5.1.3 Confusion Matrix

Figure 5.5 visualises the confusion matrix performance for each of the best-performing models for the selected departments and the prediction approach. The weighted f1-score of the best models ranges between 0.6 for the Logistic Regression in the Haematology department with updating approach and 0.81 for the Logistic Regression in the IC department. There is no model that performs distinctively better than the other model in all departments, but the Logistic Regression model performed the best in three of the six analysed department prediction approach combinations. This suggests that LR may have certain characteristics that make it suitable for predicting outcomes in these departments, such as assuming a linear relationship between the independent variables. For the confusion matrices for every department and prediction approach, the reader is referred to Appendix C.

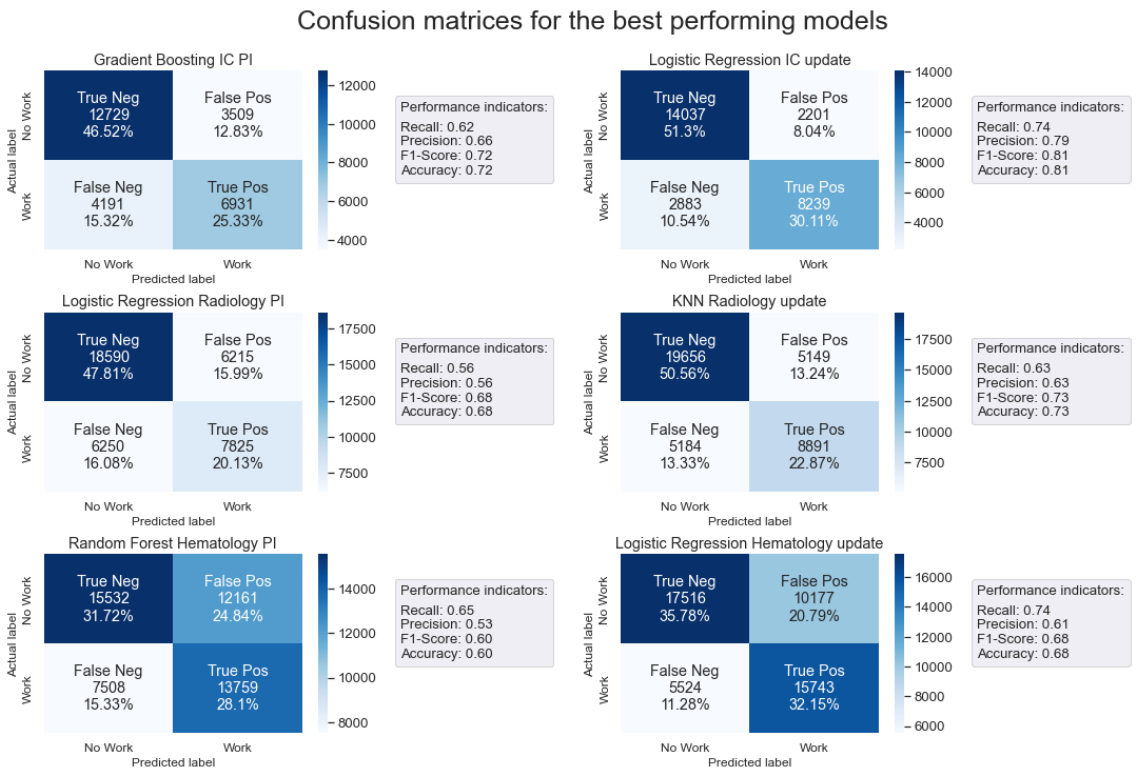



FIGURE 5.5: Confusion matrices for the best performing models

### 5.1.4 ROC curve

Figure 5.6 shows the performance of the ML models for different discrimination thresholds. The AUC value can be interpreted as the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance and thus functions as a measure of separability between predicting that a person worked or not. A method that randomly assigns a person to be working or not has an AUC of 0.5. The figure shows that all models have a higher AUC than a random method. The models in the PI approach score consistently better than the updating approach for all departments as expected.

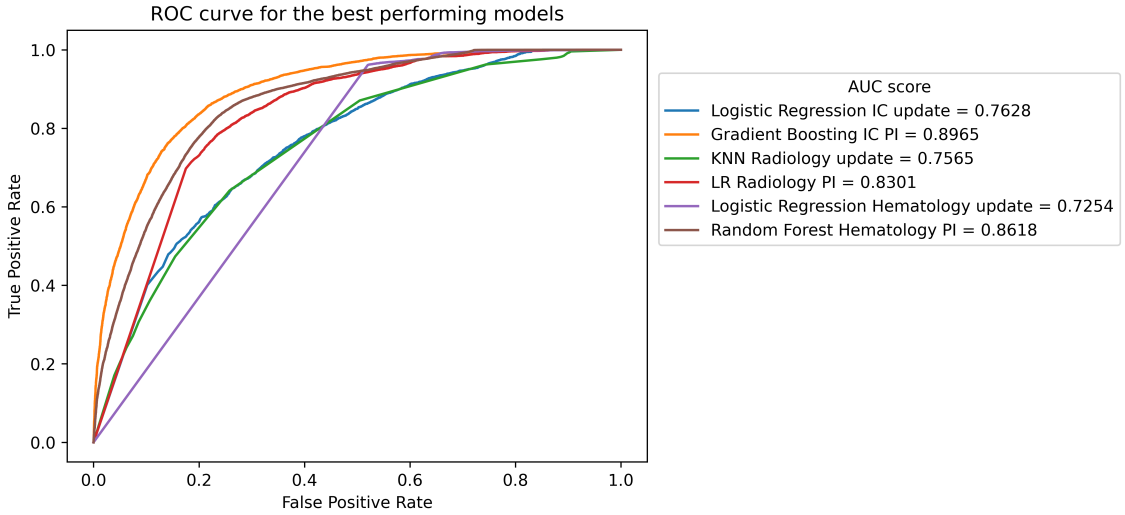
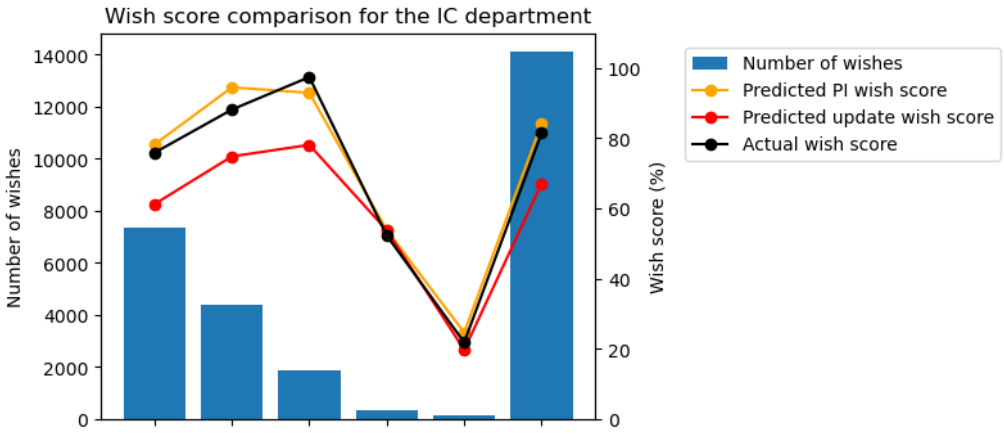



FIGURE 5.6: ROC curves of the best-performing models

### 5.1.5 Wish score

This last sub-section investigates the schedule's performance on the known wishes. One figure is shown per department with the results for the best model in the PI, updating approach and the score of the actual schedule. The graphs show the performance of the wish scores of these three schedules for the different wish types and also include the performance of all the wishes. The wish score is defined as the number of wishes that are adhered to in the schedule compared to the total number of wishes. The blue bars indicate the count of the wish type, and the connected dots indicate the relative performance of the models on the different wish types.

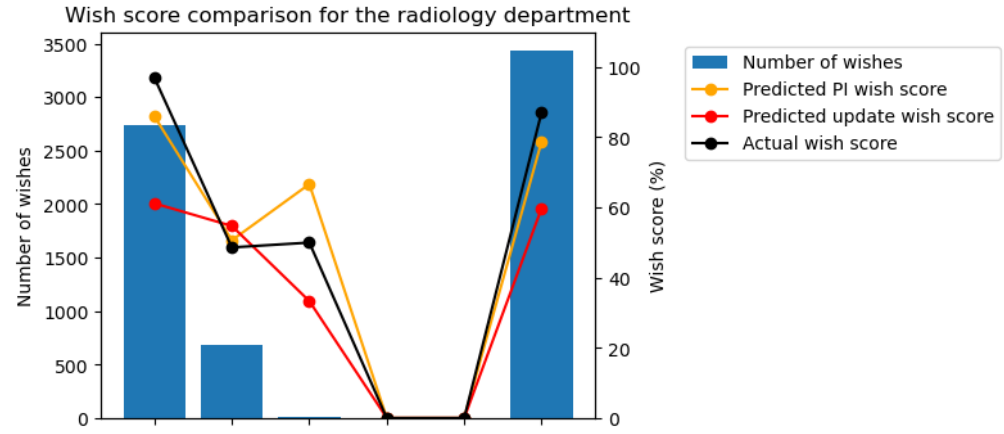
Figure 5.7 shows the wish scores for the IC department. The wish scores of the actual schedules are shown in black. The GB model performs comparable to or better than the actual schedule in terms of wish score for the IC department and only performs worse on the No Work Required wishes (NWR). Remarkably the actual schedule does not score 100% on these wishes, while these wishes are required to meet as formulated in Section 4.8. In practise, however, the wishes are not always complied with when the planning circumstances are so demanding that it is not possible to meet the leave wish. The LR model in the updating approach has a total wish score of 67.14, not far from the total wish score of 81.51 of the actual schedule.




| | DW | NWNR | NWR | DNW | WNR | Total |
|---|---|---|---|---|---|---|
| Nr. wishes | 7359 | 4400 | 1888 | 329 | 142 | 14118 |
| PI wish score (%) | 78.24 | 94.52 | 92.96 | 53.8 | 24.65 | 84.18 |
| Update wish score (%) | 61.23 | 74.84 | 78.12 | 53.8 | 19.72 | 67.14 |
| Actual wish score (%) | 75.88 | 88.18 | 97.46 | 52.28 | 21.83 | 81.51 |

FIGURE 5.7: Wish score comparison for the IC department

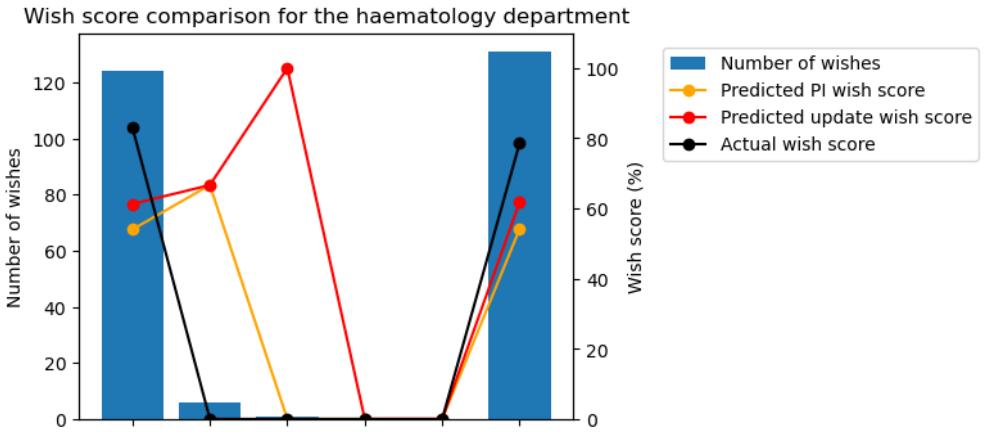
Figure 5.8 shows the performance of the three schedules on the wish types in the radiology department. We see similar results for the wish scores for this department, with comparable results for the best-performing model and the actual schedule and with slightly lower performance for the best model in the updating approach. Most of the wishes for this department are of the NWR type, which is required to comply with.




| | NWR | NWNR | WNR | DW | DNW | Total |
|---|---|---|---|---|---|---|
| Nr. wishes | 2743 | 687 | 6 | 0 | 0 | 3436 |
| PI wish score (%) | 85.96 | 50.66 | 66.67 | 0.0 | 0.0 | 78.87 |
| Update wish score (%) | 61.17 | 54.73 | 33.33 | 0.0 | 0.0 | 59.84 |
| Actual wish score (%) | 96.94 | 48.62 | 50.0 | 0.0 | 0.0 | 87.19 |

FIGURE 5.8: Wish score comparison for the radiology department

Figure 5.9 shows the results of the wish scores for the haematology department. This department has significantly fewer wishes than the IC department and radiology department, with a total of 131 wishes for the four selected time periods. The LR model performs, in this case, slightly better than the RF model but not as well as the actual schedule. Since there were not many wishes to train on, the performance of the models on the wish scores is also slightly lower than in the other two departments.




| | NWNR | DW | NWR | WNR | DNW | Total |
|---|---|---|---|---|---|---|
| Nr. wishes | 124 | 6 | 1 | 0 | 0 | 131 |
| PI wish score (%) | 54.03 | 66.67 | 0.0 | 0.0 | 0.0 | 54.2 |
| Update wish score (%) | 61.29 | 66.67 | 100.0 | 0.0 | 0.0 | 61.83 |
| Actual wish score (%) | 83.06 | 0.0 | 0.0 | 0.0 | 0.0 | 78.63 |

FIGURE 5.9: Wish score comparison for the haematology department

## 5.2 Results metaheuristics

This section presents the metaheuristics results based on the best-performing models for each prediction approach and department. The predicted schedules from these models from the time period 1-12-2021 to 28-12-2022 were selected for the metaheuristics.

As mentioned in Section 4.5.5, three different approaches are evaluated for the improvement heuristic approaches: Simulated Annealing, Gradient Descent and relaxed optimisation. Also, experimentation was done for start temperature and stop temperature and alpha value. The metaheuristics optimal values for the start, stop temperature were found to be 10 and 0.0001, respectively. This section presents the solution value over time for the three metaheuristics approaches for the alpha values: 0.99, 0.995 and 0.999.

The objective value consists of two parts: the similarity of the solution to the predicted schedule and the number of nurses missing according to the capacity required per day. The similarity is expressed in a number where 1 is a perfect identical schedule to the predicted start schedule and 0 is the opposite where every work classification is no work and vice versa compared to the predicted begin schedule. For instance, if half of the nurses work on the same days as the predicted schedule and the other half do not, the similarity score would be 50%.

Because of the size limits of the report and the similarity between the results of the objective value between the departments, we only show the results of the metaheuristics for the IC department. The reader that is interested in the objective value visualisations of the radiology and haematology department is referred to Appendix C. The metaheuristics results are discussed in the following order for the IC department: PI approach, updating approach and lastly, the naive method results are discussed.

### 5.2.1 Metaheuristics results for IC PI

Figures 5.10 show the value of the metaheuristics over time for the predicted schedule of the GB model for the IC PI. The Gradient Descent method has the lowest solution value for the best-found solution in all subplots except for the GB for the IC department with the PI approach when the alpha value is equal to 0.995, albeit with only a small margin.

The advantage of relaxed optimisation over the simulated annealing is clear in Figure 5.10: after some time, the simulated annealing approach is stuck in a local optimum and can not improve further until after a long time, a solution is found that is accepted at around 4 hours. Figure 5.10 shows that better solutions are found over time since the objective value of the solutions decreases over time.

