
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering, Mathematics & Computer Science**


# Predicting readmission of neonates to an ICU using data mining


**Betina S. Markova**


**M.Sc. Thesis**
**March 2021**


**Committee:**
dr. M. Poel
dr. ing. G. Englebienne
prof. dr. D. Heyeen

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

## Acknowledgements

I would like to express my gratitude to my supervisor, dr. Mannes Poel, for his constant guidance and useful suggestions and feedback throughout this project. I also would like to thank my 2nd supervisor, dr. ing. Gwenn Englebienne, for his valuable input and comments. Their contribution was crucial in improving the quality of this thesis. Furthermore, I express my appreciation to prof. dr. Dirk Heyeen for being part of my graduation committee. Last but not least, I would like to thank my family for facilitating my pursuit of higher education and my close friends for supporting me on the way.

# Abstract

Intensive Care Units are specialized hospital wards where critically ill patients receive enhanced medical treatment. The beds in ICUs are limited, which sometimes forces healthcare professionals to make a delicate decision of discharging a patient to make room for other seriously ill patients. However, a premature discharge can be a reason for patient readmission, which is associated with increased length of stay and deterioration of a patient's condition. Therefore, it is important to identify patients at high risk of readmission to guide the decision making concerning the discharge of the patient. While readmission prediction has been extensively studied in adult patients, little attention is paid to the readmission of neonate patients. Neonates are newborns in their first 4 weeks after birth. There are different reasons for admitting a newborn to an Intensive Care Unit, including preterm birth, low birth weight, or health conditions such as breathing troubles, heart problems, infections, etc. Yet, there is a lack of studies investigating the potentials of data-driven decision support system in neonatal readmission. Previous studies have attempted to simply explain the statistical connections between different variables and the readmission outcome. However, most works have not extended their analysis to measure predictive performance.

This study extends previous research by implementing three distinct classification models – Logistic Regression, Gradient Boosted Decision Trees, and Neural Network, for predicting the readmission of neonatal patients to Intensive Care Units. It is among the first studies applying machine learning techniques to predict neonatal readmissions. The study is carried out over an anonymized dataset collected over seven years in a public pediatric hospital in Zhejiang, China. The predictive analysis of 30-day readmission is formulated as a binary classification problem. However, because readmission is a much less frequent event than no readmission, the data is highly skewed towards the negative class. In this study, readmissions account for only 4.8% of the samples. This class imbalance causes difficulties during training and validation of a model. During training, the readmission class is underrepresented, hence, the model gets biased towards the majority class. Two different approaches for dealing with class imbalance are used - one is to adjust the weights during learning while the other is a data level approach - ADASYN oversampling technique, where synthetic samples are generated for the minority class until the class balance is restored. Moreover, certain performance metrics used for validation of the model, such as accuracy, are strongly influenced by the majority class correct classification, hence, AUROC is one of the metrics used to express the performance of the implemented models.

This study reports classification results achieved with models before and after class correction with ADASYN. The results showed that the Neural Network is the best model developed in this study, with an AUROC score of 0.71, which is an acceptable value for AUC in general. Although, comparisons with literature indicate that the data and models developed in this study are subject to improvement. Surprisingly the worst performing model is Gradient Boosted Decision Trees, achieving an AUROC score of only 0.65. Furthermore, results showed that the imbalance correction technique, ADASYN, did not improve the AUROC score for any of the implemented classification algorithms. It even had a degrading effect on Logistic Regression. Therefore, a suggestion for future research is to further explore other class imbalance handling techniques and models.

# Contents

# 1 Introduction

Intensive Care Units (ICUs) are specialized hospital wards where critically ill patients receive intensive, specialized medical treatment while their condition is being under enhanced monitoring [1]. As such, ICUs have higher costs associated with them, reflecting the high resource consumption and staffing needs. Depending on the specific medical requirements and patients, there are different types of ICUs established such as the Coronary Care Unit (CICU), Neonatal Intensive Care Unit (NICU), Pediatric Intensive Care Unit (PICU), to name a few. ICUs are among the most critically functioning environments in a hospital. In fact, the decisions taken in an ICU can increase or reduce the life-threatening risk for the patients. Because ICU beds are limited, healthcare professionals may need to make a delicate decision of discharging a patient from ICU to make room for other seriously ill patients [2]. While delayed discharge can result in reduced capacity and therefore delayed admission of patients who require critical care [3], a premature discharge can be a reason for patient readmission, which is associated with increased length of stay, costs, and mortality rates [4]. Therefore, it is necessary to investigate the possibilities of developing predictive tools and alerts to help ICU physicians avoid premature ICU discharges. Foreseeing which patients are at risk of ICU readmission would enable the ICU team to best plan the discharge and the ongoing care outside of the ICU [5].

The event of readmission is defined as a nonscheduled patient admission to a hospital ward within a short period after discharge. The standard benchmark to count a patient return as readmission is 30 days. ICU readmission rates have been used as a marker of hospital performance. To encourage the prevention of readmissions, both public and private funding policies have introduced financial penalties to hospitals that have excessive risk-adjusted readmission rates [6]. Not only are there strong financial reasons to avoid readmissions, but the readmissions to ICUs indicate a deterioration of a patient's condition and are associated with higher mortality rates, putting a burden on both patients and healthcare systems. It has been argued that a large part of the readmissions are preventable, increasing the worth of investment in the prediction of patient readmission [7]. There has been an active line of research to establish decision support tools, focusing on the adult patient population, such as scores that depict the severity of illness or even classification models that make use of the fast-growing technological advances in the field of data mining. Recent works favour the application of predictive machine learning approaches, formulating the readmission prediction as a binary classification problem. For example, the literature reports results from Logistic Regression [3], [7], Support Vector Machine [8], Neural Network [9], etc.

The readmission prediction is intrinsically class-imbalanced, which makes the prediction task difficult. As the name suggests, class imbalance refers to skewed data distribution where the classes are not equally represented – one class includes fewer samples than the other class. Usually, the positive class is underrepresented. In such cases, most classification models will focus on the negative class, which is not the interest group of research. Despite the implications that class imbalance poses, only a small number of the reviewed predictive models adjusted for it. Therefore, a further investigation into handling imbalanced electronic health record (EHR) data is needed.

Despite the long list of studies about hospital readmission for adult patients, when it comes to the pediatric patient population and particularly neonates, not that much has been done regarding readmission prevention. This is not surprising considering that Neonatal medicine is a relatively recent advancement, becoming an accepted medical discipline only in the 1960s

[10]. Neonates are newborns in their first 4 weeks after birth. After a month, a baby is no longer considered a neonate. There could be different reasons for admitting a newborn to an intensive care unit, including preterm birth, low birth weight, or health conditions such as breathing troubles, heart problems, infections, etc. Research indicates that the risk factors associated with readmission of infants to an ICU also include the operative method of birth, maternal diabetes, hypertension, ethnicity, gestational age and socioeconomic status, etc. [11], [12]. While factors suspected to have an impact on neonatal readmission have been explored in the literature, there are not sufficient studies investigating the potentials of data-driven decision support in neonatal readmission. Fortunately, the widespread adoption of electronic health record (EHR) systems can help to address these shortcomings. Thanks to the positive trends that data mining applications in healthcare have shown, now more and more digitally available medical data becomes freely accessible for research purposes. Therefore, the objective of this study is to develop and validate several classification models to predict 30-day neonatal readmission based on a recently released freely accessible pediatric-specific database. Considering the intrinsic class-imbalanced problem, an oversampling method was investigated as a method to compensate for the imbalance.

## 1.1 Problem Statement

The main goal of this research is to predict whether a neonatal patient will be readmitted to an intensive care unit. The problem is formalized as a binary classification task.

The main research questions are defined as:

**RQ1: To what extent, one can predict readmission of neonates to an intensive care unit using machine learning methods (Logistic Regression, Gradient Boosted Decision Tree, and Neural Network)?**

RQ1.1: Given the class-imbalanced dataset, does class balancing during data preprocessing improve the classification?

**RQ2: Which of the machine learning algorithms performs the best in the classification task?**

RQ2.1: Which features used in this study have the greatest impact on the performance of the model from RQ2?

## 1.2 Outline

The remainder of this thesis is structured as follows. Chapter 2 presents the background information. Chapter 3 reviews the related work. Then the general methodology is introduced in Chapter 4. Chapter 5 describes, in detail, the dataset preparation and variables used in the study. Chapter 6 reports the models' design, followed by the results presented in Chapter 7. The discussion takes place in Chapter 8, including suggestions for future work and the limitations of this study. The research conclusions are summarized in Chapter 9.

# 2 Background

In this section, the background useful for this study is introduced.

## 2.1 Data Mining

Data mining is defined as an interdisciplinary field that involves methods of machine learning and statistics to extract useful information from large and complex datasets. In this context, useful information means information that has been previously unknown [13] and is too subtle for humans to detect without the means of intelligent methods. Data mining has the task to also transform this information into a comprehensive structure that is both understandable and suitable for further use (to build predictive models etc.) [14].

### 2.1.1 Data Mining Techniques

Understanding the different data mining algorithms and their functionalities is a crucial step before applying data mining to any kind of data. Data mining techniques can be broadly separated into two types - predictive (supervised learning) and descriptive (unsupervised learning) [15].

*Descriptive data mining* is used to determine the similarities and to find unknown patterns or relationships within the data. The nature of descriptive data mining is mainly explanatory and the emphasis lies on transforming a huge amount of data into meaningful information that is presented in a comprehensible way. This type of data mining includes techniques such as clustering, association, summarizing, and sequence discovery. *Predictive data mining*, on the other hand, is used to forecast future behaviour or results. Therefore it employs prediction rules such as classification, prediction, regression and time series analysis.

In this paper, the focus is on the predictive data mining techniques due to the nature of the task - readmission prediction. The following sections provide a brief overview of the algorithms relevant to the task.

**Classification** As the name suggests, classification is used to group data into predefined classes, where each class is an attribute or feature from the dataset. The process of classification consists of two steps - training and testing. While in the training phase, a classification model, with classification rules, is built and utilized on a separate set of data - training data that contains class labels. In comparison to the testing step, which is relatively simple, the training step is a complex and very computationally expensive process. The testing phase evaluates the accuracy of the classifier or its ability to classify unknown data. This is done on testing data that was not included in the training process and it is either labelled data or unlabelled data to suit the objective of the testing. The aim of the testing can be to simply evaluate the accuracy or the ability of the classifier to predict on unknown data (also called validation process).

There are different classifiers:

**Naive Bayesian classifier**
Naive Bayesian classifier is considered to be the simplest algorithm among classification algorithms. It is based on the Bayes theorem (1) and its name, naive, comes form the assumption that all attributes are independent of one another. This is both its biggest advantage because it can easily handle data with lots of attributes, but also its biggest disadvantage as this assumption is not realistic in many cases. However, in general, Naive Bayes has shown very good classification accuracy despite its overly-simplified assumption and it is widely used in medical data mining [15].

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{1}$$

**Neural Network**
Neural Network (NN) was developed in 1957 with the attempt to emulate the function of the neurons in the human brain. NNs consist of computational nodes that are interconnected via links with adjustable weights. The nodes are classified either into Input, Hidden, or Output layers and the weights between them are adjusted during the training of the Neural Network. Despite its sufficient classification performance on new data and its ability to handle noisy data, Neural Network has several drawbacks. One of which is the need for extensive amounts of data for training, which by itself is a slow and computationally expensive process. Besides, the classification accuracy is also highly dependent on the model parameters selected by the data analyst. Another problem with Neural Networks is their lack of transparency. Even though it is possible to get an insight into the way the NN's neurons communicate with each other, it is unknown what representations they learn from the data. This makes Neural Network knowledge hardly transferable to different domains.

**Decision Tree**
Decision Tree classifiers are top-down tree-like structures that represent decisions and their possible consequences, including chances that an event occurs, resource costs, and utility [16]. The key process of constructing a decision tree is the Attribute Selection Method that selects attributes that become nodes in the tree and split the given records into class labels in the most optimal way [15]. When the number of attributes increases this task becomes more complex and overfitting can occur. In such cases, a method called pruning is applied to filter out the least important attributes while still keeping the overall structure of the data. Tree-based classifiers are considered one of the best supervised learning methods. They can solve both classification and regression problems and are often also referred to as "Classification and Regression Trees". Unlike other classification methods, the structure of decision trees is easy to visualize and understand, which is the major advantage of this classifier together with its high accuracy.

**Support vector machine**
Support Vector Machine is a classifier that can be used to solve two-classes problems. It aims to find out a line or hyperplane (in multidimensional space) that separates two classes while trying to maximize the margin between both. Two support vectors surround each hyperplane and calculate the margins as the goal is to find the margin that is equidistant as far as possible from both classes. Generally, SVM has the best classification accuracy, but it is not always the best classification algorithm as it does not fit every dataset. Moreover, its training step is extremely slow and requires a lot of computational power.