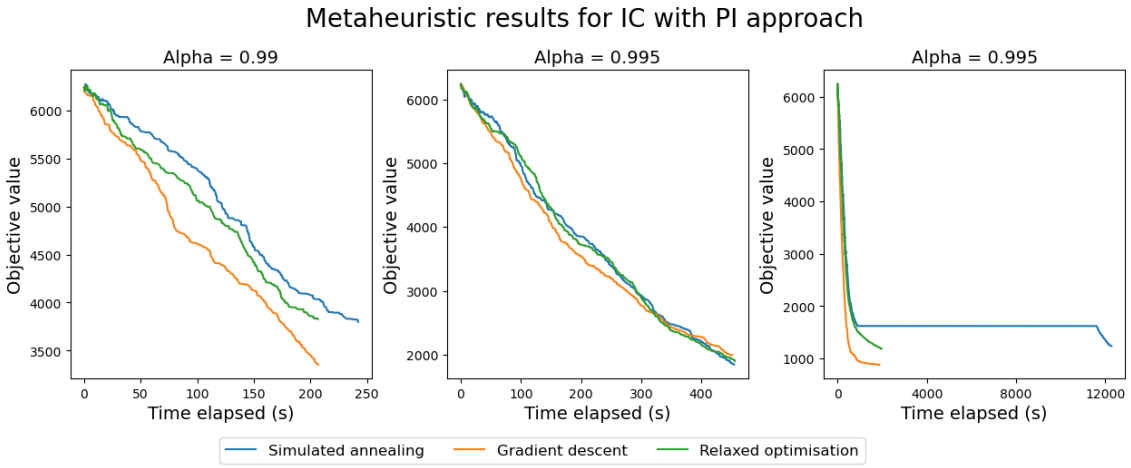



FIGURE 5.10: Objective value IC department with PI approach

Figure 5.11 shows the similarity score and missing capacity for the best-found models in the IC department. The subplot for alpha = 0.999 shows that the SA plot has found a solution that has few missing nurses but that the solution is stuck in a local optimum. Only after a long time the SA method is able to escape the local optimum because it found a solution that has a similar number of missing capacity but is more similar to the start solution. The straight line indicates the missing capacity over time, while the dotted line represents the similarity score.

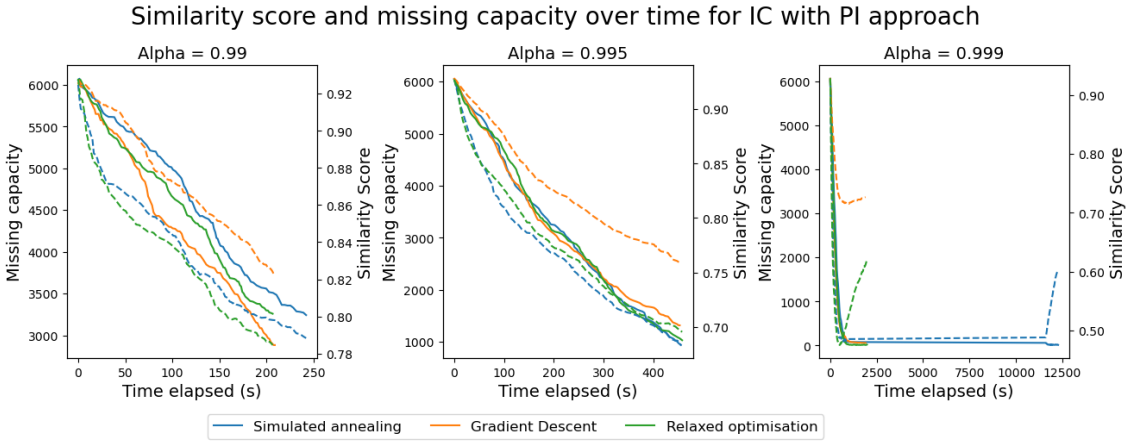



FIGURE 5.11: Similarity score and missing capacity over time for IC department with PI approach

### 5.2.2 Metaheuristics results for IC update

The LR was the best-performing model for the IC department. Figure 5.12 shows the course of the objective function over time. The Gradient Descent outperforms the other

models slightly in terms of objective value except for the smallest alpha value. The graph in Figure 5.13 shows that although the models are comparable in finding a solution with low missing capacity, the greedy Gradient Descent method deviates less from the start solution and thus has a higher similarity score. Since the final solution in the SA and the relaxed optimisation methods have comparable values for the missing capacity but score worse on the similarity score, the Gradient Descent method performs, in general, better than the other two improvement heuristic approaches.

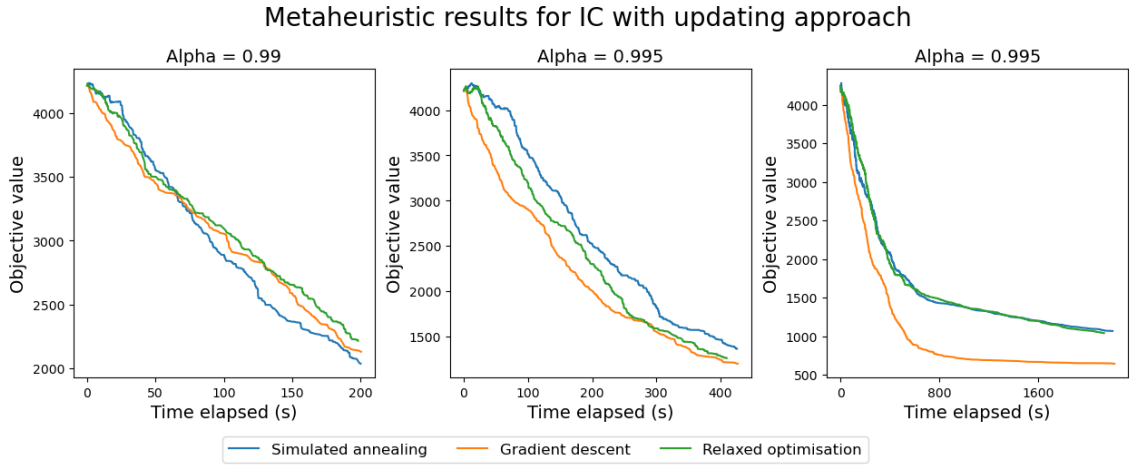



FIGURE 5.12: Objective value IC department with updating approach

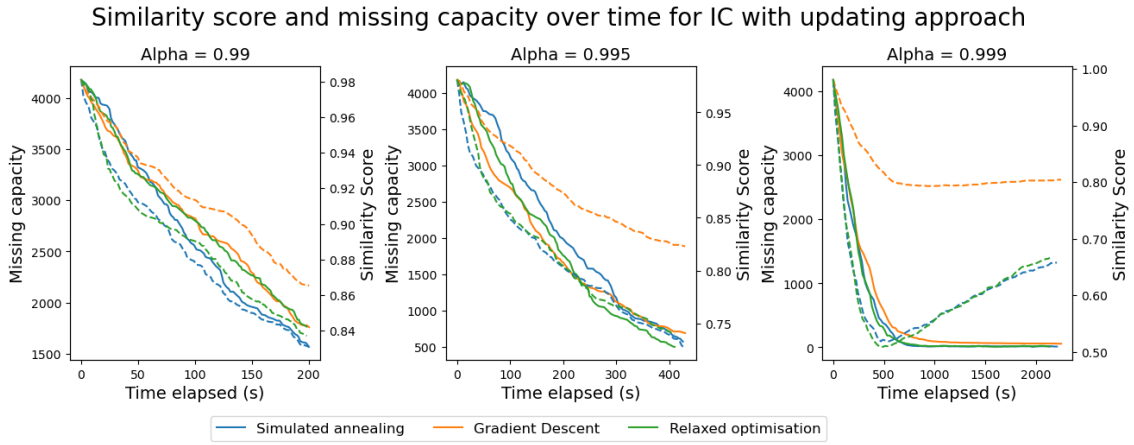



FIGURE 5.13: Similarity score and missing capacity over time for IC department with updating approach

### 5.2.3 Metaheuristics results for IC naive method

Lastly, the improvement heuristic results for the naive method for the IC department are discussed. Also, in this case, the Gradient Descent method finds a final solution that has few missing nurses according to the capacity and is more similar to the predicted start solution than SA and the relaxed optimisation approach. However, the Gradient Descent gets stuck in a local optimum after some iterations and requires a long time to find a better feasible solution in the neighbourhood. To overcome the problem of getting stuck in a solution point, the stop criterion can be changed to stop when a stop temperature

is reached or if no feasible solution is found in a given length of time. Another idea is to combine the Gradient Descent approach with the relaxed optimisation approach such that only better solutions are accepted for the Gradient Descent approach, but infeasible solutions can be accepted sometimes when the Gradient Descent can not find a better feasible solution after a given set of iterations.

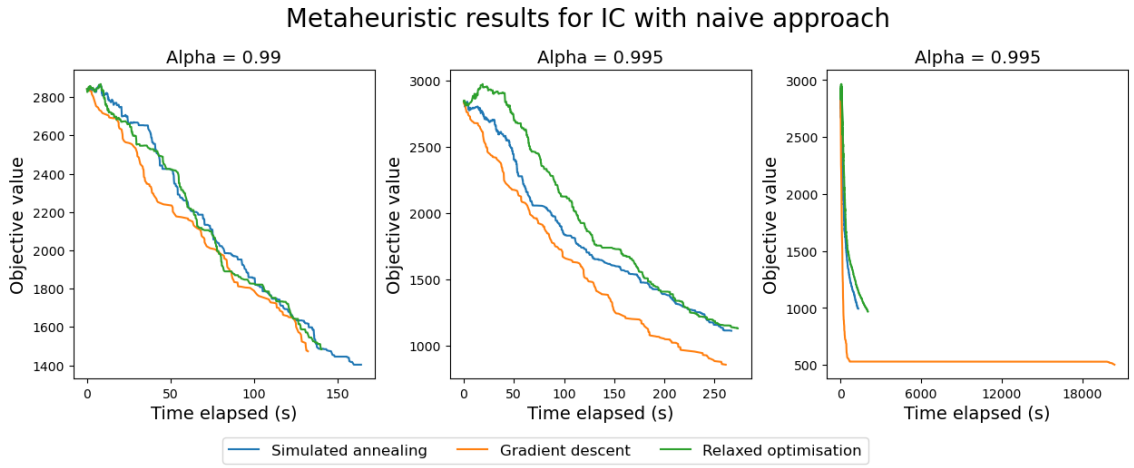



FIGURE 5.14: Objective value IC department with naive method

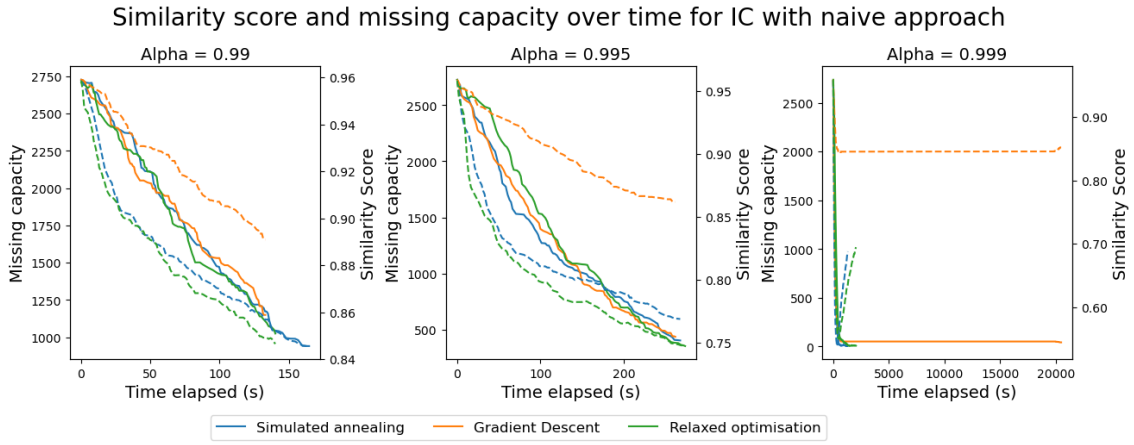



FIGURE 5.15: Similarity score and missing capacity over time for IC department with naive method

## 5.3 Comparison before and after improvement heuristic

This section shows the results of the best-performing models for the PI and updating approach and the naive method before and after the improvement heuristic. The tables show the result after the Gradient Descent with an alpha value of 0.999 since this value for alpha resulted in the best final solutions, and the Gradient Descent method consistently outperformed the SA and relaxed optimisation for the experiments where alpha was equal to 0.999.

The ML result and the ML result after optimisation for the period 01-12-2021 to 28-02-2022 are evaluated based on weighted f1-score, objective value, missing capacity, similarity

score and constraint violation. The best combination of hybrid methods is selected based on the final f1-score and the reduction of objective value after the improvement heuristic.

Section 5.3.1, 5.3.2 and 5.3.3 discusses the results for the IC, radiology and haematology department, respectively.

### 5.3.1 IC

Figure 5.4 shows the results before and after the optimisation by the improvement heuristic. Since the solution in the improvement heuristic can not have any hard constraint violation, the improvement heuristic first step is to remove the shifts that lead to an infeasible solution. The right side of the figure shows the number of constraint violations on each of the five different constraints as formulated in section 4.5.2. The f1-score, objective value, missing capacity and similarity score after the schedule is made feasible are shown on the left side of the figure. The table shows that the improvement heuristic is effective in reducing the missing capacity with a reduction of between 98% and 99% less missing capacity. As an example, the missing capacity for the IC department with the updating approach is equal to 4178. Since we have a prediction time horizon of 90 days this means that the quadratic sum of missing nurses is around 46 per day which equals to an average of around 7 nurses per day. After the optimisation this is equal to less than one missing nurse per day for the updating approach on average. Also, the objective value has decreased by more than 80% with the final solution. The f1-score decreases by 0-5% after the optimisation. Since the updating approach leads to a better f1-score than the other two approaches, while the 85% reduction of the objective value is close to the best reduction of 86% of the PI approach, the updating approach as the best approach for the IC department.

| IC department | F1-score | Objective value | Missing capacity | Similarity | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ |
|---|---|---|---|---|---|---|---|---|---|
| Updating approach | 0.70 | 4233 | 4178 | 0.98 | 5 | 0 | 37 | 13 | 18 |
| After optimisation | 0.69 | 647 | 57 | 0.80 | 0 | 0 | 0 | 0 | 0 |
| Difference | -1% | -85% | -99% | -18% | -100% | 0% | -100% | -100% | -100% |
| PI approach | 0.79 | 6239 | 6056 | 0.93 | 5 | 0 | 37 | 13 | 18 |
| After optimisation | 0.75 | 886 | 60 | 0.73 | 0 | 0 | 0 | 0 | 0 |
| Difference | -5% | -86% | -99% | -22% | -100% | 0% | -100% | -100% | -100% |
| Naive method | 0.55 | 2844 | 2727 | 0.96 | 7 | 0 | 111 | 34 | 87 |
| After optimisation | 0.55 | 506 | 43 | 0.85 | 0 | 0 | 0 | 0 | 0 |
| Difference | 0% | -82% | -98% | -11% | -100% | 0% | -100% | -100% | -100% |

TABLE 5.4: Summary results IC department before and after improvement heuristic

### 5.3.2 Radiology

Figure 5.5 shows the results before and the improvement heuristic for the radiology department. The best-performing solution contains many constraint violations except for the naive method. Therefore, the missing capacity at the start of the local search is relatively high compared to the IC department. However, the improvement heuristic is effective in reducing the missing capacity since the final solution has almost no missing capacity. The improvement heuristic, however, has to sacrifice some similarity of the solution to obtain the final solution with low missing capacity scores since the similarity is reduced by 23-29% depending on the model. Again, we see a slight reduction in the f1-score after optimisation.

| Radiology department | F1-score | Objective value | Missing capacity | Similarity | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ |
|---|---|---|---|---|---|---|---|---|---|
| Updating approach | 0.70 | 16958 | 16561 | 0.87 | 9 | 0 | 298 | 108 | 1050 |
| After optimisation | 0.67 | 1575 | 52 | 0.62 | 0 | 0 | 0 | 0 | 0 |
| Difference | -4% | -91% | -100% | -29% | -100% | 0% | -100% | -100% | -100% |
| | | | | | | | | | |
| PI approach | 0.71 | 15732 | 15475 | 0.92 | 9 | 0 | 498 | 54 | 1107 |
| After optimisation | 0.69 | 1414 | 48 | 0.66 | 0 | 0 | 0 | 0 | 0 |
| Difference | -3% | -91% | -100% | -28% | -100% | 0% | -100% | -100% | -100% |
| | | | | | | | | | |
| Naive method | 0.68 | 9396 | 9235 | 0.97 | 5 | 0 | 0 | 0 | 59 |
| After optimisation | 0.67 | 995 | 67 | 0.75 | 0 | 0 | 0 | 0 | 0 |
| Difference | -1% | -89% | -99% | -23% | -100% | 0% | 0% | 0% | -100% |

TABLE 5.5: Summary results IC department before and after improvement heuristic

### 5.3.3 Haematology

Finally, the results for the haematology department are discussed. This department also contains relatively many violations for the updating and PI approach, especially for the violation of max hours and max shifts in a row. The improvement heuristic reduces the missing capacity again effectively. The f1-score of the final solution decreases or stays the same in the haematology department. But since the final f1-score is the best for the naive method and the objective value is lowest compared to the updating and PI approach, the best approach for the haematology department is to use the naive method.

| Haematology department | F1-score | Objective value | Missing capacity | Similarity | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ |
|---|---|---|---|---|---|---|---|---|---|
| Updating approach | 0.62 | 18325 | 17625 | 0.88 | 9 | 0 | 1343 | 0 | 2673 |
| After optimisation | 0.62 | 1821 | 41 | 0.73 | 0 | 0 | 0 | 0 | 0 |
| Difference | 0% | -90% | -100% | -17% | -100% | 0% | -100% | 0% | -100% |
| | | | | | | | | | |
| PI approach | 0.70 | 26973 | 26032 | 0.83 | 9 | 0 | 460 | 0 | 1606 |
| After optimisation | 0.68 | 2658 | 550 | 0.68 | 0 | 0 | 0 | 0 | 0 |
| Difference | -3% | -90% | -98% | -18% | -100% | 0% | -100% | 0% | -100% |
| | | | | | | | | | |
| Naive method | 0.73 | 6769 | 6708 | 0.99 | 0 | 0 | 0 | 0 | 114 |
| After optimisation | 0.72 | 467 | 35 | 0.93 | 0 | 0 | 0 | 0 | 0 |
| Difference | -1% | -93% | -99% | -6% | 0% | 0% | 0% | 0% | -100% |

## 5.4 Implementation of the results

This last section concludes the result section and suggests how the results may be used for an AI-assisted scheduling decision support tool.

Two approaches for the prediction were tested: the PI approach and updating approach. The results were compared to a naive method. Since the schedule is not allowed to contain any violation of the formulated hard constraint wishes, experimentation with three types of improvement heuristics was conducted. The greedy Gradient Descent method resulted in the best results, mainly because the similarity score was higher over time during the improvement heuristic process. The start temperature, stop temperature, and alpha value were set to 10, 0.0001 and 0.999, respectively. While the PI approach resulted in the best performance for the ML models for the IC and the radiology department, the final f1-score after the improvement heuristic was higher in the updating approach. The naive method performed the best in the haematology department before and also after the improvement heuristic. Performance varies based on schedule characteristics per department, so the best approach may differ in an unexplored department at hospital X.

The proposed hybrid combination of ML and improvement heuristics is able to create schedules that contained no violation of the constraints, was missing few nurses per day

on average and had an f1-score between 0.62 and 0.75 for updating and the PI approach. Figure 5.6 gives an example of how the schedule before and after optimisation for the IC department with PI approach for the second week in the 1-12-2021 to 28-12-2022 prediction horizon. The violation of maximum shifts in a row is indicated with an orange colour, while the leave wish violation is indicated in blue. This snippet schedule did not contain any other violations. The improvement heuristic took care of the violations by changing the assignment of work among the nurses.

| Resource id | 6-12-21 | 7-12-21 | 8-12-21 | 9-12-21 | 10-12-21 | 11-12-21 | 12-12-21 |
|---|---|---|---|---|---|---|---|
| 1666061 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1666373 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1666400 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1666694 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1666724 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1666754 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1666760 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1666938 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1667038 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1667428 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

(A) Before optimisation

| Resource id | 6-12-21 | 7-12-21 | 8-12-21 | 9-12-21 | 10-12-21 | 11-12-21 | 12-12-21 |
|---|---|---|---|---|---|---|---|
| 1666061 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1666373 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1666400 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1666694 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1666724 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1666754 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1666760 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1666938 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1667038 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1667428 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

(B) After optimisation

TABLE 5.6: Predicted schedule IC PI approach for the second week in the 1-12-2021 to 28-12-2022 prediction horizon

There are two ways that this research may aid in the development of an AI-assisted workforce scheduling tool:

1. Help the nurse scheduler with creating the planning for the nurses. The workforce tool can, for example, highlight every day what set of nurses is likely to work. This may speed up the process of making the schedules by hand for the nurse planners.

2. As noted in Section 1.3.4, the genetic algorithm solution currently used in the OWS optimiser does not perform well on the most used benchmark instances in "Nurse Rostering Benchmark Instances" (2023) (Hassan, 2022). While the schedules generated by ML models could potentially serve as a viable starting solution for the GA and enhance its performance on these test cases, this requires further investigation.

# Chapter 6

# Conclusion and discussion

This last chapter provides the conclusion and discussion for the research and gives recommendations for ORTEC B.V. and for further research. The conclusion of the research is given in Section 6.1, while the discussion of the research is presented in Section 6.2. The recommendations for ORTEC and further research are given in Section 6.3.

## 6.1 Conclusion

The research of this thesis assignment was the development of a ML model that creates feasible schedules with the main research question:

> *'How can a combination of machine learning and improvement heuristics create a feasible schedule that adheres to nurse preferences for hospital X?'*

This main research question is answered by answering a number of sub-research questions.

1. *What approaches for the NSP have been applied in literature?* The literature section focused on the NSP and the solutions methodologies that were applied in this area of research. Most of the NSP research has been focused on solving a set of test instances that may not reflect reality well. However, more recent research in the NSP field develop solutions that allow for more flexibility and represent real-life planning better.

   The body of literature is still relatively thin on ML techniques for the NSP, although some authors note that applying ML techniques may prove to be successful. This research hopes to fill the ML gap in NSP research by proposing a combination of ML and metaheuristic techniques that makes feasible schedules based on actual realisations by a hospital in the past.

2. *How is the planning of the nurses organised in hospital X?* The second research question focused on describing how the nurse scheduling was organised at hospital X. Hospital X used a centralised planning method with one head nurse in charge of arranging the schedule for the nurses in the past but switched to a team rostering planning process recently. Nurse planners indicated during the hospital visit that both the planners and the nurses themselves believe that changing from a centralised to team planning approach has deteriorated the quality of the rosters. This underlines how complex the roster-making process is.

Section 3.1 investigates the data to understand the data entities better. Five essential constraints were selected for this research to check the feasibility of the schedule. These constraints are used at hospital X and are also used in some form at other hospitals. The outcome of this research question formed the basis for the design of the ML and metaheuristic design.

3. *How can the design of machine learning models be optimised for schedule prediction?* Combining the information from the hospital visit, the findings from Euser (2022), the data analysis and trial and error, we selected the features in Section 3.2. Five ML models were selected based on the criteria as defined in Section 4.2.2: KNN, LR, ANN, RF and GB.

   We selected a four-fold cross-validation method for a robust evaluation of the ML results. The test sets in these folds were each ninety days long, similar to the hospital planning horizon length. We selected three departments for this research: IC, radiology and the haematology department because of the size of the number of personnel and shifts are done that allowed for good training of the ML models.

   The performance of the models was evaluated in two different ways: PI and updating approach. The PI approach evaluated the model's performance by using the actual realisations of the feature values over time, while the updating approach updates the feature values based on the earlier predictions made. A naive method that copies the schedule from one year back was also implemented to benchmark the ML results with a simple naive method.

4. *How can metaheuristic methods be used to make the solution feasible?* Section 4.5 proposed an improvement heuristic design based on SA specifically geared to the problem at hand. Six different operators are used to create neighbour solutions based on commonly used operators for SA from the NSP in the literature. Two alterations of the SA framework are tested: Gradient Descent and relaxed problem approach.

   The objective of the local search is to obtain feasible solutions for which a minimum number of persons are working according to the duties that need to be performed. Also, we can not stray too far away from the original predicted schedule since then the final solution may not reflect the learned aspects of the ML model prediction well.

5. *How can the proposed solution aid in hospital planners' schedule-making?* The last research question investigated how the proposed solution methodology can help in the manual planning process. Chapter 5 showed the results of both the ML models and the metaheuristic. The performance of the ML models was evaluated on the weighted f1-score. For every department and prediction approach combination, the best models were selected and evaluated further on the confusion matrix, ROC curve and wish score to understand the results better.

   The ML results showed a significant improvement over the naive method for the IC department, limited performance increase for the radiology department and no model attained the performance of the naive method for the haematology department. The reasons for the lower results in the radiology and haematology department compared to the IC department are: more cyclist schedule in the haematology and radiology department and therefore higher naive method results, fewer wishes to train on and a change of planning circumstances at the start of 2020 for the haematology department.

Three different metaheuristic approaches were evaluated. The Gradient Descent method showed the best performance for almost all the departments and alpha values. This has to do with the fact that Gradient Descent does not accept worse solutions and thus remains to have a high similarity score over time.

Sections 5.3 and 5.4 compared the results before and after the improvement heuristic and discussed the implementation of the results for an AI-assisted workforce scheduler tool. The final improvement heuristic results lead to feasible solutions with few missing nurses on average over the days. The best hybrid model for the IC and radiology department were found to be the updating approach with Gradient Descent improvement heuristic approach, while the naive method with Gradient Descent method was the best for the haematology department. More research is required for a successful, robust implementation of the solution methodology as an AI-assisted support tool; Section 6.3 describes this matter.

## 6.2 Discussion

This section discusses the results. The approach for the research was based on the planning process at hospital X and hospitals in general. However, some assumptions had to be made to enable the solution approach to work in the available time frame for this research. The impact of the assumptions and the solution approach is discussed in this section.

Based on the available dataset, the data analysis and the earlier research done by Euser (2022), features for this research were selected and extracted to train the ML models. However, we know that more features could be added that most likely improve the performance of the ML models. In this research, we did not have information on the skills that the persons could perform. Also, more personal information, such as the age of the person, was not known, which could impact how the person wants to be scheduled over time.

Adding more employee-specific characteristics, such as skills, would not only improve the performance of the models but also increase the understanding of the quality of the roster. The best-performing ML model differed per department and prediction approach, with the LR model performing the best in half of the tested instances. LR is a powerful ML model when the relationship between the independent variables is linear, and the classification problem has a binary output. Adding more and more powerful features may change what model performs the best.

In this research, the assumption is made that the actual schedule is the "perfect" schedule and that the more the predicted schedule resembles the actual schedule, the better. There are, however, two issues with this approach: the actual schedule may not be the perfect schedule. Section 5.1.5 showed that also the actual schedule was not able to adhere to all of the wishes that were known in advance. Secondly, and maybe more important than we first point, there can be many different schedules that are, in fact, equally good as the actual realised schedule. As an example, the assignment of two nurses to a day with the same preferences to work on a day may be interchanged while still preserving the same quality of the roster. Let us assume we only want to schedule three nurses on a day called A, B and C. Table 6.1 shows what the schedule for an arbitrary day for these three nurses may look like.

| | Nurse A | Nurse B | Nurse C |
|---|---|---|---|
| Actual schedule | 1 | 0 | 1 |
| Predicted schedule | 0 | 1 | 1 |

TABLE 6.1: Simplified scheduling example.

In this research, only the prediction for Nurse C would be considered correct, and the prediction for Nurse A and B would not be correct. Still, it may be the case that this schedule would also be a good schedule since Nurse B can perform the same tasks as Nurse A and has the same wish preferences for this day. Nevertheless, since the information about the interchangeability of the nurses was not available, the schedules were only evaluated on the known information, such as the actual schedule and the wishes. To evaluate the performance of the predicted schedule based on "what could have been" a schedule instead of "what was the schedule", a nurse planner is required that evaluates the performance of the schedule. This is, however, a laborious process and also requires some after-processing steps such that the nurse planners understand who is planned on what day.

A drawback of this research is that can not take dependencies between nurses into account. It is possible to transform a problem to a multi-label problem to take dependencies into account, but initial results showed that this deteriorated the performance of the ML models, most likely because the transformation to a multi-label problem results in higher dimensional data while the number of observations is reduced. If more data is available, then this approach may work better than a single-label classification approach that can not take dependencies into account.

We tested two prediction approaches for the performance testing of the ML models: the PI and the updating approach. The PI approach uses the actual realised feature values till the prediction day, while the updating approach updates the feature values based on the predictions made by the ML before. The updating approach thus makes a prediction on a prediction and is considerably more difficult to get the correct classification for work for nurses.

The improvement heuristic improved the objective value of the predicted solutions and also ensured the feasibility of the schedules. The objective function consisted of two parts: similarity to the predicted schedule and the missing capacity. But since the missing capacity part was quadratic and the similarity to the predicted schedule was not in the objective function, the improvement heuristic is mainly effective in finding solutions with a low number of missing capacities, but finding a solution that is very close to the predicted ML schedule proved to be difficult. For this reason, the greedy descent method performed often times better on the objective value for the final solution. The final schedules after the optimisation resulted in slightly lower f1-score than the original predicted schedule.

One idea for improvement of the efficiency of the improvement heuristic is, for example, to create a neighbour solution in a smarter way. We know, for example, for sure that a person can not work more than six days in a row. If we have a current solution in which a person has already worked six days in a row, we cannot use the "shift-on" operator on the day preceding and succeeding this six-day work streak.

This research is part of the AI-assisted support tool project of ORTEC that is initiated to find improvements in the understanding of the quality of rosters and find ways to incorporate AI techniques in the OWS optimiser. This research provided a hybrid method that learns from previous schedules and can create a feasible schedule based on actual realisations. Some assumptions and simplifications were made, mainly involving the set of nurses that were selected for this research. Only nurses with an active contract and with a

minimum number of information available in the database were predicted for this research. Although there were only a few nurses starting in the prediction horizons, as shown in the data analysis chapter, these nurses were not accounted for in the planning process. Finally, student nurses were not included in this research since these are always planned above the capacity and are much more difficult to predict because they stay in a department for a short time. While some of these assumptions resemble the planning situation at the hospital, such as not accounting for the student nurses in the main schedule, the exclusion of some nurses can lead to a small deviation from the planning process in real life.

Lastly, more extensive testing could have improved the results. A coarse grid search was performed on the hyperparameters of the ML models, but a finer grid search could have improved the results. This was not done because of the time limitations of this project. Furthermore, five ML models were tested for this research, and it would be interesting to see what performance can be attained by different models. A suggestion for a new model to be tested is a variant of ANN that can take time dependencies into account and can forget older information: Long Short-Term Memory. Hospital environments are dynamic, and the way that nurses want to be planned can also change significantly over time. A model such as Long Short-Term Memory may capture these changes and predict a schedule based on the most relevant planning situation.

## 6.3 Recommendations

As mentioned in the discussion, there are multiple ways to improve the results of the proposed solution methodology. The main improvement points are adding more relevant features, making a more balanced trade-off between the objectives for the improvement heuristics and testing the performance in a more fair way. As part of the AI-assisted project, ORTEC is working on better understanding what distinguishes a bad roster from a good roster with, for example, the research by Rooijen (2023). Rooijen (2023) could help understand what features may improve the solution methodology results of this research. As mentioned in the discussion, we chose to make the missing capacity part quadratic and the similarity score linear in the objective function. The metaheuristic thus gives more weight to reduce the missing capacity than the similarity score, especially for large values. A more careful trade-off between missing capacity and similarity score may help to balance these two aspects better in the improvement heuristic. The last main recommendation is to test the predicted schedules in a more fair way by taking, for example, the interchangeability of the nurses into account. Current research by Rooijen (2023) is investigating how nurses perceive the quality of the roster, this research may help in understanding how to test the predicted schedule is a better way.