**Logistic Regression**
Logistic Regression is a classification algorithm which is used to categorize data into two distinct classes. Broadly, LR fits a sigmoid function (2) which returns a probability that a data sample belongs to one of two classes based on its location with respect to the sigmoid line. In fact, training the model means selecting the sigmoid function that best fits the data. In order to choose the optimal line for a given dataset, logistic regression uses maximum likelihood. Different candidate lines are generated and their likelihoods are compared. The aim is to select the line with the maximum likelihood. Logistic Regression is a simplistic model and can be easily outperformed by more complex ones, but it is easy to implement, which makes it a great baseline to measure the performance of other more complex algorithms.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

**Gradient Boosted Decision Trees**
Gradient Boosted Decision Trees (GBDT) is an ensemble learner which combines individuals decision trees to build a more robust and accurate model. Boosting means combining a learning algorithm in series to achieve a strong learner from many sequentially connected weak learners - the decision trees. In each round of training, the predictions of the current decision tree are compared to the correct outcome. A loss function is used to detect a residual - the difference between the true value and the predicted value. Each newly added tree fits the residuals from the current model as the goal of each iteration step is to reduce the classification error made by the current model (each time a new tree is added the residuals get smaller; at the final step the value of the residual should be close to 0). Two crucial hyperparameters for gradient boosting decision trees are the learning rate and the number of estimators. Each new tree modifies the overall model and the magnitude of the modification is controlled by the learning rate. Lower learning rate means that the model becomes more robust and generalized. However, the lower the learning rate, the longer it takes to train the model. There is a correlation between the learning rate and the number of estimators. If the learning rate is low, more trees (estimators) are needed to train the model. However, increasing the number of trees can cause overfitting. This makes GBDT hard to tune model, but once well tuned, its performance is outstanding.

## 2.2 Pediatric Intensive Care Database

The PIC (Paediatric Intensive Care) database is a large, paediatric-specific database that contains information about patients admitted to critical care units at the Children's Hospital of Zhejiang University School of Medicine. The main characteristics of PIC are shown in Table 1.

| | |
|---|---|
| Language | English-Chinese bilingual |
| Data source | 1900-bed children's hospital |
| Category of critical care | CICU, SICU, PICU, NICU, GICU;<br>Total 119 paediatric critical care beds |
| Number of records | 10,000+ |
| Patient age, median (Q1–Q3) | 0.8 years (0.1–3.5) |
| Vital signs | Daily nurse recorded &<br>surgery monitor generated (5 minute) |
| Laboratory data | Yes |
| Clinician notes | Extracted symptoms from notes |
| Diagnoses codes | ICD-10 Codes |
| Mortality | Only death recorded in the hospital |

Table 1: PIC's characteristics [17]

PIC can be considered as a pediatric-specific extension of the MIMIC-III database created by the Massachusetts Institute for Technology (MIT) with the goal of providing a real clinical database to support clinical research [18]. MIMIC contains health-related data associated with over forty thousand patients who stayed in critical care units of an Israeli hospital. Over the years it has supported a diverse range of analytic studies spanning epidemiology, clinical decision-rule improvement, and electronic tool development. The success of the MIMIC database has inspired the development of PIC, which upholds the goal of improving the quality of intensive care for children. Both, MIMIC and PIC, databases are freely accessible under the condition that the user completes a training course on research with human subjects and signs a data use agreement mandating the responsible handling of the data.

**Data records** PIC comprises of de-identified health-related data associated with over ten thousand pediatric patients who were admitted to critical care units of the Children's Hospital, Zhejiang University School of Medicine between 2010 and 2019. This children's hospital is the largest comprehensive paediatric medical centre in Zhejiang Province and the Chinese National Clinical Research Centre of Child Health. It has 119 critical care beds in 5 intensive care units: general ICU, paediatric ICU (PICU), surgical ICU (SICU), cardiac ICU (CICU) and neonatal ICU (NICU) [17]. Table 3 shows details (such as number of patients, age, gender, mortality) of the PIC patient population per ICU. PIC database data have been obtained from several information systems from the hospital as Hospital electronic medical record system, Laboratory information system, Nursing information system, and more. Therefore, the data available in the PIC database includes information on demographic data of the patients, laboratory measurements, charted observations during a patient's stay and vital signs during operation. This information is contained in 16 tables that are linked by unique identifiers. All the tables are distributed as a collection of comma-separated value (CSV) files that can be loaded into many relational database systems.

| Table Name | Covarage Rate | Covarage Group |
|---|---|---|
| CHARTEVENTS | 79.7% | high |
| DIAGNOSES_ICD | 99.9% | excellent |
| EMR_SYMPTOMS | 6.7% | low |
| INPUTEVENTS | 41.1% | medium |
| LABEVENTS | 93.9% | excellent |
| MICROBIOLOGYEVENTS | 89.7% | excellent |
| OR_EXAM_REPORTS | 91.9% | excellent |
| OUTPUTEVENTS | 11.7% | low |
| PRESCRIPTIONS | 51.5% | medium |
| SURGERY_VITAL_SIGNS | 47.7% | medium |

Table 2: Amount of complete data per table [17]

There are three core tables to describe the patients (PATIENTS), admissions (ADMISSIONS) and ICU stays (ICUSTAYS). The primary key of the three core tables was used to index all other clinical data. SUBJECT_ID in the PATIENTS table is a unique identifier that specifies an individual patient, HADM_ID in the ADMISSIONS table is the encounter number that uniquely identifies a particular hospitalization for patients who might have been admitted multiple times, and ICUSTAY_ID in the ICUSTAYS table refers to a unique admission to an intensive care unit. Each SUBJECT_ID has one or more related HADM_IDs, and each HADM_ID can have one or more related ICUSTAY_ID. Table 15 provides summary descriptions of the data tables. Because different clinical information systems were implemented at different times, some data was not available from the beginning of the data collection. Therefore the completeness of the different data tables varies. Table 2 shows the completeness level of PIC tables.

| Critical care unit | CICU | GICU | NICU | PICU | SICU | Total |
|---|---|---|---|---|---|---|
| Patients, no. (%) | 2583<br>20.10% | 2642<br>20.50% | 3137<br>24.40% | 1953<br>15.20% | 2566<br>19.90% | 12881<br>100% |
| Admissions, no. (%) | 2638<br>19.60% | 2725<br>20.30% | 3205<br>23.80% | 2084<br>15.50% | 2797<br>20.80% | 13449<br>100% |
| ICU stay, no. (%) | 2803<br>20.10% | 2788<br>20.00% | 3282<br>23.50% | 2166<br>15.50% | 2902<br>20.80% | 13941<br>100% |
| Age, years, mean (Q1–Q3) | 2.5<br>(0.5–3.1) | 3.8<br>(0.4–6.3) | 14.5 days<br>(0.0–18.0) | 3.6<br>(0.5–5.7) | 3.2<br>(0.4–4.9) | 2.5<br>(0.1–3.3) |
| Gender, male, % of unit stays | 1391<br>49.6% | 1712<br>61.4% | 1984<br>60.5% | 1250<br>57.7% | 1680<br>57.9% | 8017<br>57.5% |
| ICU LOS, mean days (Q1–Q3) | 3.9<br>(0.9–4.0) | 7.3<br>(0.9–8.9) | 21.6<br>(2.5–32.8) | 9.7<br>(2.0–11.1) | 2.3<br>(0.8–1.6) | 9.3<br>(0.9–9.2) |
| HLOS, mean days (Q1–Q3) | 16.6<br>(9.0–19.2) | 12.8<br>(3.8–16.2) | 27.1<br>(9.6–37.8) | 14.5<br>(4.6–16.7) | 14.7<br>(7.0–18.7) | 17.6<br>(7.0–21.0) |
| ICU mortality, % | 48<br>1.70% | 414<br>14.8% | 236<br>7.20% | 200<br>9.20% | 57<br>2.00% | 955<br>6.90% |
| Hospital mortality, % | 53<br>2.00% | 417<br>15.30% | 239<br>7.50% | 205<br>9.80% | 57<br>2.00% | 971<br>7.20% |

Table 3: Details of the PIC patient population by Intensive Care Unit on hospital admission [17]

## 2.3 Limitations of Healthcare Data

First and foremost, the sharing of real clinical data for carrying out studies and research is strictly restricted. Medical data is primarily generated through providing patient care which consequently makes it sensitive to ethical, privacy and legal issues. The legal considerations associated with the use of medical data are one of the reasons that uphold the sharing of hospital data for research purposes. The process of data mining may reveal previously unknown medical errors, which leads to lawsuits against health providers. Therefore, the confidentiality preserving strategies that comply with regulations regarding human subject research must be followed to ensure the anonymity of the patient [19]. Even when available, healthcare data have limitations due to its nature. It may contain missing, corrupted, inconsistent, or non-standardized data values [20]. Patients diagnosed with the same disease do not always undergo similar medical treatment because of difference in age, symptoms, complications and many other factors. This inevitably results in different data being generated [21]. As a result, the data might often contain inconsistent medical vocabulary, which is a serious hurdle to data mining [22]. Not only this, but healthcare data often have highly skewed class distribution, also referred to as imbalanced data. Preprocessing of the data can minimise the effect of those limitations. There are various ways to deal with missing and/or imbalanced data. The most common are described in the following sections of this report.

## 2.4 Missing Data

Because of its nature, medical data is prone to missing values. Missing data can significantly decrease the quality of the data and the performance of the predictive models utilized in the developing of medical guidelines [23]. There could be different reasons for the data absence and before jumping to the methods of data imputation, we have to understand the reason why data goes missing. In general, there are three types of missing data according to the missing data mechanisms [24]:

- **Missing Completely at Random (MCAR):** occurs when the absence of a value is not caused by or dependent on any other values, observed or missing.
- **Missing at Random (MAR):** occurs when the absence of a value is dependent on other observed features in the dataset.
- **Missing Not at Random (MNAR):** occurs when the probability of a missing value depends on the variable itself. (This missing data type is the most problematic one in terms of both finding it and dealing with it.)

There are two standard approaches that are widely used to deal with missing data. One is to delete the variables that have missing values and the other is to impute values for all missing data [25]. Since missing data is ubiquitous, a correct approach must be found to avoid loss of information. If the data is MCAR, the missing values can be discarded upon their occurrence, however, with MAR and NMAR data this might introduce bias to the model. Moreover, deleting MCAR values is also not very desirable as it reduces the size of the dataset [26]. The amount of missing values for a variable or the amount of missing values a unit has is also important to take into account when choosing a suitable approach.

**Data imputation** is a method that fills in missing values with some plausible values. This is done in the data preprocessing phase meaning that the missing data treatment is independent of the learning algorithm used and the user can select the most suitable imputation technique for each situation [27]. There are various methods for missing value imputation, such as:

- **KNN imputation**
  The idea behind k-nearest neighbour imputation is that a missing value can be approximated by the values closest to it. As the name suggests, a missing value is filled with the average of its k-nearest neighbours found in the training dataset.
- **Multivariate Imputation By Chained Equations (MICE)**
  Multivariate Imputation operates under the assumption that the data is missing at random and that it is possible to make an educated guess about its true value by looking at other observed values. MICE imputes missing values by fitting a regression model to predict the missing value by using the observed variables.

Dealing with missing values also depends on the type of data. If the missing values are numerical data they can be mean imputed, while if the values are categorical they can be imputed with a category randomly drawn from its distribution, where these categories are then 1-hot encoded, as done by [8].

## 2.5 Imbalanced Data

Imbalanced data, also known as skewed data, is characterized by a significantly unequal distribution of the minority and majority classes. This is a problem since most of the supervised learning algorithms work under the assumption that the classes in a dataset are distributed evenly, which is not always the case. As a result, the classification models get bias towards the majority class whereas the minority class is often the class that researchers are the most interested in.

Generally, class imbalance may also exist in datasets with multi-classes, but as this study is concerned with a binary classification problem, the focus is only on two-class imbalanced learning problems. There are various ways to deal with the a two-class imbalance problem; some of the most popular methods are described below.

### 2.5.1 Random Oversampling and Undersampling

Random resampling methods deal with imbalanced data by adding or removing examples form the training dataset until an even class distribution is achieved. They are referred to as naive techniques because when performed they assume nothing of the data. As the names suggest, oversampling methods duplicate examples in the minority class, whereas undersampling methods delete or merge examples in the majority class. Both techniques can achieve good results when used alone, however, it can be more effective to use them together.

Because the random oversampling method duplicates examples of the minority class, it may increase the likelihood of overfitting, especially for higher oversampling rates [28]. A limitation of the random undersampling is that it may delete examples from the majority class that are important or even critical to fitting a robust decision boundary. This is problematic as it results in a loss in classification performance.

### 2.5.2 The Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) uses the existing dataset to generate new synthetic data that represents the minority class. For a given minority class example ($x_i$), a new observation is generated by interpolating between one of its k-nearest neighbours, $x_j$. See Eq.3,

$$x_{new} = x_i + \alpha\,(x_j - x_i) \tag{3}$$

where $\alpha$ is a random number in the range [0,1]. This interpolation will create a new sample on the line between $x_i$ and $x_j$.

### 2.5.3 ADASYN: Adaptive synthetic sampling

ADASYN is another oversampling approach where synthetic examples are generated in order to deal with class imbalance. The difference with SMOTE is that ADASYN generates a greater amount of synthetic data based on minority class samples that are harder to learn, compared to those minority samples that are easier to learn. In particular, an observation from the minority class is "hard to learn" if there are many examples from the majority class with features similar to that observation. The key idea of ADASYN is to used density

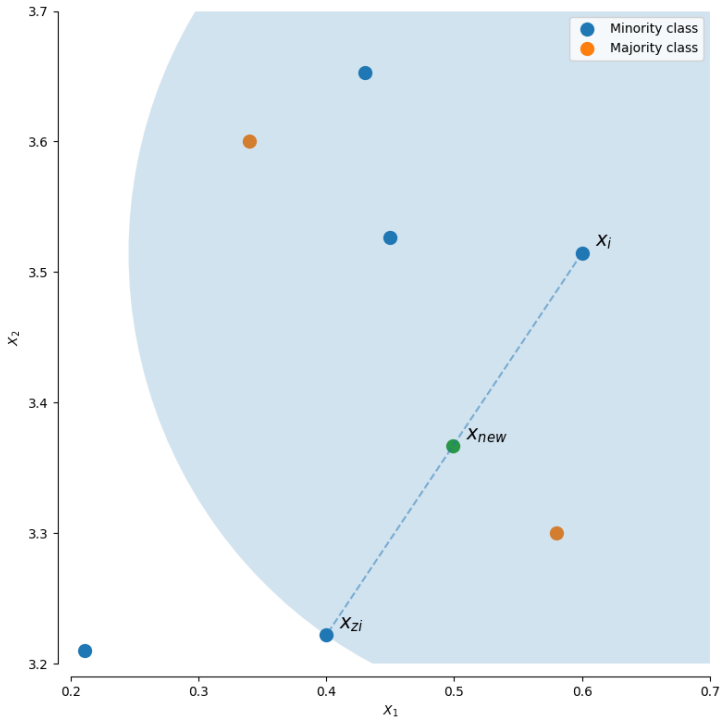



Figure 1: Generating synthetic samples with SMOTE

distribution as a criterion to decide the number of synthetic examples to be generated based on each minority class example [29].

### 2.5.4 Adjusting Class Weights/Cost-sensitive learning

Another approach to deal with imbalanced data is to adjust the class weights during the learning. The idea is to make the classifier aware of the imbalanced data by introducing a weight for each class into the cost function. Intuitively, the weights for the minority class are higher so that the end result is a classifier which can learn equally from all classes. However, cost-sensitive approaches have the downside of not knowing the actual cost of misclassification and the search for or the generation of an effective cost lead to overhead [30].

## 2.6 Evaluation Metrics

The problem of hospital readmission prediction can be acknowledged as a binary classification problem. A binary classifier predicts all instances of a test dataset as either belonging to positive or negative class. In this case, in the positive class are the patients who have returned to the hospital within 30-days after their discharge. In the negative class are the patients who have not returned to the hospital in 30-days. There are four possible classification outcomes - true positive, true negative, false positive and false negative:

- **True Positives (TP):** Outcomes where the model correctly predicts the positive class.
- **True Negatives (TN):** Outcomes where the model correctly predicts the negative class.
- **False Positives (FP):** Outcomes where the model incorrectly predicts the positive class (originally samples belong to the negative class).
- **False Negatives (FN):** Outcomes where the model incorrectly predicts the negative class (originally samples belong to the positive class).

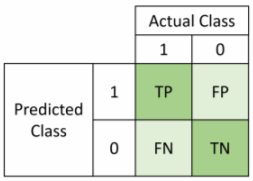



Figure 2: Confusion matrix

By counting of the number of the four outcomes of a binary classifier one can form a confusion matrix (Fig.2). Various informative performance measurements can be derived from a confusion matrix:

**Specificity or True Negative Rate (TNR)** evaluates the proportion of actual negatives that are correctly predicted

$$Specificity = \frac{TN}{FP + TN} \tag{4}$$

**Recall or Sensitivity or True Positive Rate (TPR)** evaluates the proportion of actual positives that are correctly predicted

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

**False Positive Rate (FPR)** evaluates the proportion of actual negatives that are incorrectly predicted

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity \tag{6}$$

**Precision or Positive Predictive Value (PPV)** evaluates the proportion of actual positives that are predicted as positives

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

**Negative Predictive Value (NPV)** evaluates the proportion of actual negatives that are predicted as negatives

$$NPV = \frac{TN}{TN + FN} \tag{8}$$

**F1-score** evaluates the balance between precision and recall

$$F1\,score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{9}$$

**Accuracy** evaluates the proportion of all correctly predicted samples regardless of their class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

**Receiver Operating Characteristic (ROC)** is a probability curve where TPR is plotted against FPR. Analyzing this curve gives a better understanding of the trade-off between specificity and sensitivity. Area Under the ROC Curve (AUROC) represents the capability of a model to distinguish between classes. AUC between 0.7 and 0.8 is considered as acceptable discrimination, AUC between 0.8 and 0.9 is considered as excellent, AUC above 0.9 is considered as outstanding.

**Hosmer–Lemeshow test** is a goodness of fit test that is widely used for evaluating logistic regression models. HL test statistic is given by:

$$\begin{aligned} HL &= \sum_{g=1}^{G} \left( \frac{(O_{1g} - E_{1g})^2}{E_{1g}} + \frac{(O_{0g} - E_{0g})^2}{E_{0g}} \right) \\ &= \sum_{g=1}^{G} \left( \frac{(O_{1g} - E_{1g})^2}{N_g \pi_g} + \frac{(N_g - O_{1g} - (N_g - E_{1g}))^2}{N_g (1 - \pi_g)} \right) \\ &= \sum_{g=1}^{G} \frac{(O_{1g} - E_{1g})^2}{N_g \pi_g (1 - \pi_g)} \end{aligned} \tag{11}$$

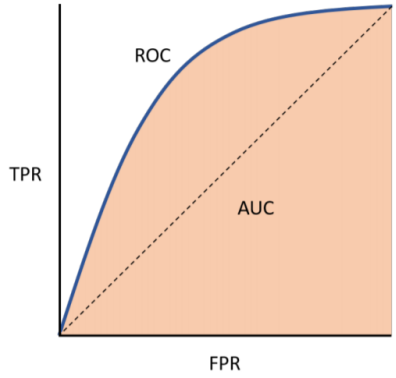



Figure 3: AUC-ROC plot

where $O_g$ denotes the observed events, $E_g$ denotes the expected events and $N_g$ denotes the number of observations for the $g$th group, whereas $G$ signifies the number of groups. The test statistic follows a Chi-squared distribution with (G-2) degrees of freedom. A large value of Chi-squared (with small p-value $< 0.05$) indicates that there is a lack of fit in the model. Small Chi-squared values (with larger p-value closer to 1) indicate a good logistic regression model fit.

# 3 Related Work

The objective of the literature review is to identify which machine learning methods are used for readmission prediction and to assess which of these models have better performance, expressed in AUROC. The literature was retrieved from Google Scholar search engine because of its extensive coverage of scientific publications. In order to collect relevant research papers, several keywords were used, namely: *predicting (unplanned) readmission, predict ICU readmission, predictive modelling for readmission, pediatric hospital readmission, neonates readmission, neonates readmission to ICU, etc.* To determine if a paper is suitable to be included in the literature review, a few criteria have been established. First of all, the search was limited to only publications written in English. Secondly, papers should have evaluated the ability of some predictive models in the task of readmission prediction. Papers that do not include any machine learning techniques are excluded. Thirdly, the studies were not limited by a diagnosis within their medical population. And lastly, there was no strict time frame, however, the intention was to look for recent scientific papers, considering that they have investigated top-notch data mining methods. Therefore, the reviewed literature dates from the year 2015 and on.

The literature review is presented in the upcoming sections of this chapter. First, an overview of the methods used in recent research on readmission prediction in the case of adults patients is presented. Then literature with a focus on pediatric patients is reviewed, followed by a conclusion on the findings from the related work.

## 3.1 Readmission prediction of adult patients

McWilliams et al. [3] aimed to classify the patients that appear physiologically fit to be safely discharged from an Intensive Care Unit (ICU) by evaluating the risk of readmission. They used both Random Forest (RF) and a Logistic Classifier (LC) to compare them upon the performance of a nurse-led discharge (NLD) criteria for a safe discharge proposed by Knight [31]. The NLD criteria consist of 15 constraints on various routinely collected laboratory results and vital signs of patients in high dependency units (Appx. F). Knight states that if a patient meets all the constraints for at least 4 hours, then they may be safely discharged by a nurse. McWilliams et al. studied two historical cohorts - GICU at The Bristol Royal Infirmary and MIMIC-III to investigate whether the data mining algorithms are better estimators for safe discharge than the NLD criteria. By using two cohorts they increased the volume of the data available for training and were able to study the generalization of their results between different patient populations. Both Random Forest and Logistic Classifier showed improved performance compared to the NLD criteria (AUROC = 0.82) for each of the datasets. However, the RF (AUROC = 0.89) outperformed the LC (AUROC = 0.87) on the MIMIC-III dataset, while on the GICU cohort, both RF (AUROC = 0.87) and LC (AUROC = 0.88) produced similar results. Those results prove that depending on the characteristics of the dataset, different models will perform better.

Rojas et al. [32] investigated the prediction of readmission to an ICU. They used a Gradient-Boosted Machine (GBM) model trained and validated on an internal dataset from an academic hospital in the United States. Moreover, external validation of the model has been performed on the MIMIC-III dataset. The results were compared with the Stability and Workload Index for Transfer (SWIFT) score and the Modified Early Warning Score (MEWS). The SWIFT score was developed by Gajic et al. [33] to predict unplanned readmission; the MEWS was proposed by Stenhouse et al. [34] to identify patients at risk of clinical deterioration and in need of a higher level of care. The GBM showed significantly better performance (AUROC = 0.76) than SWIFT (AUROC = 0.65) and MEWS (AUROC = 0.58). Those results were yielded from the internal dataset that Rojas et al. used in their study, however, they reported similar accuracy improvements over SWIFT and MEWS within the MIMIC-III cohort.

Because of its promising results, a variation of Gradient-Boosted Machine models can be seen more often in recent studies related to the prediction of safe discharge and/or readmission. For instance, Pakbin et al. [8] employed Gradient-Boosted model (XGBoost) and Logistic Regression model trained on MIMIC-III dataset to predict the risk of adult patients being readmitted to ICU in short-term – within 72 hours after discharge and longer-term – within 30 days after discharge. For the XGBoost, the mean value of AUROC for 72h and 30-day ICU readmission were 0.76 and 0.75, respectively, whereas the Logistic Regression achieved slightly lower AUROC values - 0.74 and 0.73. The results show no significant difference between the performance of XGBoost and Logistic Regression. In a very similar manner, Nguyen, Paris and Parrot [4] also used XGBoost method to predict ICU readmission at different time points (3, 7, 15, 30-days) after discharge. The model was trained and evaluated on OMOP-CDM version of the freely accessible MIMIC-III dataset. The results were evaluated with AUROC and compared to the results of two other studies. For the 3-days, 7-days, 15-days and 30-days readmission cases, the AUROC mean values were of 0.802, 0.809, 0.803, and 0.795, respectively. The researchers reported that their results outperformed compatible existing solutions.

Several studies compared the performance of different data mining techniques on the task

of predicting patient readmission ([35], [7], [9]). A common observation was that there exists some trade-off between accuracy and model interpretation. Models that yield better accuracy are often not easily interpreted and the other way around, models that are easy to interpret typically have worse accuracy as discussed by Caruana et al. [35]. Futoma et al. [7] compared variations of Logistic Regression (Maximum-likelihood LR, Penalized LR and multi-step LR), Random Forest, Support Vector Machine, and Stochastic Gradient Descent to conclude that non of the models shows significant superiority in performance. Furthermore, the authors compared Deep Neural Networks with the Penalised Logistic Regression models for predicting early hospital readmissions across various disease cohorts. Their results show that the Deep Neural Networks (AUC mean: 0.735) outperformed the Penalised Logistic Regressions (AUC mean: 0.715) with statistical significance, though training of the DNN models was complex due to the large number of parameters and lack of interpretability. Ben-Assuli and Padman [9], compared five classifiers: Logistic Regression (LR), Boosted Decision Trees (BDTs), Support Vector Machine (SVM), Bayes Point Machine (BPM), and Two-Class Neural Network (TCNN). They applied these methods to predict 30-day readmission risk for patients who stayed in the emergency department. The obtained results showed that each of the five methods exhibited good performance for predicting the readmission within 30 days, however, the Boosted Decision Trees had the highest AUROC score - 0.925, followed by the LR (0.912), BPM (0.912), TCNN (0.878), and SVM (0.846).