Another suggestion for further research is to evaluate how the solution methodology can be effectively incorporated into the planning process of planners. The provided solution framework of this research provides a starting point for creating a feasible schedule for the nurses based on past realisations. More steps are required for an effective implementation of the solution methodology such as adding more hard constraints in the metaheuristic.

This research focused on the assignment of nurses to days in the prediction horizon. The actually realised schedule also contains information about what duty the nurses performed and what time this duty started and stopped. A plan that also predicts the duty that is performed and the start and stop time may create a new understanding of the roster process and increase the ability of the solution methodology to be implemented in real-life. New research as part of the AI-assisted workforce project starts directly after this research that builds upon this research and also takes shift assignment into account.

# Bibliography


Abdennadher, S., & Schlenker, H. (1999). An interactive constraint based nurse scheduler. *In Proceedings of The First International Conference and Exhibition on The Practical Application of Constraint Technologies and Logic Programming, PACLP*.

Aickelin, U., & Dowsland, K. A. (2004). An indirect genetic algorithm for a nurse-scheduling problem. *Computers & operations research*, *31*(5), 761–778.

Aickelin, U., & Li, J. (2007). An estimation of distribution algorithm for nurse scheduling. *Annals of Operations Research*, *155*(1), 289–309.

Aiken, L. H., Clarke, S. P., Sloane, D. M., Sochalski, J., & Silber, J. H. (2002). Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. *Jama*, *288*(16), 1987–1993.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40–79.

Asgeirsson, E. I. (2014). Bridging the gap between self schedules and feasible schedules in staff scheduling. *Annals of Operations Research*, *218*, 51–69.

Bailyn, L., Collins, R., & Song, Y. (2007). Self-scheduling for hospital nurses: An attempt and its difficulties. *Journal of nursing management*, *15*(1), 72–77.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83. https://doi.org/https://doi.org/10.1016/j.csda.2017.11.003

Bester, M., Nieuwoudt, I., & Van Vuuren, J. H. (2007). Finding good nurse duty schedules: A case study. *Journal of scheduling*, *10*(6), 387–405.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239–287.

Blythe, J., Baumann, A., Zeytinoglu, I., Denton, M., & Higgins, A. (2005). Full-time or part-time work in nursing: Preferences, tradeoffs and choices. *Healthcare Quarterly (Toronto, Ont.)*, *8*(3), 69–77, 4.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, *7*(6), 441–499. https://doi.org/10.1023/B:JOSH.0000046076.75950.0b

Burke, E. K., Li, J., & Qu, R. (2010). A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research*, *203*(2), 484–493.

Castillo, I., Joro, T., & Li, Y. Y. (2009). Workforce scheduling with multiple objectives. *European Journal of Operational Research*, *196*(1), 162–170.

Cheang, B., Li, H., Lim, A., & Rodrigues, B. (2003). Nurse rostering problems a bibliographic survey. *European journal of operational research*, *151*(3), 447–460.

Chen, Z., De Causmaecker, P., & Dou, Y. (2020). Neural networked assisted tree search for the personnel rostering problem. *arXiv preprint arXiv:2010.14252*.

Chen, Z., De Causmaecker, P., & Dou, Y. (2022). A combined mixed integer programming and deep neural network–assisted heuristics algorithm for the nurse rostering problem. *Applied Soft Computing*, 109919.

Cissen, N. P. (2022). *Predicting rosters in healthcare by using machine learning* (Thesis).

Dantzig, G. B. (1954). A comment on edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, *2*(3), 339–341.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, *7*, 1–30.

Detienne, B., Péridy, L., Pinson, É., & Rivreau, D. (2009). Cut generation for an employee timetabling problem. *European Journal of Operational Research*, *197*(3), 1178–1184.

Dietterich, T. G. (2000). Ensemble methods in machine learning, 1–15.

Dong, G., & Liu, H. (2018). *Feature engineering for machine learning and data analytics*. CRC Press.

Drennan, V. M., & Ross, F. (2019). Global nurse shortages: The facts, the impact and action for change. *British medical bulletin*, *130*(1), 25–37.

Edie, L. C. (1954). Traffic delays at toll booths. *Journal of the operations research society of America*, *2*(2), 107–138.

Entezari-Maleki, R., Rezaei, A., & Minaei-Bidgoli, B. (2009). Comparison of classification methods based on the type of attributes and sample size. *J. Convergence Inf. Technol.*, *4*(3), 94–102.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, *153*(1), 3–27.

Euser, S. (2022). *Fairness on quality of nurse rosters* (Thesis).

Fernández-Delgado, M., Sirsat, M. S., Cernadas, E., Alawadi, S., Barro, S., & Febrero-Bande, M. (2019). An extensive experimental survey of regression methods. *Neural Networks*, *111*, 11–34.

Fogarty, C., & Cronin, P. (2008). Waiting for healthcare: A concept analysis. *Journal of advanced nursing*, *61*(4), 463–471.

Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Metaheuristic algorithms in modeling and optimization. *Metaheuristic applications in structures and infrastructures*, *1*.

Garde, A. H., Albertsen, K., Nabe-Nielsen, K., Carneiro, I. G., Skotte, J., Hansen, S. M., Lund, H., Hvid, H., & Hansen, Å. M. (2012). Implementation of self-rostering (the prio project): Effects on working hours, recovery, and health. *Scandinavian Journal of Work, Environment & Health*, *38*(4), 314–326. Retrieved February 16, 2023, from http://www.jstor.org/stable/41508898

Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). *An introduction to statistical learning: With applications in r*. Spinger.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media. https://books.google.nl/books?id=HHetDwAAQBAJ

Gomes, C. P. (2000). Artificial intelligence and operations research: Challenges and opportunities in planning and scheduling. *The Knowledge Engineering Review*, *15*(1), 1–10.

Gradient boosting machines. (2023). http://uc-r.github.io/gbm_regression

Griffiths, J. D., Price-Lloyd, N., Smithies, M., & Williams, J. E. (2005). Modelling the requirement for supplementary nurses in an intensive care unit. *Journal of the Operational Research Society*, *56*(2), 126–133.

Günther, M., & Nissen, V. (2010). Particle swarm optimization and an agent-based algorithm for a problem of staff scheduling. *Applications of Evolutionary Computation: EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part II*, 451–461.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 986–996.

Haddad, L. M., Annamaraju, P., & Toney-Butler, T. J. (2022). Nursing shortage. In *Statpearls [internet]*. StatPearls Publishing.

Hall, R. W. (2012). *Handbook of healthcare system scheduling*. Springer.

Hassan, S. (2022). *Performance OWS optimizer on standard NSP benchmarks* (Unpublished Work).

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Henderi, H., Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer. *International Journal of Informatics and Information Systems*, *4*(1), 13–20.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832–844.

Hulshof, P. J., Kortbeek, N., Boucherie, R. J., Hans, E. W., & Bakker, P. J. (2012). Taxonomic classification of planning decisions in health care: A structured review of the state of the art in or/ms. *Health systems*, *1*(2), 129–175.

Hung, R. (2002). A note on nurse-self-scheduling. *Nursing Economics*, *20*(1), 37.

Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International journal of engineering research and applications*, *3*(5), 605–610.

Jenkins, J. L., O'Connor, R. E., & Cone, D. C. (2006). Differentiating large-scale surge versus daily surge. *Academic Emergency Medicine*, *13*(11), 1169–1172.

Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., & Talbi, E.-G. (2022). Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research*, *296*(2), 393–422.

Kearns, M. (1988). *Thoughts on hypothesis boosting* [Unpublished].

Kellogg, D. L., & Walczak, S. (2007). Nurse scheduling: From academia to implementation or not? *Interfaces*, *37*(4), 355–369.

Khalil, E., Le Bodic, P., Song, L., Nemhauser, G., & Dilkina, B. (2016). Learning to branch in mixed integer programming. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Kingma, M. (2018). *Nurses on the move: Migration and the global health care economy*. Cornell University Press.

Kinha, Y. (2022). An easy guide to choose the right machine learning algorithm. https://www.kdnuggets.com/2020/05/guide-choose-right-machine-learning-algorithm.html

Knust, F., & Xie, L. (2019). Simulated annealing approach to nurse rostering benchmark and real-world instances. *Annals of Operations Research*, *272*(1-2), 187–216.

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

Kumar, M., Teso, S., De Causmaecker, P., & De Raedt, L. (2019). Automating personnel rostering by learning constraints using tensors. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 697–704.

Kumari, R., & Srivastava, S. K. (2017). Machine learning: A review on binary classification. *International Journal of Computer Applications*, *160*(7).

Kurt, I., Ture, M., & Kurum, A. T. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. *Expert Systems with Applications*, *34*(1), 366–374. https://doi.org/https://doi.org/10.1016/j.eswa.2006.09.004

Laurens, F. v. D., Gerard, P., Bart, V., & Wessel, W. (2006). *Harmonious personnel scheduling* (Unpublished Work).

Legrain, A., Bouarab, H., & Lahrichi, N. (2014). The nurse scheduling problem in real-life. *Journal of Medical Systems*, *39*(1), 160. https://doi.org/10.1007/s10916-014-0160-8

Li, J., & Aickelin, U. (2003). A bayesian optimization algorithm for the nurse scheduling problem. *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, *3*, 2149–2156.

Liu, Z., Liu, Z., Zhu, Z., Shen, Y., & Dong, J. (2018). Simulated annealing for a multi-level nurse rostering problem in hemodialysis service. *Applied Soft Computing*, *64*, 148–160.

Lü, Z., & Hao, J.-K. (2012). Adaptive neighborhood search for nurse rostering. *European Journal of Operational Research*, *218*(3), 865–876.

Mandrekar, J. N. (2010). Receiver Operating Characteristic Curve in Diagnostic Test Assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316. https://doi.org/https://doi.org/10.1097/JTO.0b013e3181ec173d

Morgado, E. M., & Martins, J. P. (1993). An AI-based approach to crew scheduling. *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*, 71–77.

Murray, M. K. (2002). The nursing shortage: Past, present, and future. *JONA: The Journal of Nursing Administration*, *32*(2), 79–84.

Nurse rostering benchmark instances. (2023). http://www.schedulingbenchmarks.org/nrp/

Ortec. (2022). https://ortec.com/en

Pandey, A., & Jain, A. (2017). Comparative analysis of KNN algorithm using various normalization techniques. *International Journal of Computer Network and Information Security*, *9*(11), 36.

Petrovic, S., & Vanden Berghe, G. (2012). A comparison of two approaches to nurse rostering problems. *Annals of Operations Research*, *194*(1), 365–384.

Prechelt, L. (2012). Early stopping but when? In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade: Second edition* (pp. 53–67). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_5

Rais, A., & Viana, A. (2011). Operations research in healthcare: A survey. *International transactions in operational research*, *18*(1), 1–31.

Rönnberg, E., Larsson, T., & Bertilsson, A. (2013). Automatic scheduling of nurses: What does it take in practice? In *Systems analysis tools for better health care delivery* (pp. 151–178). Springer.

Rooijen, E. (2023). *Happy planner* (Thesis).

Russell, E., Hawkins, J., & Arnold, K. A. (2012). Guidelines for successful self-scheduling on nursing units. *JONA: The Journal of Nursing Administration*, *42*(9), 408–409.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1–47.

Serengil, S. I., & Ozpinar, A. (2017). Workforce optimization for bank operation centers: A machine learning approach.

Silvestro, R., & Silvestro, C. (2000). An evaluation of nurse rostering practices in the national health service. *Journal of advanced nursing*, *32*(3), 525–535.

Solos, I. P., Tassopoulos, I. X., & Beligiannis, G. N. (2013). A generic two-phase stochastic variable neighborhood approach for effectively solving the nurse rostering problem. *Algorithms*, *6*(2), 278–308. https://doi.org/10.3390/a6020278

Stølevik, M., Nordlander, T. E., Riise, A., & Frøyseth, H. (2011). A hybrid approach for solving real-world nurse rostering problems. *Principles and Practice of Constraint Programming–CP 2011: 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings 17*, 85–99.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, *36*(2), 111–133.

Tien, J. M., & Kamiyama, A. (1982). On manpower scheduling algorithms. *SIAM Review*, *24*(3), 275–287. https://doi.org/10.1137/1024063

Tsai, C.-C., & Li, S. H. (2009). A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert systems with applications*, *36*(5), 9506–9512.

Turhan, A. M., & Bilgen, B. (2020). A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. *Computers & Industrial Engineering*, *145*, 106531. https://doi.org/https://doi.org/10.1016/j.cie.2020.106531

Uhde, A., Schlicker, N., Wallach, D. P., & Hassenzahl, M. (2020). Fairness and decision-making in collaborative shift scheduling systems. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European journal of operational research*, *226*(3), 367–385.

Voß, S. (2001). Meta-heuristics: The state of the art. *Local Search for Planning and Scheduling: ECAI 2000 Workshop Berlin, Germany, August 21, 2000 Revised Papers*, 1–23.

Wiers, V. C. S. (1997). A review of the applicability of OR and AI scheduling techniques in practice. *Omega*, *25*(2), 145–153. https://doi.org/https://doi.org/10.1016/S0305-0483(96)00050-3

Wikipedia contributors. (n.d.). *Receiver operating characteristic. - wikipedia*. https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Wright, P. D., & Mahar, S. (2013). Centralized nurse scheduling to simultaneously improve schedule cost and nurse satisfaction. *Omega*, *41*(6), 1042–1052.

Wynendaele, H., Gemmel, P., Pattyn, E., Myny, D., & Trybou, J. (2021). Systematic review: What is the impact of self-scheduling on the patient, nurse and organization? *Journal of Advanced Nursing*, *77*(1), 47–82. https://doi.org/https://doi.org/10.1111/jan.14579

Zhu, X., Zhang, L., & Huang, Z. (2014). A sparse embedding and least variance encoding approach to hashing. *IEEE transactions on image processing*, *23*(9), 3737–3750.

# Appendix A

# Hyperparameter tuning

The Python package GridSearchCV was used to tune the parameters of the selected models. The hyperparameter tuning of the models is discussed in the following order: KNN, LR, GB, RF and lastly, ANN. The grid search parameters used for the tuning of the models are shown with the best-performing parameters highlighted in dark green. The Design Of Experiments method is applied for the first four models, but a one or two-factor at a time method is used for the ANN because of the many parameters that need to be tuned and the long running time of this model.