## 3.2 Readmission prediction of pediatric patients

A great number of the reviewed papers have excluded patients below 16 or 18 years old from their research – [4], [3], [8], [36], [37]. This decision is motivated by the assumption that factors contributing to readmission prediction might be different within the younger age group. Other scientific works, focused on the pediatric population, also confirm that "the underlying illness in children and reasons that children are critically ill is quite different in comparison with adults" [38]. Moreover, the identification of a serious illness in children is more difficult as children's immunity can compensate well and show late but sudden deterioration. Therefore, the analysis of the medical conditions of adults cannot be extended directly into young patients [39].

The population of pediatric patients has received little attention in the task of hospital readmission prediction. Studies often focus on identifying factors associated with readmission for pediatric patients [40], [38], [41], [42], rather than investigating predictive modelling approaches. Only recently studies have worked on predicting pediatric patients readmission to the emergency department [43], [44], [45]. Artetxe et al. [44] have tested a novel ensemble classifier architecture, Anticipative Hybrid Extreme Rotation Forest (AHERF), over balanced samples of data gathered at emergency services. The design of AHERF is motivated by the no-free lunch theorems, which state that there is no optimal machine learning approach for all instances of classification and regression problems. Therefore, AHERF estimates which kinds of classification architectures are better suited for the problem at hand and the best-fitted classifiers are used to build the model. Artetxe et al. reported that their AHERF (78.57%) shows superior performance, expressed in accuracy, when compared over Support Vector Machine (59%) and Random Forest (72.72%), however, these results are not definitive because they obviate the strong imbalance of the dataset.

Bergese et al. [43] compared the effectiveness of Artificial Neural Network (ANN) and Decision Trees (DT) on the task of predicting pediatric return visits to the emergency

department within 120 hours after discharge. The models achieved high accuracy, 81% for DT and 91.3% for ANN, but also performed superbly on other measures such as sensitivity and specificity. When comparing both DT and ANN, Bergese et al. concluded that DT is a better model for predicting readmissions in a pediatric setting as it outperforms ANN in regard to the sensitivity measure – 79.8% versus 6.9%, respectively.

Wolff et al. [45] reported the following classification results achieved with Support Vector Machines (SVM) (AUROC = 0.6), Multilayer Perceptron (MLP) (AUROC = 0.64), and Naive Bayes (NB) (AUROC = 0.65) approaches after data preprocessing for correction of class imbalance. They carried on repeated cross-validation with decreasing number of folders to assess performance and sensitivity to the effect of class imbalance. The data was class-balanced using SMOTE up-sampling procedure with five-nearest neighbour on the minority class samples until the number of samples in each class was the same. Wolff et al. have reported large and significant increase in the recall for the positive class due to the class imbalance correction – recall [%] of SVM: 0.95 before SMOTE, 45.63 after SMOTE; MLP: 27.60 before SMOTE, 96.29 after SMOTE; NB: 14.81 before SMOTE, 70.8 after SMOTE.

Jovanovic et al. [46] focused not only on developing an accurate predictive model, but also an interpretable one. Their approach involves the integration of domain knowledge (in the form of ICD-9-CM[1] taxonomy) and a data-driven, sparse predictive algorithm – Tree-Lasso Logistic Regression (AUROC = 0.779). This approach was compared with traditional Lasso Logistic Regression (AUROC = 0.783). They reported that the Tree-Lasso Logistic Regression model with the ICD-hierarchy is more interpretable than the traditional Lasso model, without a significant loss in performance.

## 3.3 Conclusion

Various classification models have already been tested in predicting hospital readmission for adult patients. The approaches used in the reviewed studies cannot be directly compared due to the particular characteristics of each research. However, according to the results reported by studies where different models have been compared under similar conditions, generally, more complex models outperform traditional ones [4], [7]. For example, Futoma et al. [7] reported that Deep Neural Networks (AUC mean: 0.735) outperform Penalised Logistic Regression (AUC mean: 0.715) with statistical significance. Ben-Assuli and Padman [9] also compared multiple classifiers, where Gradient Boosted Decision Tree yields the highest AUROC score - 0.925. Generally, the Gradient Boosted Decision Tree models have proven to be successful in predicting readmission of adult patients (AUROC = 0.76 [32]; AUROC = 0.802 [4]; AUROC = 0.76 [8]). As a traditional statistical approach, Logistic Regression has also been widely used and has shown good results (AUROC = 0.912 [9]; AUROC = 0.74 [8]). Moreover, Logistic Regression is known for its straightforward interpretability [47], which, as already discussed, is of great importance in medical applications. On the contrary, more complex models, such as Neural Networks, are hard to interpret but often outperform simpler ones. Therefore, both complex and simple algorithms should be considered when choosing models of interest.

Despite the successful application of machine learning models in predicting readmission of adult patients, the possibilities in the neonatal domain have been left unexplored. For example, as already concluded, the Gradient Boosted Models have proven to be successful in predicting hospital readmission for adult patients, but they have not been previously tested for a pediatric, let alone neonatal, patient population. Therefore, this work can contribute

[1]International Classification of Diseases 9th—revision Clinical Modification

by conducting a comparative study to assess the effectiveness of GBM, among NN and LR, in the domain of neonatal readmission prediction.

# 4 Methodology

In this section, the methodology used to address the research objectives of this study is presented. The diagram in Figure 4 provides a general understanding of the stages in the methodology.

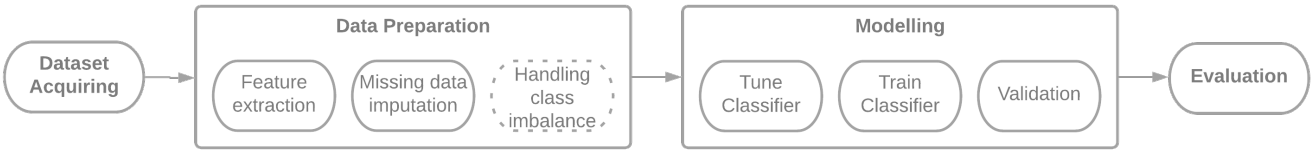



Figure 4: Methodology Diagram

## 4.1 Dataset acquiring

The first main stage of the methodology is to get access to the PIC dataset. The data is freely accessible upon completing a training course on research with human subjects and signing a data use agreement mandating the responsible handling of the data.

## 4.2 Data preparation

Several preparation steps are needed before the data is suitable for use in building a predictive model. Details on the data preparation are described in Chapter 5.

## 4.3 Modelling

This phase consists of tuning the machine learning algorithms that were selected in advance based on literature, setting up experiments and the training and testing of the models. Chapter 6 explains the steps taken during this phase.

## 4.4 Results analysis

The performance of the algorithms and the results from the experiments are analysed during this phase. AUROC score was selected as a metric to evaluate the models. The higher this metric becomes, a model can better discriminate between the two classes. Descriptions of this and related performance metrics can be found in Section 2.6. Further explanation of the evaluation of the models is available in Section 6.2.

# 5 Dataset Preparation

## 5.1 Data Extraction

The data available for this study is the Pediatric-specific Intensive Care (PIC) database (see Section 2.2). The data is stored in a relational structure, meaning that multiple tables, each with a different number of columns and rows, are linked by a key id. The large inter-related tables are separated for clarity into categories such as PATIENTS, ICUSTAYS, CHARTEVENTS, etc (see Appx. C). Before going any further, it is necessary to transform this structure into a new one, where the relevant information from each table is extracted and then combined within one data frame. As PIC is originally stored in SQL database system, PostgreSQL queries were used to adjust and obtain the relevant subsets of data from the PIC database, which were then further preprocessed in Python. The next section goes into detail about the features extraction.

## 5.2 Feature Extraction

**Age - in days** Patient's age is not a given attribute in the PIC dataset, however the date of birth for each patient is available. The feature age is therefore extracted by finding the difference between patient's date of birth and the date of admission to an ICU. The age is noted in days because the interest is on neonatal patients, which are up to 30 days old. Data rows with patients older than that are filtered out.

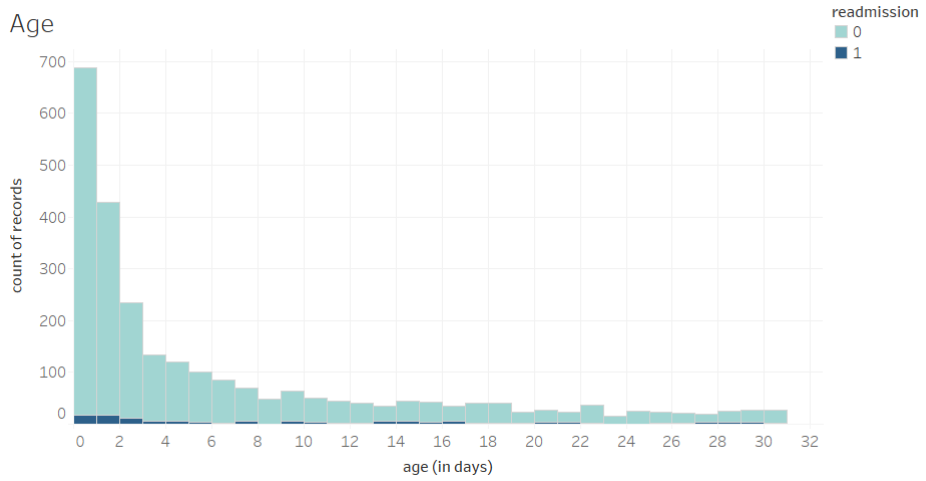



Figure 5: Age Density

**Chartevents** Chartevents are all the charted data available for a patient. The electronic chart displays patients' routine vital signs and any additional information: demographic data, input and output, and so on[2]. Chartevent measures appear as dynamic data, meaning that a patient can have multiple measures of a single event over the ICU stay. Aggregation was performed to extract the minimum, maximum and mean values of the chartevents measures of interest. Moreover, a pre-filtering of the values was necessary due to typographical errors (Table 4).

[2]PIC documents. [online] Available at: `http://pic-doc.nbscn.org/#/pmimictables/chartevents` [Accessed 3 March 2021].

| | |
|---|---|
| Respiratory rate | 10 <value <100 |
| Heart rate | 60 <value <300 |
| Temperature | 20 <value <40 |

Table 4: Filtering of the values for respiratory rate, heart rate, and temperature

Initially, the most commonly measured chart events were extracted, namely: Temperature, Respiratory rate, Heart rate, Systolic blood pressure, Diastolic blood pressure, and Urine. The reason behind extracting the most commonly measured events was to avoid features with a lot of missing values. The Weight measure was also derived as it is an important feature for the neonatal patient population and it is relevant for the preterm conditions. Out of these features, Systolic blood pressure, Diastolic blood pressure, and Urine were dropped due to a high percentage (over 30%) of missing values. For the remaining chartevents measures, the observations where all or multiple features were missing were dropped. Additionally, since the Weight also had a great amount of missing values (over 40%), a decision was made to extract two boolean features out of it - Low birth weight and Very low birth weight. Table 5 shows the value ranges for VLBW and LBW, validated by literature – [48] and the distribution of the weight in preterm newborns, shown in Figure 6.

| | |
|---|---|
| VLBW | 0.5 kg <weight_max <1.5 kg |
| LBW | 1.5 kg <weight_max <2.5 kg |

Table 5: The value ranges for VLBW (Very low birth weight) and LBW (Low birth weight)

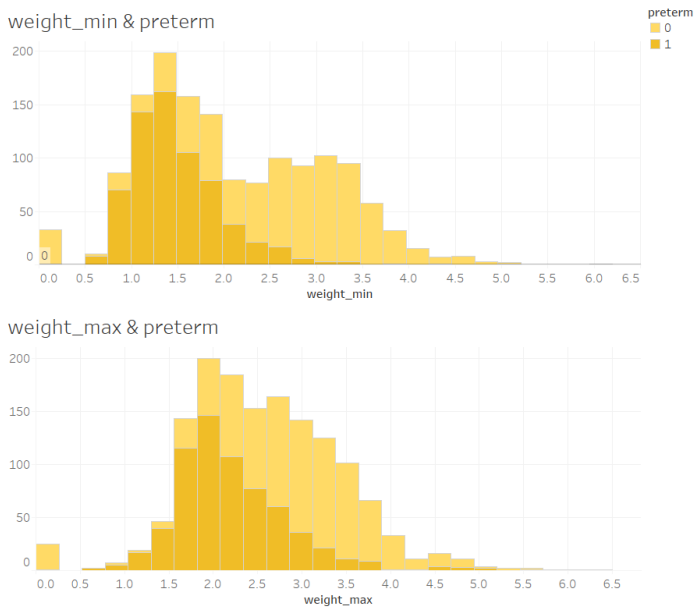



Figure 6: Distribution of the weight in preterm newborns
,

**ICD10 codes** ICD is a classification system for diagnosis coding, which clusters a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease. Due to the high number of unique ICD10 codes, they were reduced into their more general categories, extracted from the World Health Organization website[3] (see Appx. A). The ICD10 categories were further filtered and only the top 2 categories were kept. As we can see from Figure 7, the top ICD10 Categories are P00-P96 and Q00-Q99 with highest numbers of records belonging to the readmission class.

[3] `https://icd.who.int/browse10/2010/en`

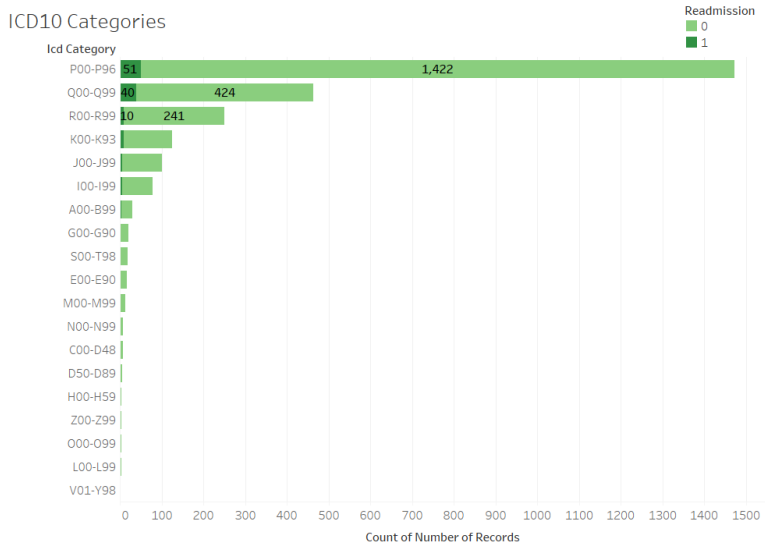



Figure 7: Top ICD10 Categories

,

**Diagnoses** The diagnoses column contains around 241 unique values. In order to enrich the dataset, a few features were extracted from the most common diagnoses (see Figure 8). Those most common diagnoses and the features extracted from them are: Preterm, Birth asphyxia, Congenital malformation of heart and Congenital condition in general. A search for the specific diagnoses therms was performed in order to extract all the instances with the given diagnosis (Appx. B).

**Dependent variable - Readmission** The dependent variable here is the readmission and therefore a readmission label column should be defined. The readmission labels quantify whether the patient was readmitted within 30 days after being discharged from the ICU (coded as 1) or not (coded as 0). To create the labels, it is needed to identify the patients who returned to the ICUs. The first step is to extract the next ICU admission time, if it exists. Once the next ICU admission time is available, the days-until-next-admission are calculated and only the ones that are within 30 days are coded as readmissions ($days\_next\_icuadm <= 30$). Records, where the patient died during their index ICU stay, are removed. However, if a death occurs during a readmission, the precedent ICU stay is labeled as a readmission.

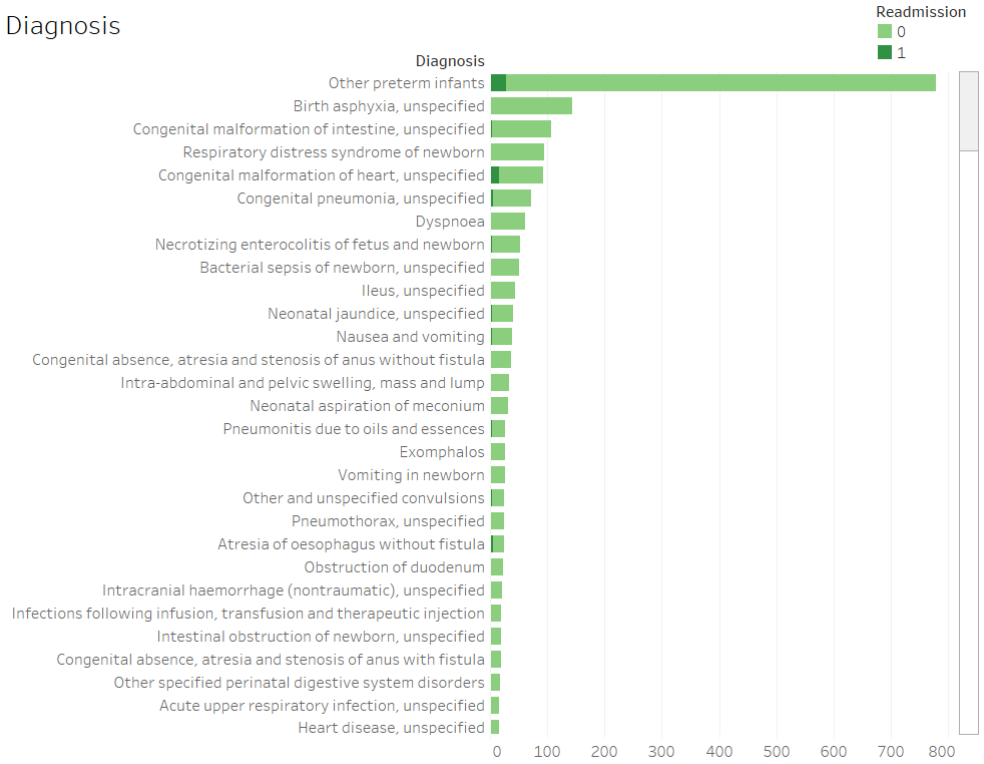
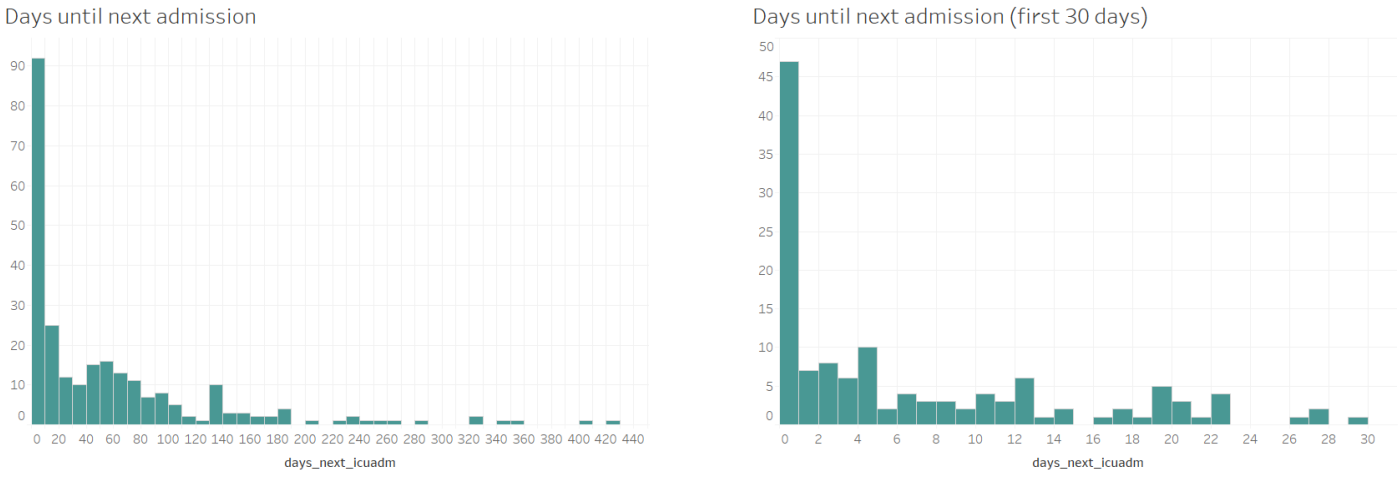



Figure 8: Diagnosis




(a) **Days until the next ICU admission** (b) **Days until the next ICU admission (30 days)**

Figure 9: A bar chart of the days until the next ICU admission

**Final data** The final data consists of 27 features, presented in Table 6, and a total of 2640 records, of which only 129 are readmissions.

| | |
|---|---|
| gender_male | categorical |
| agedays | numerical |
| han_ethnic | categorical |
| premiums_insurance | categorical |
| medical_insurance | categorical |
| generalcard_insurance | categorical |
| NICU_careunit | categorical |
| CICU_careunit | categorical |
| SICU_careunit | categorical |
| los | numerical |
| icdP00-P96 | categorical |
| icdQ00-Q99 | categorical |
| preterm | categorical |
| congenital_condition | categorical |
| birth_asphyxia | categorical |
| heart_malformation | categorical |
| VLBW | categorical |
| LBW | categorical |
| temp_mean | numerical |
| temp_max | numerical |
| temp_min | numerical |
| resp_rate_mean | numerical |
| resp_rate_max | numerical |
| resp_rate_min | numerical |
| heart_rate_mean | numerical |
| heart_rate_max | numerical |
| heart_rate_min | numerical |
| readmission | categorical |

Table 6: All selected data features

## 5.3 Data Preprocessing

The selected features are a combination of numerical and categorical features, which requires preprocessing before being fed into a classifier. All numerical features were standardized by *Z-score Normalization* so that each feature's distribution has a mean value of 0 and a standard deviation of 1. The categorical features were transformed into one-hot encoded vectors. Furthermore, there are missing values in the features extracted from the chartevents. It is unknown if the values are missing simply due to the fact that the clinical measures are not routinely performed on all patients or due to other reason. To handle the missing values, besides deletion, K-Nearest Neighbour (KNN) imputation technique as proposed by [3] was used. As the name suggests, this imputation technique utilizes the k-Nearest Neighbours method to replace the missing values. KNN as imputation method is less prone to bias than other sampling methods such as the mean imputation. Moreover, KNN is widely used because of its efficiency [49]. In this study, the number of neighboring samples to use for imputation was decided to be k = 11; moreover, the distance measure is weighed proportional to the distance between instances (rows). A few experiments with different values for k, the number of nearest neighbours, were carried out on small section of the data. Eventually, k = 11 was chosen as it scored the smallest RMSE (Root Mean Square Error) with default support vector machine model, which was chosen randomly.

Another critical issue during classification is the class imbalance of the data. There are different ways to deal with this problem. One way is to adjust the weights of each class during the learning process; another way is to preprocess the data so that the balance between the classes is restored. It is subjective, which one of these techniques is going to yield better results, therefore, in this study, both methods are tested in order to investigate weather an additional step for class balancing is necessary during the data preprocessing.

## 5.4 Data Visualization

In this section, some visualizations of the ready-to-use data are going to be presented. This is to give a better understanding of the data.

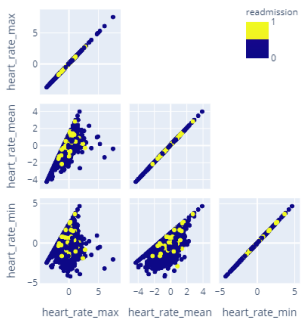



Figure 10: A scatter matrix of the *heart_rate* features

Figure 11 shows a scatter matrix of selected numerical features after data preprocessing.

Scatter plots are useful for finding and interpreting trends in the data. Each square shows the scatter plot corresponding to the two features defined by the row and column. To reduce the size of the scatter plot and make it easy to read, some of the aggregated values for the vital signs are excluded and only the mean value is included in the plot. It is expected that the min, mean, and max values of a single parameter (e.g. *heart_rate*, see Figure 10) are having a strong correlation. However, as we can see from Figure 11, the data is not clustered along an obvious line, meaning that the features have a weak correlation with each other.