## A.1 K-Nearest Neighbours

The main parameter for the KNN is the weight calculation and the number of neighbours considered. Both a uniform and distance method are used, and for K, the values 5, 10, 15, 20 and 50 are considered.

| Number of neighbours | Weight | F1-score |
|---|---|---|
| 5 | Uniform | 0.735476 |
| 5 | Distance | 0.732931 |
| 10 | Uniform | 0.746198 |
| 10 | Distance | 0.743924 |
| 15 | Uniform | 0.750685 |
| 15 | Distance | 0.747463 |
| 20 | Uniform | 0.752824 |
| 20 | Distance | 0.750008 |
| 25 | Uniform | 0.751423 |
| 25 | Distance | 0.75061 |
| 50 | Uniform | 0.747839 |
| 50 | Distance | 0.74686 |

## A.2 Logistic Regression

The main hyperparameters for the Logistic Regression are the penalty, the C value and the solver used.

| C | Penalty | Solver | F1-score |
|---|---|---|---|
| 0.1 | l1 | newton-cg | nan |
| 0.1 | l1 | lbfgs | nan |
| 0.1 | l1 | liblinear | 0.779433 |
| 0.1 | l1 | sag | nan |
| 0.1 | l1 | saga | 0.779583 |
| 0.1 | l2 | newton-cg | 0.779644 |
| 0.1 | l2 | lbfgs | 0.779689 |
| 0.1 | l2 | liblinear | 0.779553 |
| 0.1 | l2 | sag | 0.779674 |
| 0.1 | l2 | saga | 0.78008 |
| 0.1 | elasticnet | newton-cg | nan |
| 0.1 | elasticnet | lbfgs | nan |
| 0.1 | elasticnet | liblinear | nan |
| 0.1 | elasticnet | sag | nan |
| 0.1 | elasticnet | saga | nan |
| 1 | l1 | newton-cg | nan |
| 1 | l1 | lbfgs | nan |
| 1 | l1 | liblinear | 0.779704 |
| 1 | l1 | sag | nan |
| 1 | l1 | saga | 0.77996 |
| 1 | l2 | newton-cg | 0.779583 |
| 1 | l2 | lbfgs | 0.779478 |
| 1 | l2 | liblinear | 0.779644 |
| 1 | l2 | sag | 0.779508 |
| 1 | l2 | saga | 0.77993 |
| 1 | elasticnet | newton-cg | nan |
| 1 | elasticnet | lbfgs | nan |
| 1 | elasticnet | liblinear | nan |
| 1 | elasticnet | sag | nan |
| 1 | elasticnet | saga | nan |
| 10 | l1 | newton-cg | nan |
| 10 | l1 | lbfgs | nan |
| 10 | l1 | liblinear | 0.779629 |
| 10 | l1 | sag | nan |
| 10 | l1 | saga | 0.77996 |
| 10 | l2 | newton-cg | 0.779493 |
| 10 | l2 | lbfgs | 0.779448 |
| 10 | l2 | liblinear | 0.779568 |
| 10 | l2 | sag | 0.779493 |
| 10 | l2 | saga | 0.77996 |
| 10 | elasticnet | newton-cg | nan |
| 10 | elasticnet | lbfgs | nan |
| 10 | elasticnet | liblinear | nan |
| 10 | elasticnet | sag | nan |
| 10 | elasticnet | saga | nan |

## A.3 Gradient Boosting

The following values were tried for Gradient Boosting: learning rate, max depth, min samples leaf and the number of estimators.

| Learning rate | Max depth | Min samples leaf | N estimators | F1-score |
|---|---|---|---|---|
| 0.1 | 2 | 0.1 | 1000 | 0.813104 |
| 0.1 | 2 | 0.1 | 2000 | 0.813104 |
| 0.1 | 2 | 0.1 | 3000 | 0.812878 |
| 0.1 | 2 | 0.2 | 1000 | 0.781029 |
| 0.1 | 2 | 0.2 | 2000 | 0.781029 |
| 0.1 | 2 | 0.2 | 3000 | 0.781029 |
| 0.1 | 2 | 0.3 | 1000 | 0.724378 |
| 0.1 | 2 | 0.3 | 2000 | 0.724378 |
| 0.1 | 2 | 0.3 | 3000 | 0.724378 |
| 0.1 | 4 | 0.1 | 1000 | 0.814686 |
| 0.1 | 4 | 0.1 | 2000 | 0.814686 |
| 0.1 | 4 | 0.1 | 3000 | 0.815167 |
| 0.1 | 4 | 0.2 | 1000 | 0.782445 |
| 0.1 | 4 | 0.2 | 2000 | 0.782445 |
| 0.1 | 4 | 0.2 | 3000 | 0.782445 |
| 0.1 | 4 | 0.3 | 1000 | 0.724378 |
| 0.1 | 4 | 0.3 | 2000 | 0.724378 |
| 0.1 | 4 | 0.3 | 3000 | 0.724378 |
| 0.1 | 6 | 0.1 | 1000 | 0.814264 |
| 0.1 | 6 | 0.1 | 2000 | 0.814264 |
| 0.1 | 6 | 0.1 | 3000 | 0.814475 |
| 0.1 | 6 | 0.2 | 1000 | 0.782445 |
| 0.1 | 6 | 0.2 | 2000 | 0.782445 |
| 0.1 | 6 | 0.2 | 3000 | 0.782445 |
| 0.1 | 6 | 0.3 | 1000 | 0.724378 |
| 0.1 | 6 | 0.3 | 2000 | 0.724378 |
| 0.1 | 6 | 0.3 | 3000 | 0.724378 |
| 0.5 | 2 | 0.1 | 1000 | 0.812366 |
| 0.5 | 2 | 0.1 | 2000 | 0.812366 |
| 0.5 | 2 | 0.1 | 3000 | 0.812728 |
| 0.5 | 2 | 0.2 | 1000 | 0.780653 |
| 0.5 | 2 | 0.2 | 2000 | 0.780653 |
| 0.5 | 2 | 0.2 | 3000 | 0.780653 |
| 0.5 | 2 | 0.3 | 1000 | 0.72632 |
| 0.5 | 2 | 0.3 | 2000 | 0.72632 |
| 0.5 | 2 | 0.3 | 3000 | 0.72632 |
| 0.5 | 4 | 0.1 | 1000 | 0.808436 |
| 0.5 | 4 | 0.1 | 2000 | 0.808436 |
| 0.5 | 4 | 0.1 | 3000 | 0.809942 |
| 0.5 | 4 | 0.2 | 1000 | 0.780291 |
| 0.5 | 4 | 0.2 | 2000 | 0.780291 |
| 0.5 | 4 | 0.2 | 3000 | 0.780291 |
| 0.5 | 4 | 0.3 | 1000 | 0.72632 |
| 0.5 | 4 | 0.3 | 2000 | 0.72632 |
| 0.5 | 4 | 0.3 | 3000 | 0.72632 |
| 0.5 | 6 | 0.1 | 1000 | 0.806734 |
| 0.5 | 6 | 0.1 | 2000 | 0.806734 |
| 0.5 | 6 | 0.1 | 3000 | 0.809581 |
| 0.5 | 6 | 0.2 | 1000 | 0.780291 |
| 0.5 | 6 | 0.2 | 2000 | 0.780291 |
| 0.5 | 6 | 0.2 | 3000 | 0.780291 |
| 0.5 | 6 | 0.3 | 1000 | 0.72632 |
| 0.5 | 6 | 0.3 | 2000 | 0.72632 |
| 0.5 | 6 | 0.3 | 3000 | 0.72632 |
| 1 | 2 | 0.1 | 1000 | 0.808948 |
| 1 | 2 | 0.1 | 2000 | 0.808948 |
| 1 | 2 | 0.1 | 3000 | 0.810665 |
| 1 | 2 | 0.2 | 1000 | 0.779086 |
| 1 | 2 | 0.2 | 2000 | 0.779086 |
| 1 | 2 | 0.2 | 3000 | 0.779086 |
| 1 | 2 | 0.3 | 1000 | 0.72617 |
| 1 | 2 | 0.3 | 2000 | 0.72617 |
| 1 | 2 | 0.3 | 3000 | 0.72617 |
| 1 | 4 | 0.1 | 1000 | 0.802804 |
| 1 | 4 | 0.1 | 2000 | 0.802804 |
| 1 | 4 | 0.1 | 3000 | 0.805891 |
| 1 | 4 | 0.2 | 1000 | 0.776496 |
| 1 | 4 | 0.2 | 2000 | 0.776496 |
| 1 | 4 | 0.2 | 3000 | 0.776496 |
| 1 | 4 | 0.3 | 1000 | 0.72617 |
| 1 | 4 | 0.3 | 2000 | 0.72617 |
| 1 | 4 | 0.3 | 3000 | 0.72617 |
| 1 | 6 | 0.1 | 1000 | 0.803301 |
| 1 | 6 | 0.1 | 2000 | 0.803301 |
| 1 | 6 | 0.1 | 3000 | 0.804551 |
| 1 | 6 | 0.2 | 1000 | 0.776496 |
| 1 | 6 | 0.2 | 2000 | 0.776496 |
| 1 | 6 | 0.2 | 3000 | 0.776496 |
| 1 | 6 | 0.3 | 1000 | 0.72617 |
| 1 | 6 | 0.3 | 2000 | 0.72617 |
| 1 | 6 | 0.3 | 3000 | 0.72617 |

TABLE A.1: Hyperparameter tuning GB

## A.4 Random Forest

The parameters selected for the Random forest are similar to GB with the exception that instead of learning rate, we tune based on class weight.

| Class weight | max depth | min samples leaf | n_estimators | F1-score |
|---|---|---|---|---|
| Balanced | 5 | 0.1 | 1000 | 0.753246 |
| Balanced | 5 | 0.1 | 2000 | 0.753276 |
| Balanced | 5 | 0.1 | 3000 | 0.753065 |
| Balanced | 5 | 0.2 | 1000 | 0.663947 |
| Balanced | 5 | 0.2 | 2000 | 0.664534 |
| Balanced | 5 | 0.2 | 3000 | 0.664549 |
| Balanced | 5 | 0.3 | 1000 | 0.604961 |
| Balanced | 5 | 0.3 | 2000 | 0.604946 |
| Balanced | 5 | 0.3 | 3000 | 0.604946 |
| Balanced | 10 | 0.1 | 1000 | 0.753246 |
| Balanced | 10 | 0.1 | 2000 | 0.753276 |
| Balanced | 10 | 0.1 | 3000 | 0.753065 |
| Balanced | 10 | 0.2 | 1000 | 0.663947 |
| Balanced | 10 | 0.2 | 2000 | 0.664534 |
| Balanced | 10 | 0.2 | 3000 | 0.664549 |
| Balanced | 10 | 0.3 | 1000 | 0.604961 |
| Balanced | 10 | 0.3 | 2000 | 0.604946 |
| Balanced | 10 | 0.3 | 3000 | 0.604946 |
| Balanced | 15 | 0.1 | 1000 | 0.753246 |
| Balanced | 15 | 0.1 | 2000 | 0.753276 |
| Balanced | 15 | 0.1 | 3000 | 0.753065 |
| Balanced | 15 | 0.2 | 1000 | 0.663947 |
| Balanced | 15 | 0.2 | 2000 | 0.664534 |
| Balanced | 15 | 0.2 | 3000 | 0.664549 |
| Balanced | 15 | 0.3 | 1000 | 0.604961 |
| Balanced | 15 | 0.3 | 2000 | 0.604946 |
| Balanced | 15 | 0.3 | 3000 | 0.604946 |
| Balanced subsample | 5 | 0.1 | 1000 | 0.753321 |
| Balanced subsample | 5 | 0.1 | 2000 | 0.753653 |
| Balanced subsample | 5 | 0.1 | 3000 | 0.753065 |
| Balanced subsample | 5 | 0.2 | 1000 | 0.663962 |
| Balanced subsample | 5 | 0.2 | 2000 | 0.663917 |
| Balanced subsample | 5 | 0.2 | 3000 | 0.664338 |
| Balanced subsample | 5 | 0.3 | 1000 | 0.604961 |
| Balanced subsample | 5 | 0.3 | 2000 | 0.604946 |
| Balanced subsample | 5 | 0.3 | 3000 | 0.604946 |
| Balanced subsample | 10 | 0.1 | 1000 | 0.753321 |
| Balanced subsample | 10 | 0.1 | 2000 | 0.753653 |
| Balanced subsample | 10 | 0.1 | 3000 | 0.753065 |
| Balanced subsample | 10 | 0.2 | 1000 | 0.663962 |
| Balanced subsample | 10 | 0.2 | 2000 | 0.663917 |
| Balanced subsample | 10 | 0.2 | 3000 | 0.664338 |
| Balanced subsample | 10 | 0.3 | 1000 | 0.604961 |
| Balanced subsample | 10 | 0.3 | 2000 | 0.604946 |
| Balanced subsample | 10 | 0.3 | 3000 | 0.604946 |
| Balanced subsample | 15 | 0.1 | 1000 | 0.753321 |
| Balanced subsample | 15 | 0.1 | 2000 | 0.753653 |
| Balanced subsample | 15 | 0.1 | 3000 | 0.753065 |
| Balanced subsample | 15 | 0.2 | 1000 | 0.663962 |
| Balanced subsample | 15 | 0.2 | 2000 | 0.663917 |
| Balanced subsample | 15 | 0.2 | 3000 | 0.664338 |
| Balanced subsample | 15 | 0.3 | 1000 | 0.604961 |
| Balanced subsample | 15 | 0.3 | 2000 | 0.604946 |
| Balanced subsample | 15 | 0.3 | 3000 | 0.664201 |

TABLE A.2: Hyperpameter tuning RF

## A.5 Artificial Neural Network

ANN has many parameters that can be tuned. Because two layers are often enough to make accurate predictions, we have selected two hidden layers for the ANN model and activation of 'relu' for the input and hidden layer and 'sigmoid' for the output layer. As explained in the chapter introduction, we use one or two factor at a time to find the optimal values of the ANN model.

First, we optimise over the batch size and epoch, then the optimiser, next the dropout rate and lastly, the number of neurons in the input and hidden layer.

| Batch size | Epochs | Mean test score |
|---|---|---|
| 10 | 10 | 0.81 |
| 10 | 50 | 0.81 |
| 10 | 100 | 0.81 |
| 20 | 10 | 0.81 |
| 20 | 50 | 0.71 |
| 20 | 100 | 0.81 |
| 40 | 10 | 0.81 |
| 40 | 50 | 0.81 |
| 40 | 100 | 0.74 |
| 60 | 10 | 0.72 |
| 60 | 50 | 0.81 |
| 60 | 100 | 0.74 |
| 80 | 10 | 0.65 |
| 80 | 50 | 0.81 |
| 80 | 100 | 0.82 |
| 100 | 10 | 0.83 |
| 100 | 50 | 0.84 |
| 100 | 100 | 0.85 |

TABLE A.3: Hyperparameter tuning batch size and epochs

| Optimiser | Mean test score |
|---|---|
| SGD | 0.75 |
| RMSprop | 0.81 |
| Adagrad | 0.78 |
| Adadelta | 0.63 |
| Adam | 0.81 |
| Adamax | 0.81 |
| Nadam | 0.81 |

TABLE A.4: Hyperparameter tuning optimiser

| Dropout rate | Mean test score |
|---|---|
| 0 | 0.81 |
| 0.1 | 0.81 |
| 0.2 | 0.81 |
| 0.3 | 0.8 |
| 0.4 | 0.8 |
| 0.5 | 0.8 |
| 0.6 | 0.79 |
| 0.7 | 0.63 |
| 0.8 | 0.57 |
| 0.9 | 0.64 |

TABLE A.5: Hyperparameter tuning drop out rate

| Neurons input layer | Neurons hidden layer | |
|---|---|---|
| 25 | 1 | 0.83 |
| 25 | 25 | 0.82 |
| 25 | 50 | 0.83 |
| 25 | 100 | 0.843 |
| 50 | 1 | 0.82 |
| 50 | 25 | 0.81 |
| 50 | 50 | 0.8 |
| 50 | 100 | 0.81 |
| 75 | 1 | 0.79 |
| 75 | 25 | 0.82 |
| 75 | 50 | 0.83 |
| 75 | 100 | 0.83 |
| 100 | 1 | 0.79 |
| 100 | 25 | 0.81 |
| 100 | 50 | 0.82 |
| 100 | 100 | 0.83 |

TABLE A.6: Hyperparameter neurons

# Appendix B

# Feasible schedule algorithm

This appendix gives the pseudocode to make a schedule feasible.