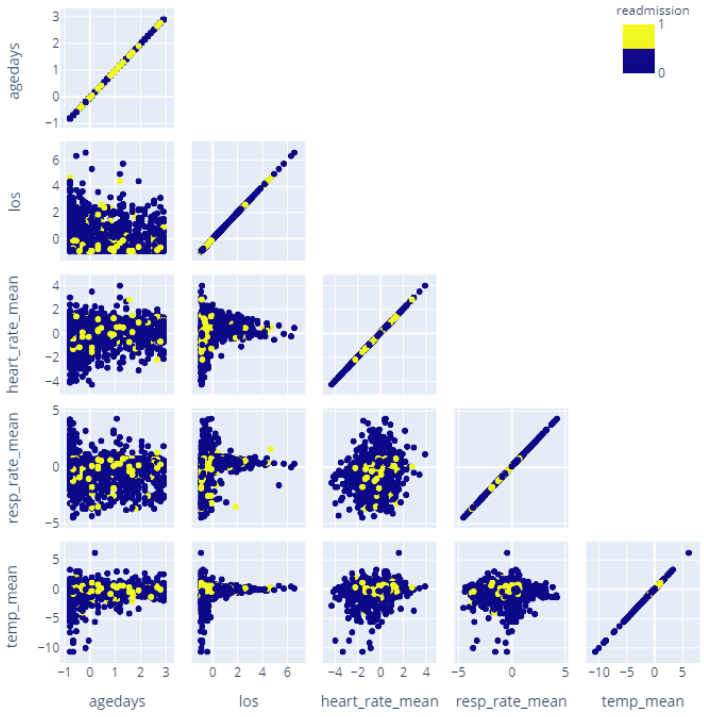



Figure 11: A scatter matrix of selected numerical features after data preprocessing

# 6 Models Design


Once the data is ready, the models' design phase can take place. Several machine learning approaches have been selected for predictive modeling in this study. Namely Logistic Regression, Gradient Boosted Decision Trees, and Neural Network. These models have been reported to have good performance (expressed in AUROC score) in the literature about readmission prediction for adult patients but have not been used in readmission prediction for neonatal patients before (see Section 3.3). The application of deep learning models is discarded as the available data is too small for this purpose. The models' implementation phase involves hyperparameters tuning, evaluation and comparison of models performance, further introduced in the follow-up subsections of this chapter. Additionally, an experiment involving a class balancing technique is conducted to see if an additional data preprocessing step is necessary. In the end, I look at the features that have the biggest impact on the predictions by the best performing model.

## 6.1 Hyperparameters Tuning

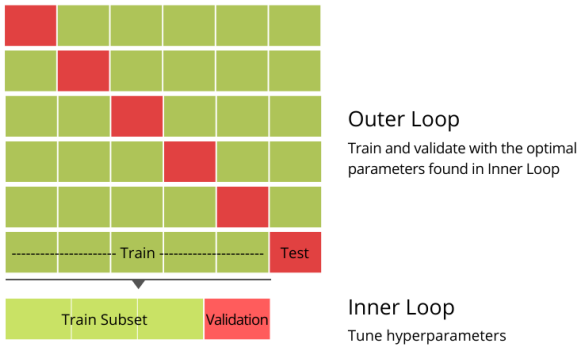



Figure 12: The illustration of the Cross-validation Grid Search used in this study

Hyperparameters are specified parameters that are tuned to control a machine learning algorithm's behaviour. These values are usually determined before the training process. The hyperparameters are crucial to the performance, speed, and quality of the machine learning models. Hence, they should be optimized. The search for the best hyperparameter values for the Logistic Regression and Gradient Boosted Trees models is done through Nested Cross-Validation and Grid Search. In K-fold Cross-Validation, the dataset is divided into $k$ equal-sized partitions, where one of the $k$ partitions is used as the test/validation set and the other $k-1$ partitions are put together to form a training set. This step is repeated $k$

times as each time a different partition is used as a test set. This significantly reduces bias as all the data is used for fitting and also reduces variance as all that data is being validated on.

Due to the unbalanced nature of the data in this study, Stratified K-fold was used to preserve the class ratio throughout the different folds. Nested Cross-Validation contains an outer loop for error estimation, where k = 6, and inner loop for parameter tuning, where k = 4 (see Figure 12). Grid Search is performed on the inner loop of the Nested Cross-Validation. Grid Search is a traditional method for hyperparameter optimization which search through a manually specified subset of the hyperparameter space of a learning algorithm. The top-performing set of parameters is selected based on the results on the validation set, with AUROC as the scoring method.

### 6.1.1 Logistic Regression

The first model considered is Logistic Regression - the most popular method in prior research that has been implemented in many readmission prediction studies. To decide on optimization, regularization and regularization strength, tuning experiments are done, as explained in the section above. See Table 7.

| GridSearchCV for LogisticRegression | |
|---|---|
| penalty | *L1, L2, none* |
| solver | *lbfgs, liblinear, saga, newton-cg* |
| class weight | *balanced* |
| C | *0.001, 0.01, 0.1, 1* |

Table 7: Lists of parameter settings to try as values for the Logistic Regression model

### 6.1.2 Gradient Boosted Decision Trees

The second model used in this study is Gradient Boosted Decision Trees. This technique has shown to be one of the best-calibrated machine learning methods for predicting hospital readmission. The optimal number of splits for each individual tree, the total number of trees, and learning rate, etc. were determined through Grid Search as explained at the beginning of this section. Note that the final lists of parameter settings for the Grid Search (Table 8) were decided after several initial experiments focused on defining the learning rate and the number of estimators, again based on AUROC value (see Appx. D).

| GridSearchCV for Gradient Boosted Decision Trees | |
|---|---|
| learning_rate | *0.01, 0.001* |
| n_estimators | *1550, 1750* |
| criterion | *'friedman_mse', 'mse'* |
| max_depth | *3, 4* |
| max_leaf_nodes | *None, 3, 4* |

Table 8: Lists of parameter settings to try as values for the GBDT model

### 6.1.3 Neural Network

Hyperparameters related to the Neural Network structure include the number of hidden layers, dropout regularisation, network weight initialisation, activation function, learning rate, number of epochs, and batch size. See Table 9. Performing a Cross-validated Grid Search over all configuration choices is too computationally expensive. Hence, instead of the Grid Search, the hyperparameters of the Neural Network model were tuned manually. Furthermore, because the data is too small, instead of implementing cross-validation, the data was split and 20% of it was dedicated for validation. Some of the components of the network are fixed, such as the activation function, the optimizer and the loss function.

| Hyperparameters of Neural Network | |
|---|---|
| # of hidden layers | *1, 2, 3* |
| neurons | *16, 32* |
| activation function | *ReLU, Sigmoid (for output)* |
| dropout | *0.3, 0.4* |
| weight for positive class | *1, 1.5, 2 divided by # of positive samples in training dataset* |
| weight for negative class | *1/# of negative samples in training dataset* |
| optimizer | *Root Mean Square Propagation* |
| loss function | *binary cross-entropy* |
| epochs | *EarlyStopping*<br>*epochs: 5000*<br>*patience: 500* |

Table 9: Lists of parameter settings to try as values for the Neural Network model

## 6.2 Evaluation Metrics

Since this is a binary classification problem, it is convenient to use the Area Under The Receiver-operating Characteristic Curve (AUROC) to compare the performance of the selected classifiers. Even though AUROC scores have been frequently used as the main evaluation metric in many of the reviewed literature, it is also important to check the actual ROC curves because even when two models have the same AUCROC values, their ROC curves can be quite different. Therefore in this study, both the AUROC scores and the curves themselves are analysed. To get more insight into what types of errors the models made, additional performance metrics are taken into account. For this study, it is desired to minimize the false negatives (FN) - incorrectly classified as not readmitted, and maximize the true positives (TP) and true negatives (TN). Therefore, the Recall and Precision values are included in the evaluation. Furthermore, to be able to track the cost of the prediction models with regards to the false negative predictions, F-score is also included in the evaluation.

## 6.3 Class Balancing

When one class is the underrepresented minority class in the dataset, as is the case in this study, there are two common preprocessing methods that can be used to restore the balance of the classes - oversampling and undersampling. As the names suggest, undersampling removes instances from the majority class until a balance between the classes is achieved.

Conversely, oversampling adds (synthetic) instances to the minority class until restoring balance. Generally, oversampling is preferable as undersampling can result in the loss of important data. Moreover, undersampling is suggested when the amount of data available is larger than ideal. Since the available data for this study is rather limited, an oversampling technique is chosen to investigate the effects that class balancing have in predicting cases of readmission. The method used in this study is ADASYN. It finds the k-nearest neighbours based on Euclidean distance for each sample from the minority class and it generates synthetic samples based on the harder to learn examples. A few experiments with different values for k, the number of nearest neighbours, were carried out on small section of the data. Eventually, k = 3 was chosen as it scored the smallest RMSE (Root Mean Square Error) with default support vector machine model, which was chosen randomly. Once the oversampling process is done, the next step is to retrain the already optimized models, but this time with the balanced data. This is done to investigate whether the class balancing technique is a necessary step in neonatal readmission prediction.

# 7 Results

## 7.1 Results from hyperparameter tuning

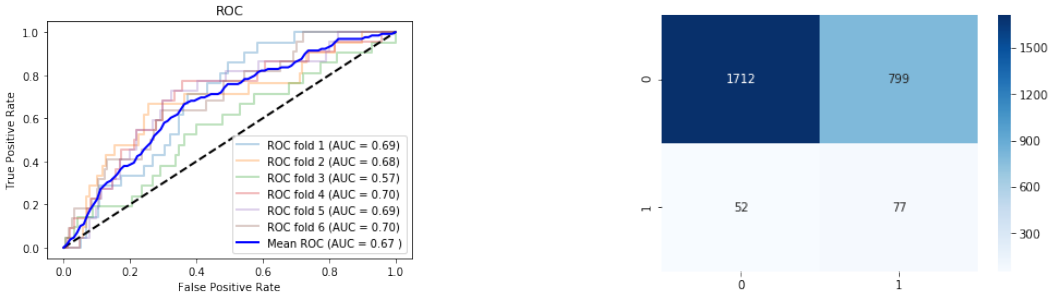
### 7.1.1 Logistic Regression

| Outer Fold | Outer Fold AUC | Best AUC from GridSearch | Best Parameters |
|---|---|---|---|
| 1 | 0.670 | 0.654 | {*'C': 0.01, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'newton-cg'*} |
| 2 | 0.685 | 0.653 | {*'C': 0.01, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'saga'*} |
| 3 | 0.566 | 0.663 | {*'C': 0.01, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'lbfgs'*} |
| 4 | 0.674 | 0.633 | {*'C': 0.1, 'class_weight': 'balanced', 'penalty': 'l1', 'solver': 'liblinear'*} |
| 5 | 0.674 | 0.650 | {*'C': 0.001, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'liblinear'*} |
| 6 | 0.650 | 0.652 | {*'C': 0.001, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'liblinear'*} |

Table 10: Grid Search results for the Logistic Regression

The results from the Grid Search for the Logistic Regression model show relatively consistent best scores for AUROC ($\sigma = 0.009$). The results for best parameters show a few differences for the regularization strength values ($C$) and the solver algorithm. When validated on the outer fold, the best AUC score is 0.685 in the second fold with best parameters being: *C: 0.01, class_weight: balanced, penalty: l2, solver: saga*. However, after further experiments with optimization algorithms, it was decided to use the liblinear solver as it yielded slightly better TP and TN values.

Hence, the Logistic Regression model used in the further experiments in this study is with the following parameters: *C: 0.01, class_weight: balanced, penalty: l2, solver: liblinear*. See results from the 6-fold stratified cross-validation in Figure 13.




(a) ROC curves from 6-fold cross-validation

(b) Confusion Matrix

Figure 13: LR with optimized hyperparameters

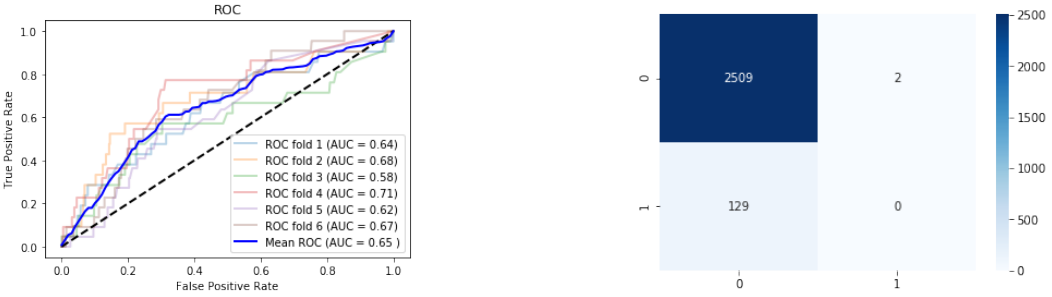
### 7.1.2 Gradient Boosted Decision Trees

| Outer Fold | Outer Fold AUC | Best AUC from GridSearch | Best Parameters |
|---|---|---|---|
| 1 | 0.635 | 0.618 | {*'criterion': 'friedman_mse', 'learning_rate': 0.001, 'max_depth': 3, max_leaf_nodes': None, 'n_estimators': 1750*} |
| 2 | 0.683 | 0.611 | {*'criterion': 'friedman_mse', 'learning_rate': 0.001, 'max_depth': 3, 'max_leaf_nodes': None, 'n_estimators': 1550*} |
| 3 | 0.483 | 0.634 | {*'criterion': 'friedman_mse', 'learning_rate': 0.01, 'max_depth': 3, 'max_leaf_nodes': 3, 'n_estimators': 1550*} |
| 4 | 0.710 | 0.593 | {*'criterion': 'mse', 'learning_rate': 0.001, 'max_depth': 3, 'max_leaf_nodes': None, 'n_estimators': 1750*} |
| 5 | 0.612 | 0.645 | {*'criterion': 'friedman_mse', 'learning_rate': 0.001, 'max_depth': 3, 'max_leaf_nodes': None, 'n_estimators': 1550*} |
| 6 | 0.673 | 0.631 | {*'criterion': 'mse', 'learning_rate': 0.001, 'max_depth': 3, 'max_leaf_nodes': None, 'n_estimators': 1750*} |

Table 11: Grid Search results for the Gradient Boosted Decision Trees

The results from the Grid Search for the GBDT are presented in Table 11. As it can be seen, the AUROC scores from the Grid Search and the outer folds fluctuate across the 6 folds. For example, in the 3rd and 4th fold, there are significant differences between the outer fold AUC scores and the Grid Search AUC scores, which indicates that the data reserved for validation in the outer folds is not a good representation of the data used while searching for the best parameters. Therefore to decide on the model parameters, the most common best parameters were selected, rather than the ones associated with the highest AUROC in the outer fold.