**Algorithm 1** Feasible schedule

**while** violations of 2 weekends off per 6 weeks ($c_1$) $> 0$ **do**
    Find the first weekend that violates $c_1$ constraint.
    Set the shifts in this weekend to zero.
**end while**
**while** violation maximum shifts in a week 0.5 fte ($c_2$) $> 0$ **do**
    Find the first shift that violates $c_2$ constraint.
    Set this shift to zero.
**end while**
**while** violation maximum shifts in a row ($c_3$) $> 0$ **do**
    Find the first shift that violates $c_3$ constraint.
    Set this shift to zero.
**end while**
**while** violation leave wishes ($c_4$) $> 0$ **do**
    Find the first shift that violates $c_4$ constraint.
    Set this shift to zero.
**end while**
**while** violations of 2 weekends off per 6 weeks ($c_1$) $> 0$ **do**
    Find the first shift that violates $c_5$ constraint.
    Set this shift to zero.
**end while**

# Appendix C

# Confusion Matrices

## C.1 Confusion Matrix IC PI

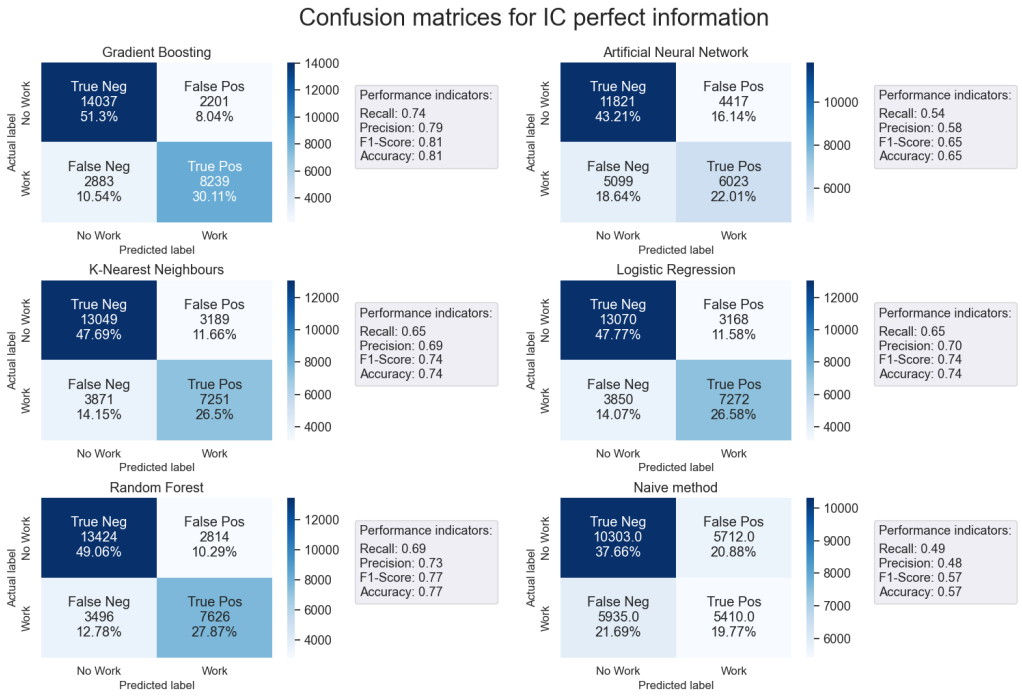



FIGURE C.1: Confusion Matrix IC PI

## C.2 Confusion Matrix IC updating

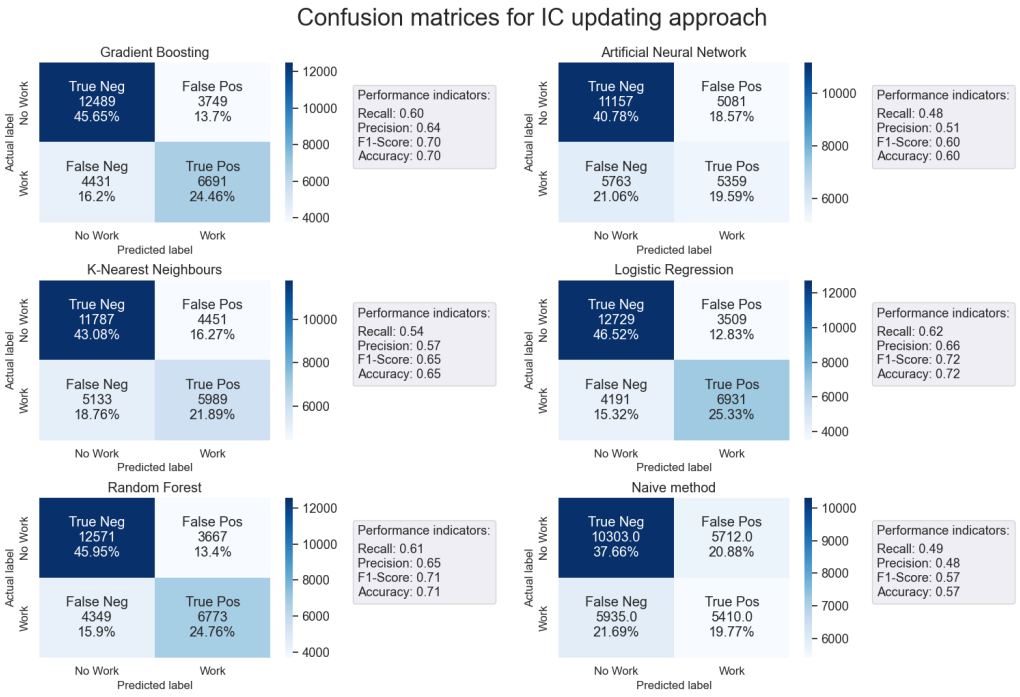



FIGURE C.2: Confusion Matrix IC updating

## C.3 Confusion Matrix radiology PI

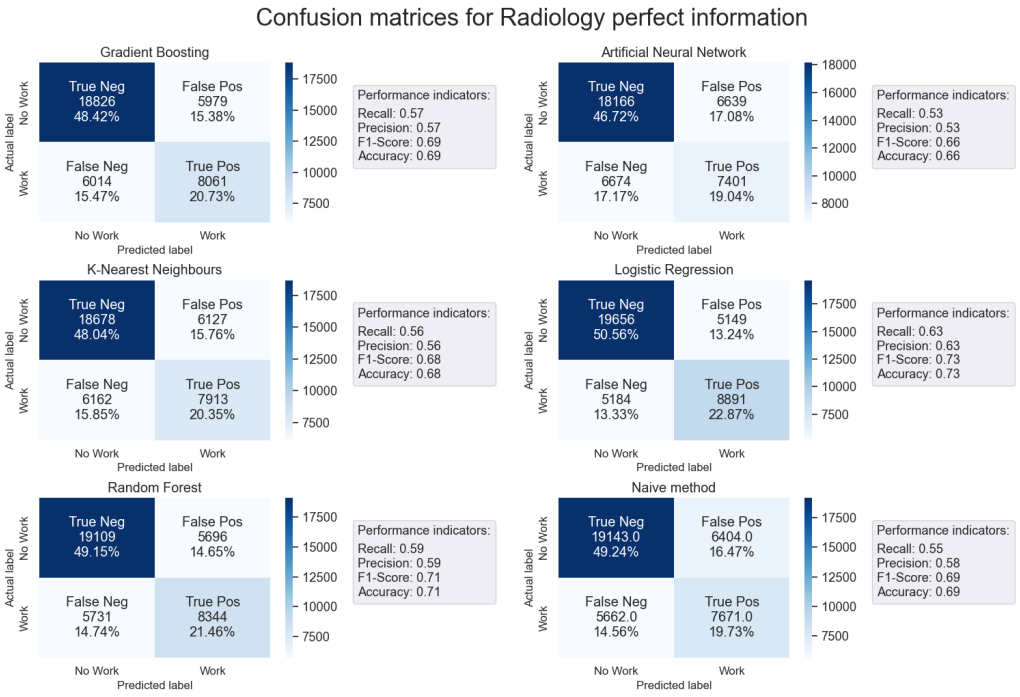



FIGURE C.3: Confusion Matrix radiology PI

## C.4 Confusion Matrix radiology updating

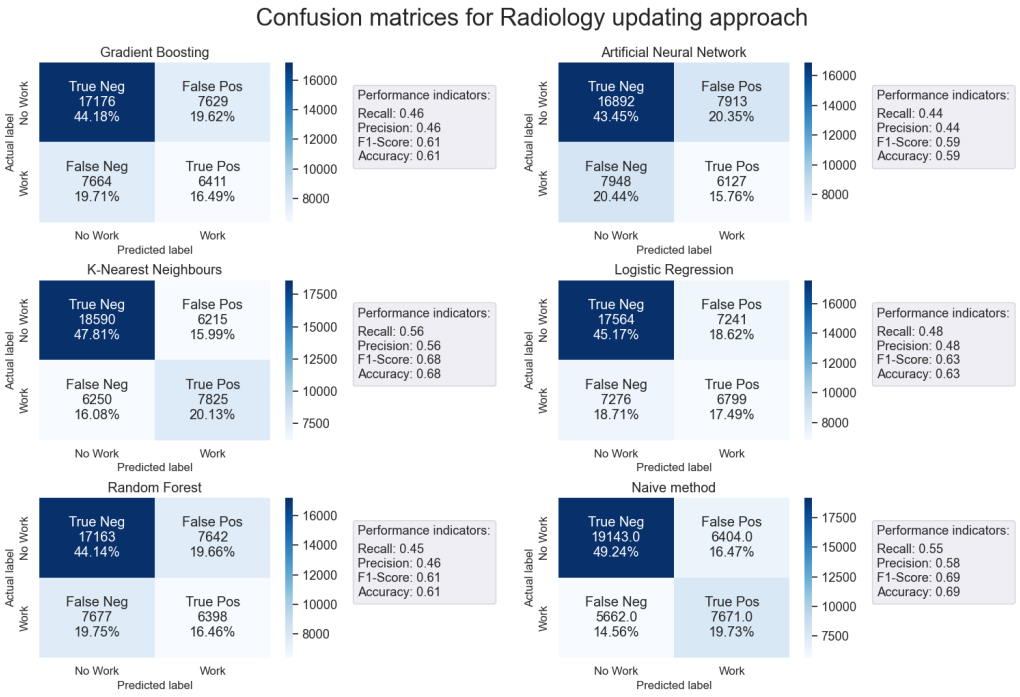



FIGURE C.4: Confusion Matrix radiology updating

## C.5 Confusion Matrix haematology PI

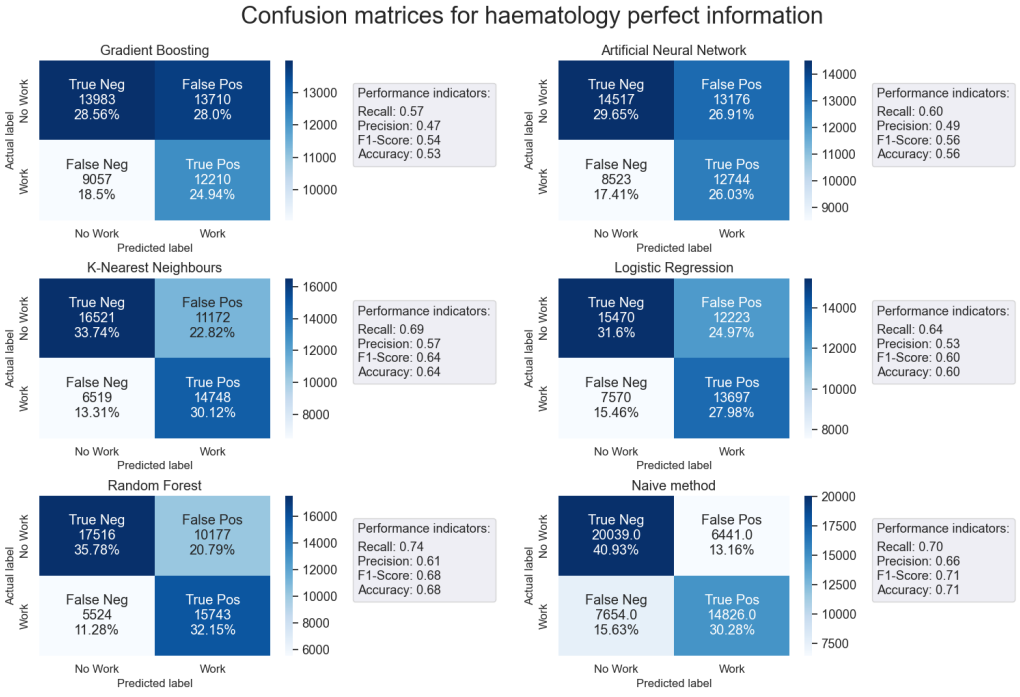



FIGURE C.5: Confusion Matrix haematology PI

## C.6 Confusion Matrix haematology updating

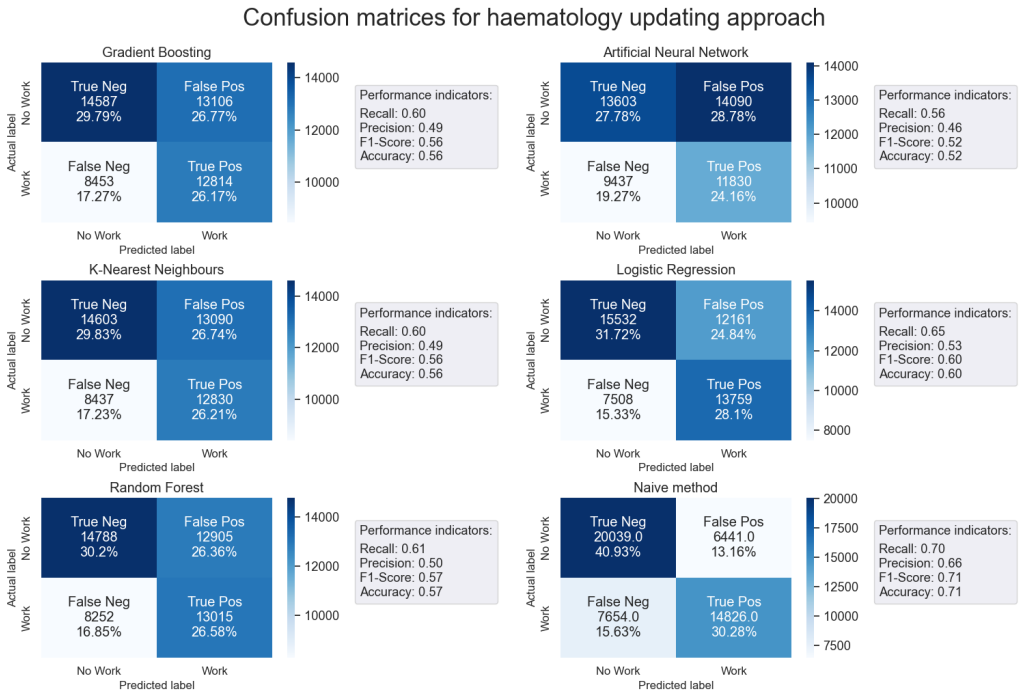



FIGURE C.6: Confusion Matrix haematology PI

# Appendix D

# SHAP value analysis

This appendix shows the SHAP value results for the departments and prediction approach combinations, except for the PI approach for the IC department since this was already shown in the main document.

## D.1 IC update

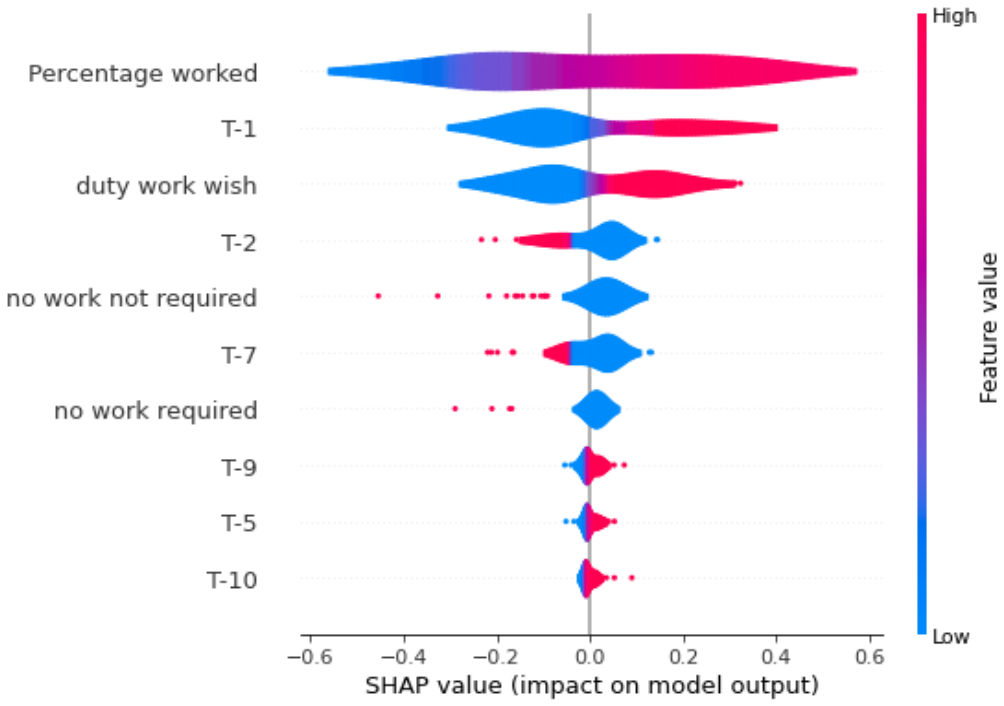



FIGURE D.1: SHAP value analysis IC update LR model

## D.2 Radiology PI

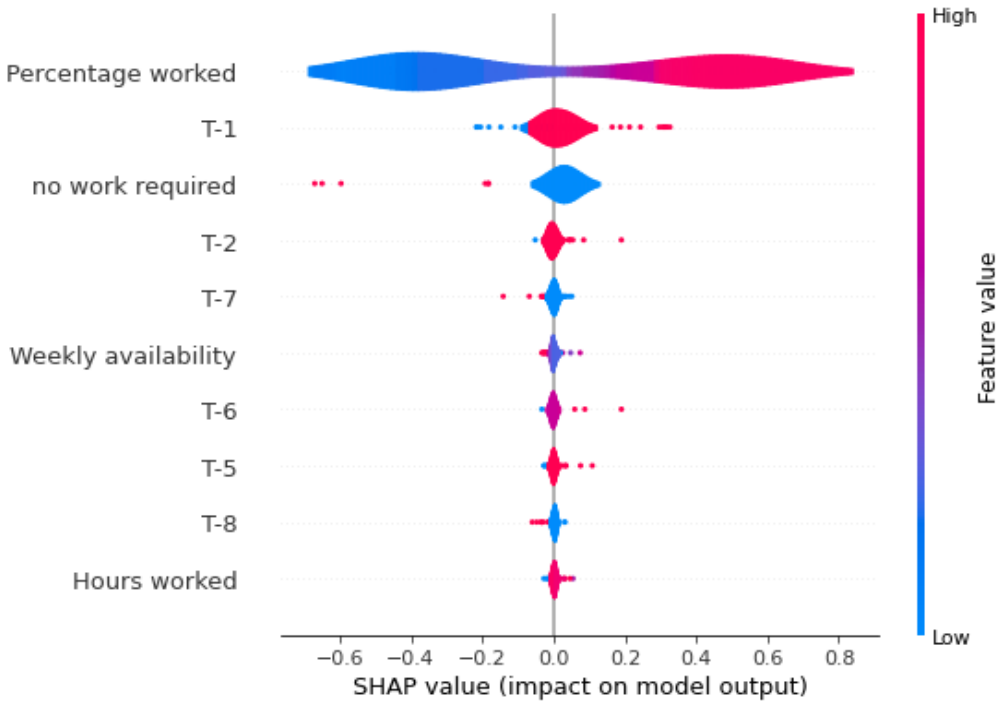



FIGURE D.2: SHAP value analysis Radiology PI LR model

## D.3 Radiology update

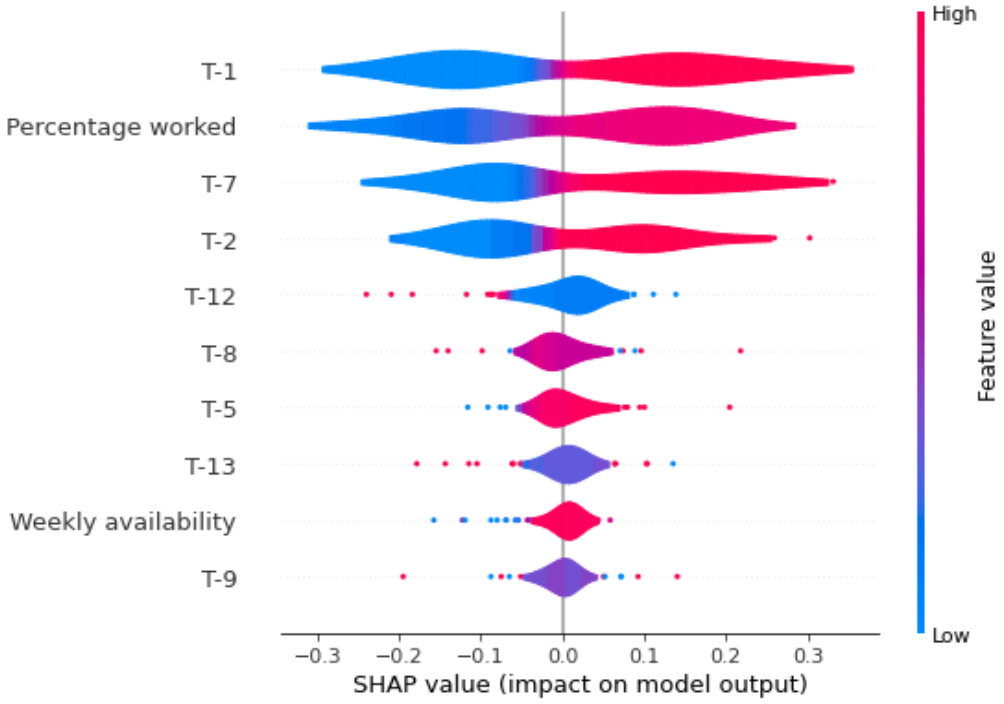



FIGURE D.3: SHAP value analysis radiology update KNN model

## D.4 Haematology PI

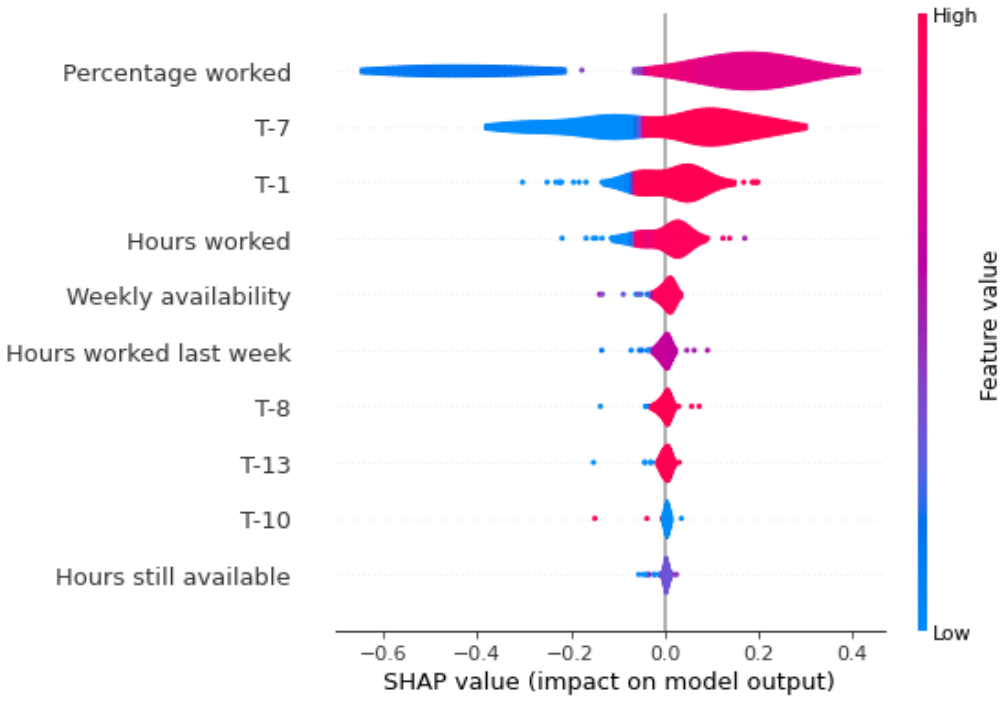



FIGURE D.4: SHAP value analysis haematology PI RF model

## D.5 Haematology update

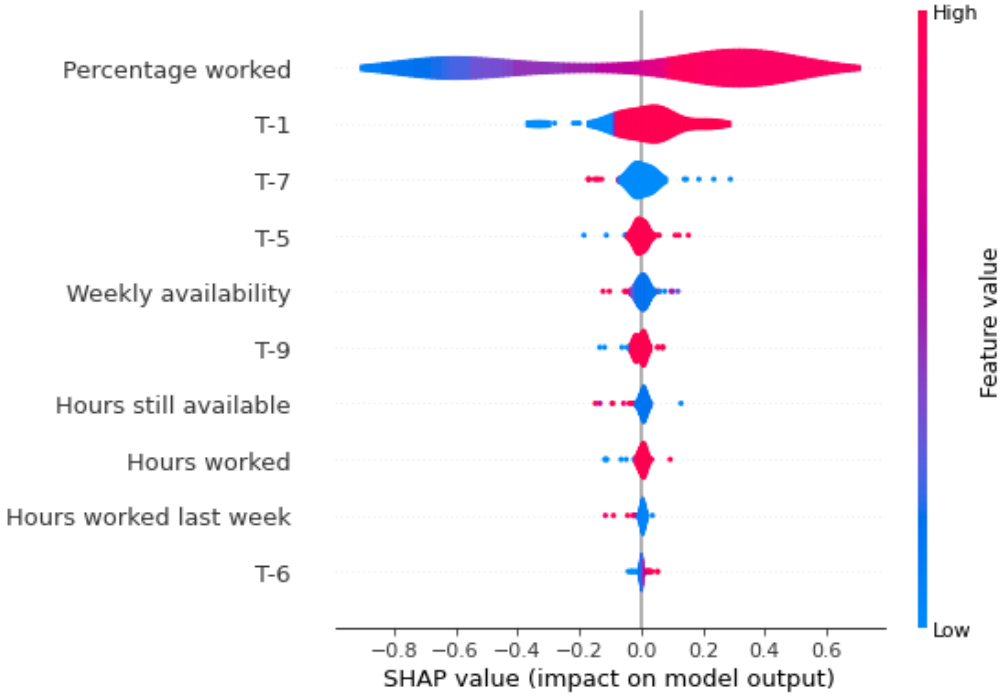



FIGURE D.5: SHAP value analysis haematology update LR model

# Appendix E

# Improvement heuristic results

## E.1 Radiology PI

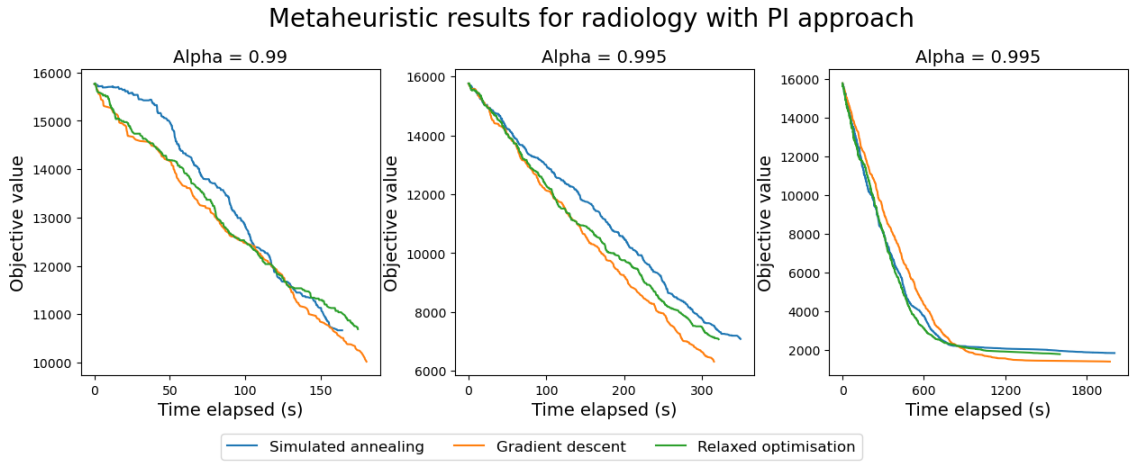



FIGURE E.1: Objective value radiology department with PI approach

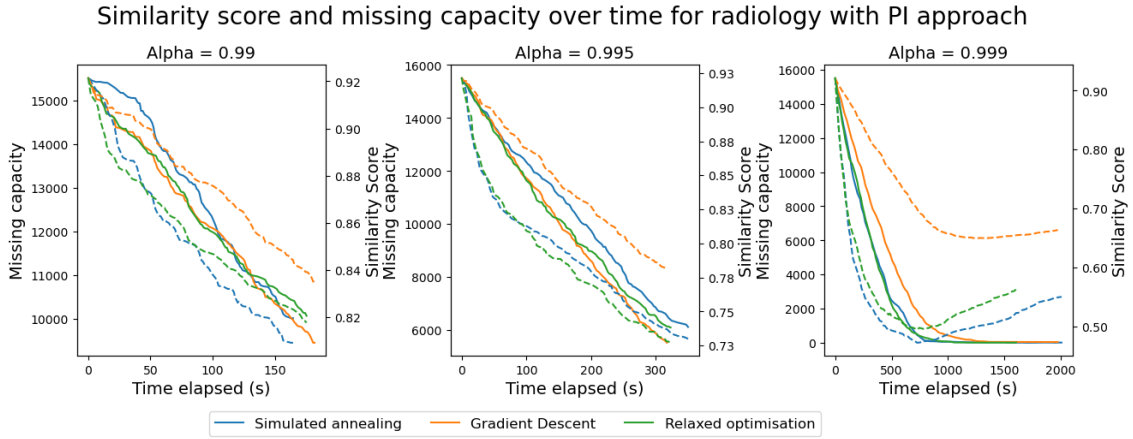



FIGURE E.2: Similarity score and missing capacity over time for Radiology department with PI approach