As a result, the GBDT model used in the further experiments in this study is with the following parameters: *criterion: friedman_mse, learning_rate: 0.001, max_depth: 3, max_leaf_nodes: None, n_estimators: 1750.* See results from the 6-fold stratified cross-validation in Figure 14.




(a) ROC curves from 6-fold cross-validation

(b) Confusion Matrix

Figure 14: GBDT with optimized hyperparameters

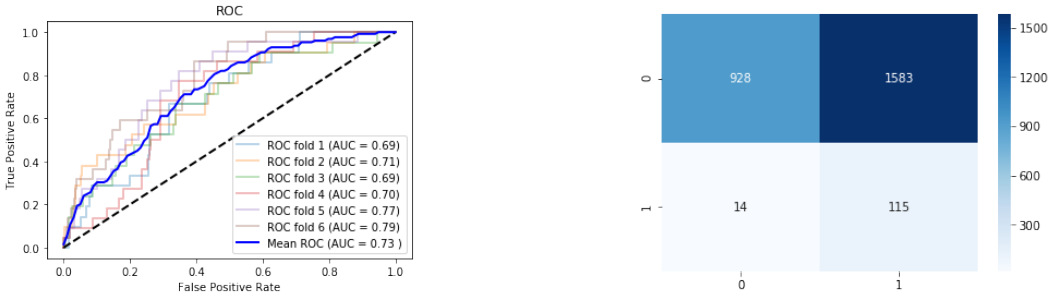
### 7.1.3 Neural Network

Unlike the tuning of the LR and GBDT models, the Neural Network model was tuned manually. A variety of architectures, ranging from one to three hidden layers of nodes with varying the sizes of layers were tested (see Appx. E). Ultimately, found that using simply one hidden layer performed better than two or three layers.

A hidden layer has perceptrons/neurons, in this case, the hidden layer consists of 16 perceptrons. Every perceptron unit takes input from the input layer, multiplies it and adds it to initially random values. Then the resulting output has to be activated by an activation function. In this case, a ReLU activation function is used on the hidden layer. The output from the hidden layer serves as an input to the last layer of the neural network - the output layer. There the outputs from the hidden layer are multiplied and added to initial random values. Then an activation function takes care of calculating the prediction. In this case, a Sigmoid activation function is used.

The learning process consists of a loss and an optimizer function. These functions define how the learning process is progressing. Essentially, the optimizer updates the model parameters - weight and bias and aims to reach the global minima where the loss function attains the least possible value. The binary cross-entropy is used as a loss function and RMSprop (Root Mean Square Propagation) is used as an optimizer. RMSProp uses an adaptive learning rate instead of treating the learning rate as a hyperparameter. The learning rate for each trainable model parameter is iteratively updated as RMSProp maintains slower learning in the vertical direction (bias) and faster learning in the horizontal direction (weight). This boosts the speed and the accuracy of the model.

Dropout is used as a regularization technique. The idea behind it is to reduce the interconnecting neurons within a neural network by dropping them out randomly. While the neurons are trained they can become co-dependent on each other, meaning that their weights can affect how the weights of other neurons get optimized. Dropping out neurons at random prevents this co-dependency between neurons. The dropping out happens by attaching Bernoulli random variables to the neuron's output. At each epoch, each neuron has a chance of being dropped that is determined by a dropout rate. It is highly impossible that the same neurons are excluded at any two training steps, meaning that a variety of different networks are trained at each step. This simulates an ensemble of neural networks, which is known to reduce overfitting, but it is highly computationally expensive. Two values for the dropout were tested - 0.3 and 0.4. At the end, dropout of 0.3 reported better results.

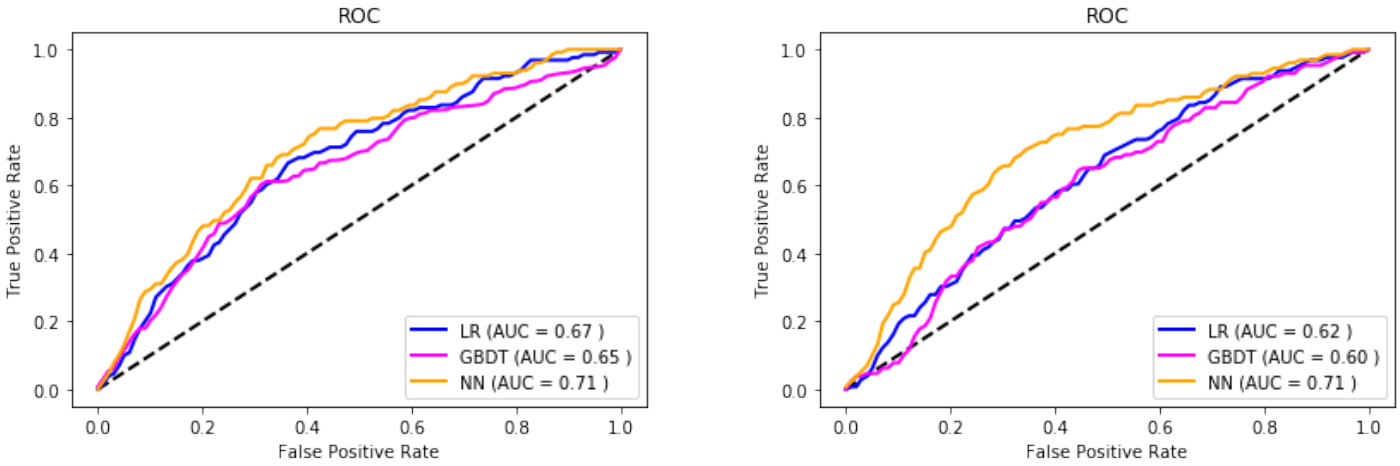



(a) ROC curves from 6-fold cross-validation

(b) Confusion Matrix

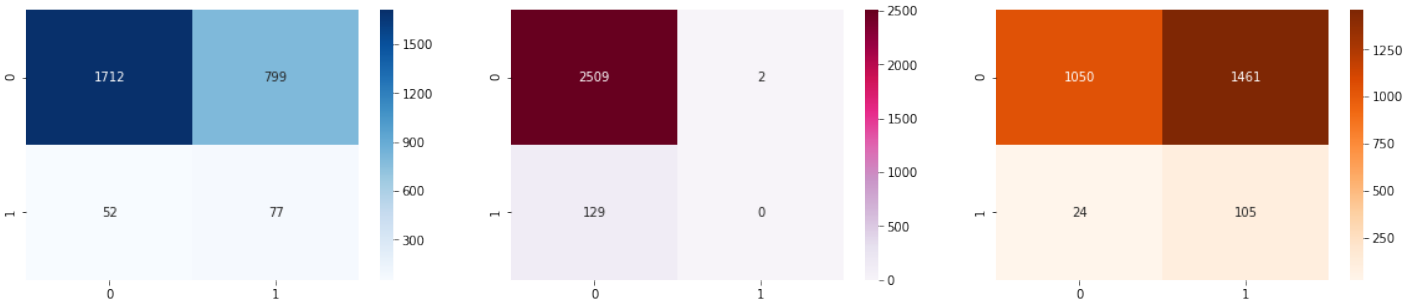
Figure 15: NN with optimized architecture

## 7.2 Comparing the selected models



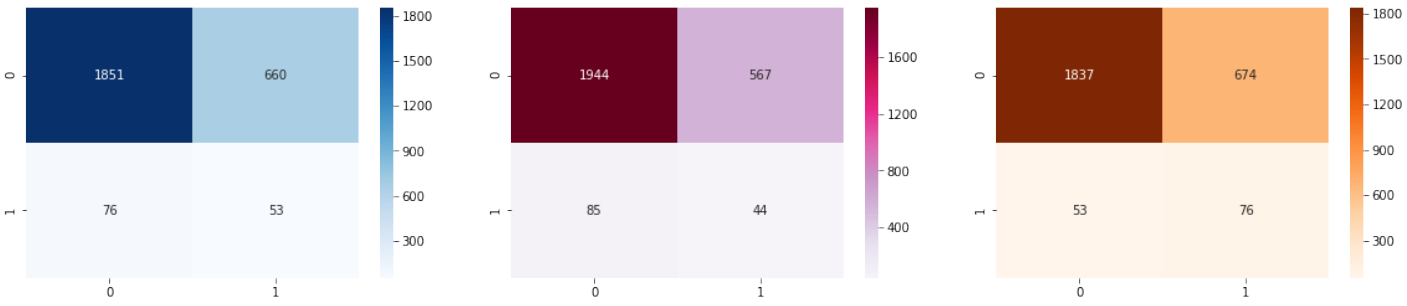

(a) **noADASYN** (b) **ADASYN**

Figure 16: Average ROCs of LR, GBDT, and NN in 6-fold RCV (noADASYN, ADASYN)




(a) LR (b) GBDT (c) NN

Figure 17: Confusion Matrices for LR, GBDT, and NN in 6-fold RCV **(noADASYN)**




(a) LR (b) GBDT (c) NN

Figure 18: Confusion Matrices for LR, GBDT, and NN in 6-fold RCV **(ADASYN)**

Following the most common evaluation metric for predicting readmission, in this study, AUROC is used as a quantitative means of comparing the predictive performance among the classifiers. In general, an AUC of 0.5 suggests no class separation capacity, 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding. By observing the AUROC plots, for both - before and after class imbalance correction (Figure 16), it is apparent that the only model with acceptable performance is the Neural Network with a score of 0.71. This means that there is a 71% chance that the model will be able to distinguish between the readmission and no-readmission classes. To test whether the AUROC of the Neural Network is significantly higher than the AUCs of the other methods, a $p$-value from one-sided t-test statistics is used. The reported $p$-value, equal to 0.25 ($p > 0.05$), shows that the AUROC scores of LR and GBDT, with or without ADASYN, are statistically indistinguishable from the AUROC score for NN. Comparing the AUROC scores with respect to class imbalance correction shows that applying ADASYN actually decreases the performance at least in the case of Logistic Regression and Gradient Boosted Decision Trees models. No difference in the values of AUROC is observed for Neural Network with or without ADASYN class imbalance correction. However, AUROC scores are not enough for a fair comparison. Other performance measures, with the default decision threshold which is 0.5, are derived from the confusion matrices. These are Recall, Precision, Negative Predictive Value, and F1-score (refer to Section 2.6 for an overview of the mentioned metrics). Note that these scores are all subject to change once the decision threshold is adjusted. One can do that by taking the ROC curves as a reference. All evaluation measures for the methods with and without class imbalance correction are shown in Table 12.