## E.2 Radiology update

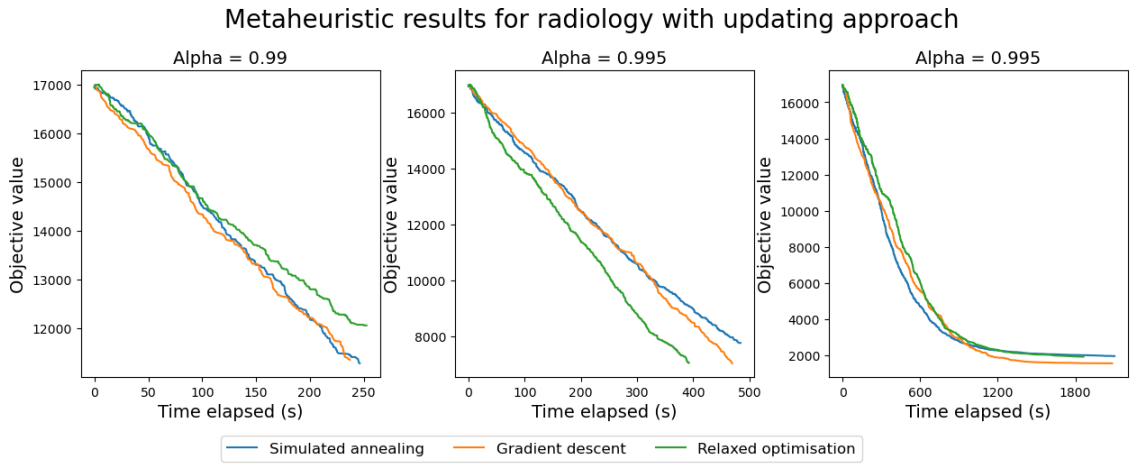



FIGURE E.3: Objective value radiology department with updating approach

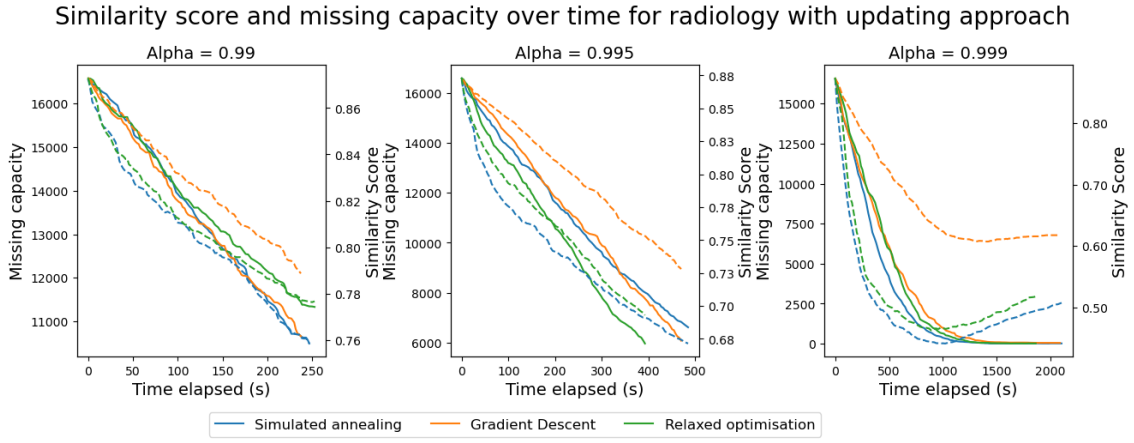



FIGURE E.4: Similarity score and missing capacity over time for Radiology department with update approach

## E.3 Haematology PI

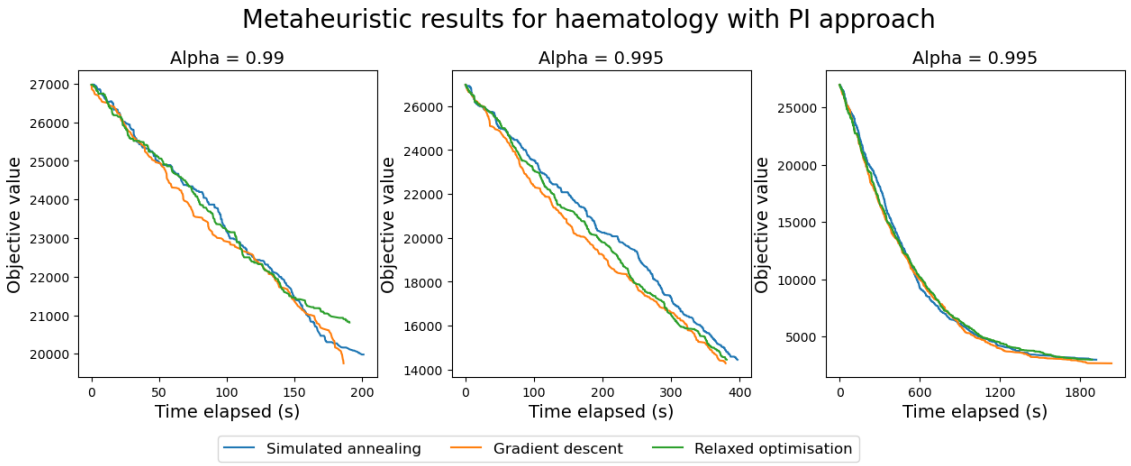



FIGURE E.5: Objective value haematology department with PI approach

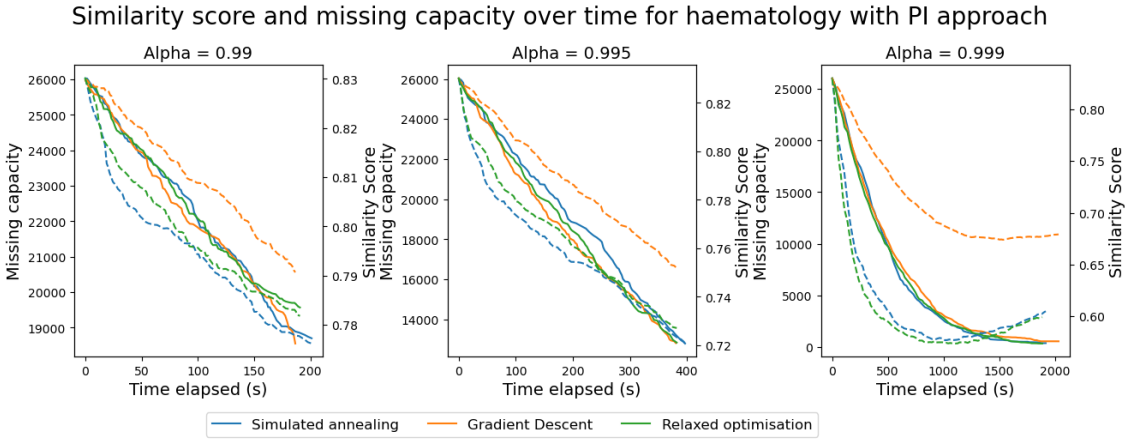



FIGURE E.6: Similarity score and missing capacity over time for Haematology department with PI approach

## E.4 Haematology update

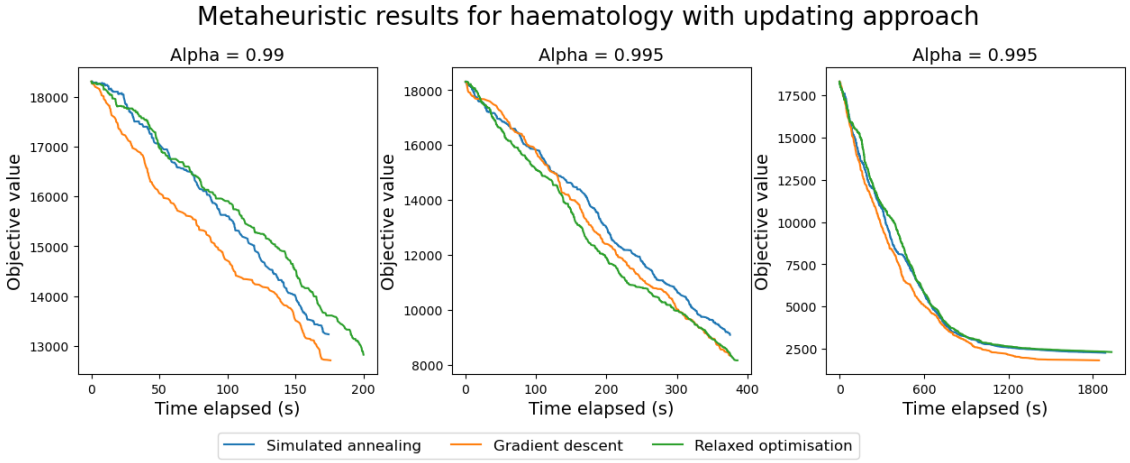



FIGURE E.7: Objective value haematology department with updating approach

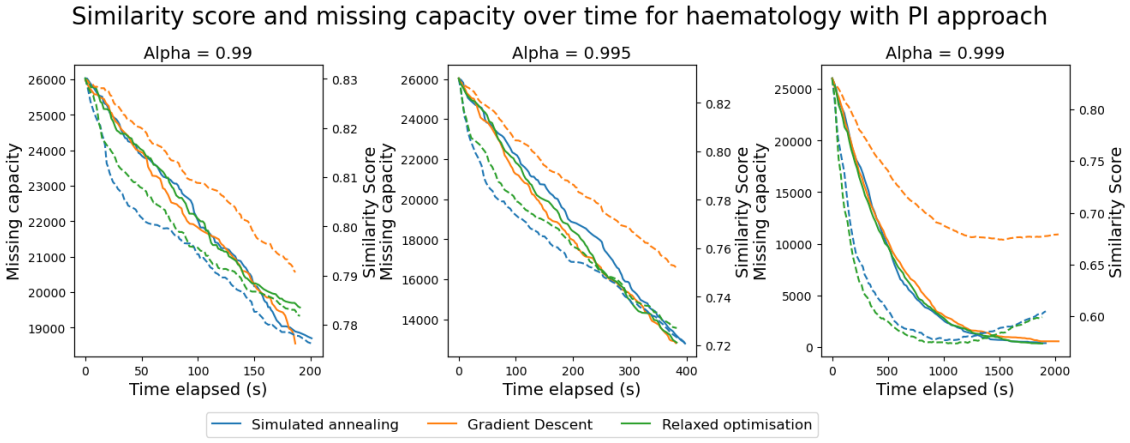



FIGURE E.8: Similarity score and missing capacity over time for Haematology department with updating approach

## E.5 Radiology naive

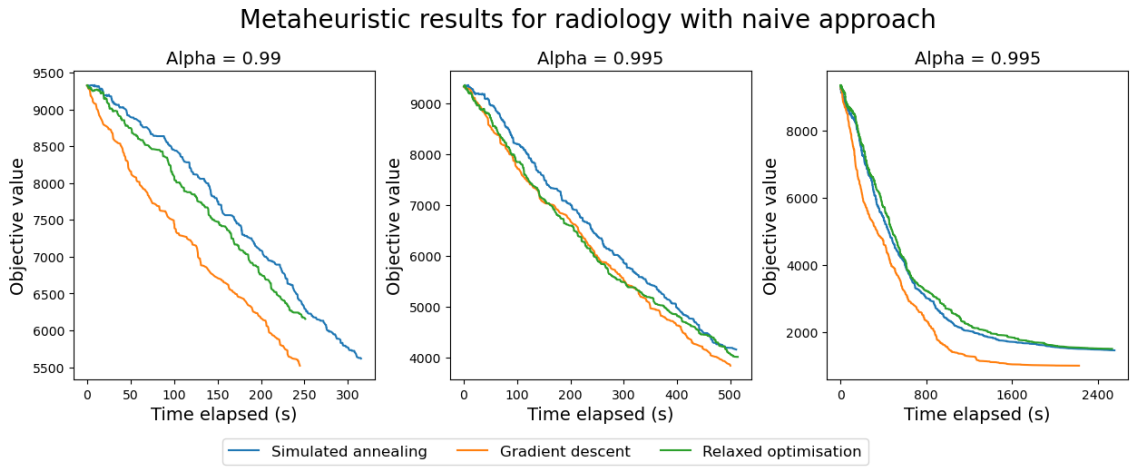



FIGURE E.9: Similarity score and missing capacity over time for radiology department with naive method

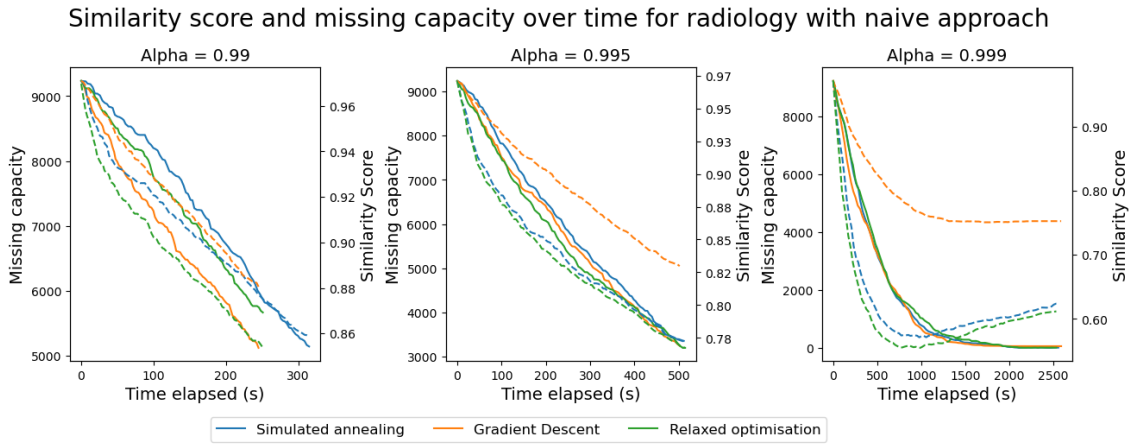



FIGURE E.10: Similarity score radiology department with naive method

## E.6 Haematology naive

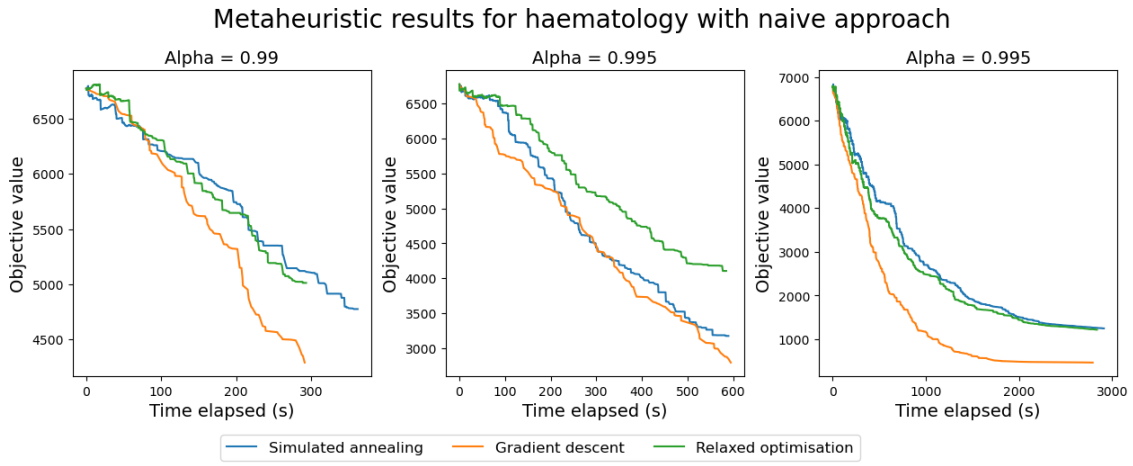



FIGURE E.11: Objective value haematology department with naive method

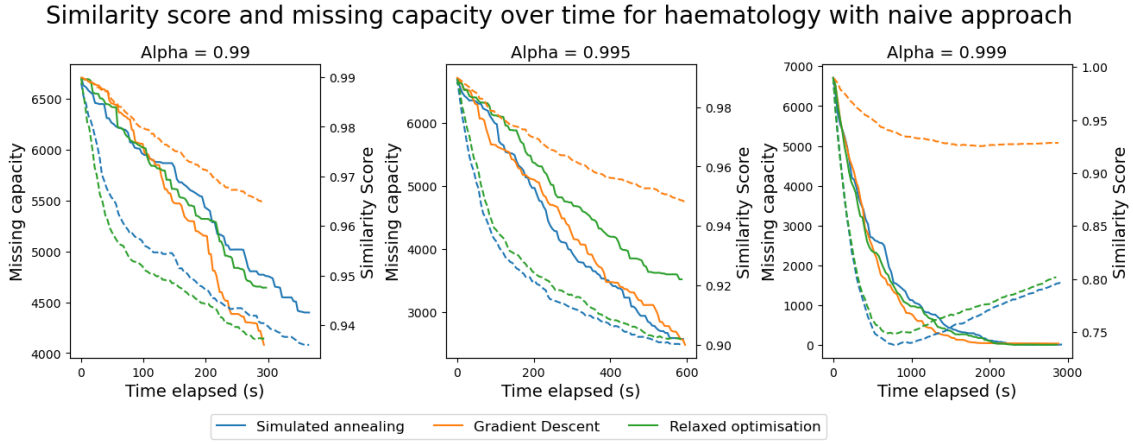



FIGURE E.12: similarity score aematology department with naive method