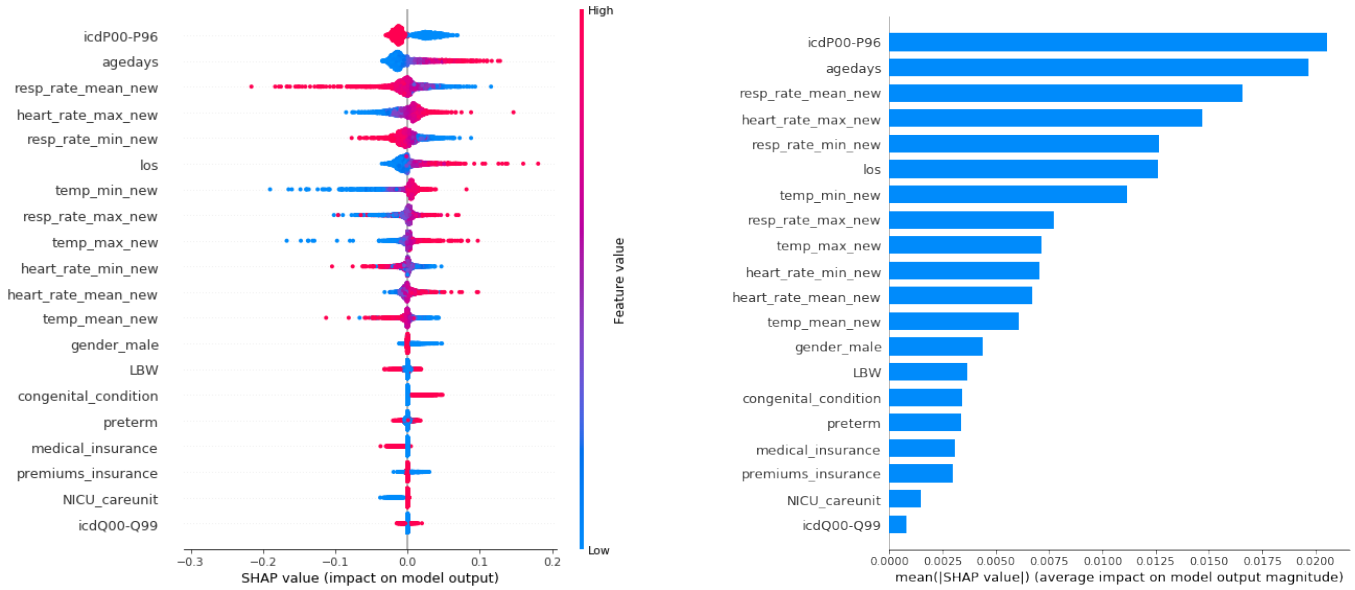
| Model | Data Sampling | AUROC | Recall | Precision - PPV (%) | NPV (%) | F1-score |
|---|---|---|---|---|---|---|
| LR | noADASYN | 0.67 | 0.6 | 8.78% | 97.06% | 0.15 |
| | ADASYN | 0.62 | 0.41 | 7.42% | 96.06% | 0.13 |
| GBDT | noADASYN | 0.65 | 0 | 0 | 95.04% | 0 |
| | ADASYN | 0.6 | 0.34 | 7.19% | 95.82% | 0.12 |
| NN | noADASYN | 0.71 | 0.81 | 6.70% | 97.77% | 0.12 |
| | ADASYN | 0.71 | 0.59 | 10.12% | 97.20% | 0.17 |

Table 12: Performance measures of LR, GBDT, and NN before and after oversampling correction of class imbalance with ADASYN

Now that more measures are included, it can be confirmed that applying ADASYN does not improve the performance of Logistic Regression. In fact, a decline in scores can be observed in all of the evaluation measures. While there is a decrease in the AUROC score of GBDT after applying ADASYN, an increase in the Recall, Precision, NPV, and F1-score are observed. This is due to the fact that the model failed to make any true positive predictions before class imbalance correction was applied (Figure 17 (b)). As a result, no recall, precision or F1-score could be derived, leaving them with values of 0 - hence the improvement observed after ADASYN sampling. Despite, Gradient Boosted Decision Tree, with or without ADASYN, is still the worst-performing of all methods used in this study. In the case of Neural Network, the results pre and after class imbalance correction are further analysed.

An ideal score for the Recall is 1.0, which means that 100% of the samples that should have been labelled as a positive class were labelled correctly. In the task of readmission prediction, having a high score for the recall measure is very important as we want to correctly classify readmissions and minimize the instances of readmissions incorrectly classified as no-readmissions. The results reported in Table 12 show that NN with no ADASYN imbalance correction achieves the greatest recall score of 0.81, meaning that 81% of readmissions were correctly classified, which can also be observed from the Confusion Matrix in Figure 17 (c). However, this score says nothing about how many other samples were also labelled as readmissions, but should not have been. The precision score reflects that information. Again, the Neural Network, this time, with ADASYN imbalance correction, scores the best Precision among the other methods - 10.12%. This means that the chance that a positive classification being indeed correct is only 10.12%. This low value is due to the highly imbalanced classes. For example, the chance of a negative classification being indeed correct is above 95% among all methods (see NPV at Table 12). To find harmony between Recall and Precision, F-score is observed. Based on the F1-score (0.17) and F2-score(0.30), the overall best model is the Neural Network with ADASYN.

## 7.3 Features impact




(a) **a density scatter plot of SHAP values**

(b) **a bar plot of the average impact per feature**

Figure 19: Features impact on Neural Network (with ADASYN) output

To identify how much impact each model has on the Neural Network model output, Shapley (Shap) values were calculated. A Shapley value is the average marginal contribution of a feature value over all possible combinations of the other features. Shap values are the most useful for understanding the positive/negative effect of each feature on the model's prediction. The proper interpretation of the Shapley value is: "Given the current set of

feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction is the estimated Shapley value."[4]

Figure 19 (a) shows a density scatter plot of the SHAP values. Each row corresponds to a feature and each dot corresponds to a training sample. The color indicates how changes in the value of a feature affect the change in risk of readmission. Red represents high feature value, while blue represents low feature value. All the features are sorted by the sum of the Shap value magnitudes actress all training samples. We can see that the primary indicator of readmission is the lack of conditions originating in the perinatal period (*icdP00-P96*). The next most powerful indicator of readmission is the age of the newborn - the older, the higher the risk. Next is the respiratory rate, heart rate, length of stay, and so on.

[4]Molnar, C., 2021. 5.9 Shapley Values — Interpretable Machine Learning. [online] Available at: `https://christophm.github.io/interpretable-ml-book/shapley.html` [Accessed 3 March 2021].

# 8 Discussion

While readmission prediction has been extensively studied in adult patients, little attention is paid to the readmission of pediatric patients, let alone neonatal patients. Studies have attempted to simply explain the statistical connections between different variables and the readmission outcome for neonates. However, they have not extended their analysis to measure predictive performance for readmission. This study extends on previous studies by implementing three distinct prediction models to predict readmission of neonatal patients to Intensive Care Units. Due to the heavy class imbalance of the data, an experiment with Adaptive Synthetic sampling approach was also conducted to further improve the prediction. In this chapter, the methods, and the results and their meaning are discussed and reflected on. Future work and limitations are integrated into the discussion of relevant parts.

## 8.1 Discussion on data preprocessing

Medical databases are particularly susceptible to missing data. Often researchers address missing data by including only complete cases – ones that have no missing data in most of the features needed for the analysis. However, even then it is not possible to completely reduce the amount of missing data to zero. In this study, the features that have missing data are the vital signs, such as temperature, respiratory rate, oxygen saturation, etc. Missingness in such type of features is expected since not all patients undergo the same examination in the ICUs. Because some of the variables have a large portion of missing values, features were selected based on their completeness. Features that have above 30% missing data were excluded from the analysis. Moreover, records with 3 or more missing features were also excluded – so that if a patient has missing temperature, respiratory rate, and heart rate values all together, then the patient is excluded from the dataset. After deleting these records, the readmission rate remained the same and the class imbalance problem was not intensified. While this method is common and easy to implement, it can cause both bias and loss of statistical power. Discarding records or features with missing values is not always a good idea. The fact that data are missing can hold important information. This is a limitation of this study and in the future, more emphasis should be placed on investigating the reasons for data missingness. While it is not possible to distinguish between *missing at random* and *missing not at random* using observed data, biases caused by data that are *missing not at random* can be addressed by sensitivity analyses examining the effect of different assumptions about the missing data mechanism. When it is plausible that data are *missing at random*, biases can be overcome by using methods such as multiple imputation that allow patients with incomplete data to be included in analyses.

Not only the vital signs features contained missing values, but they also contained mistakes - presumably typographical errors. For example, temperature of 370°C, respiratory rate of 660 bpm[5], or heart rate of 1160 bpm[6]. This is a problem when extracting the minimum, maximum and mean values of the vital signs of interest. To deal with those errors, a filter was applied to each feature. The boundaries of the filters were decided on approximations of acceptable values for each vital sign. While this method helped with removing outliers, it is not robust against information loss. A suggestion for the future would be to use the median of the vital signs instead of the minimum, maximum and mean values. In

[5]breaths per minute
[6]beats per minute

such a way, the possible typographical errors would not play a role and information loss would be avoided.

## 8.2 Discussion on hyperparameters optimization

Cross-validate Grid Searches took place to optimize the hyperparameters of the Logistic Regression and Gradient Boosted Decision Trees, whereas the Neural Network architecture was optimized manually. The Grid Search was performed on the imbalanced data and the cross-validation results were evaluated on their average AUC scores. What is important to note is that after class imbalance correction with ADASYN, the balanced data was fed to the model optimized on the imbalanced data, as the parameters responsible for weighting the classes for Logistic Regression and Neural Network were excluded. New hyperparameters optimization was not performed, however, different hyperparameters might optimize a model which is fed with the balanced data.

The results from the Grid Search for the Logistic Regression were fairly consistent throughout the folds. However, the results for the Gradient Boosted Decision Trees were abnormal. There could be observed significant differences between the outer fold (test) AUC scores and the Grid Search (validation) AUC scores (see Table 11). In the 3rd fold, the best AUC from the Grid Search is significantly higher than the validation on the test set in the outer fold. The reverse can be observed in the 4th fold. These abnormal results could mean that the data reserved for validation in the outer folds is not a good representation of the data used while searching for the best parameters. Eventually, it was decided to select the best hyperparameter based on their frequency, rather than the ones associated with the highest AUC scores in the outer fold.

## 8.3 Discussion on imbalance correction

In this study, the readmissions account for only 4.8% of the samples. This class imbalance in the targeted class is a common problem in medical datasets. Two different approaches for dealing with class imbalance are used here - one is to adjust the class weights during learning and the other is an oversampling approach - ADASYN, where synthetic samples are generated based on harder to learn minority class samples until the class balance is restored. The results showed that the oversampling approach caused degradation of the Logistic Regression model's performance. This could be due to different class distribution in train and test data. Oversampling the minority class in the training set balances the ratio of positive and negative classes, while the ratio of the classes in the testing set stays unchanged - unbalanced. Logistic Regression optimizes deviance, which is strongly distributional; hence it is more sensitive to a mismatch between training and test class distributions. Moreover, in ADASYN, some of the harder to learn samples might be outliers, thus, the algorithm produces some synthetic instances based on noise. This type of oversampling behaviour could be problematic for classifiers if the synthetic values are just representative of very rare samples.

Restoring the 1:1 class balance, when in reality there is a large disparity between the classes will not necessarily improve the performance. While it is relevant in some cases, oversampling changes the underlying distribution of the data, as already mentioned above. This means that the insights from the classifier trained on over-sampled training data do not transfer to the unseen test data that by definition has a very different distribution for features to the over-sampled training set. The implementation of Adaptive Synthetic

sampling helped GBDT to classify some true readmissions, but it also increased the False Positive classification errors. This could be due to the shifted distribution of the training set. The reverse can be observed in the case with the Neural Network, which was trained with weighted classes, first, and then with ADASYN sampled data. After applying ADASYN the number of False Positive classifications decreased, but so did the True Positive ones (observe Figure 17 & 18 (c)). The Neural Network with weighted classes (and without ADASYN) truthfully predicted many readmissions, but at the same time, it classified wrongly more than half of the negative cases as positive, which decreased the precision of the model. While for this study, it is desired to maximize the True Positives, the high values for False Positives, achieved with the Neural Network without ADASYN, decrease the authenticity of the truthful model's predictions. Thus, the Neural Network with ADASYN was named as the best performing model in this study, despite the decrease in TPs that it showed. As a future recommendation, different class imbalance correction techniques and class balance ratios should be explored.

## 8.4 Discussion on models performance

| paper | LR | GBDT (or similar) | NN (or similar) | Patient population |
|---|---|---|---|---|
| McWilliams et al. [3] | 0.85 AUC | | | adult |
| Rojas et al. [32] | | 0.76 AUC | | adult |
| Pakbin et al. [8] | 0.73 AUC | 0.75 AUC | | adult |
| Nguyen et al. [4] | | 0.79 AUC | | adult |
| Futoma et al. [7] | 0.72 AUC | | | adult |
| Ben-Assuli [9] | 0.91 AUC | 0.925 AUC | 0.88 AUC | adult |
| Jovanovic et al. [46] | 0.78 AUC | | | pediatric |
| Wolff et al. [45] | | | 0.64 AUC | pediatric |
| This study | 0.67 AUC | 0.65 AUC | 0.71 AUC | neonatal |

Table 13: Comparing the AUROC values achieved in the reviewed literature and this study

Contrary to the reported in the literature results, the performance of the Gradient Boosted Decision Trees model in this study is rather poor (Table 13). Even though the Gradient Boosted Machines are known to deal well with class imbalance by constructing successive training sets based on incorrectly classified examples, in this case, the GBDT model is being outperformed by the LR model. This could be because the Gradient Boosted Machines cannot handle well information that is out of the range of the training data, while LR has no problem with extrapolating. It could be also that the GBDT model needs more extensive hyperparameters tuning than what has been done. Overall, the poor performance of GBDT indicates that the data used in this study fits better with Logistic Regression or Neural Network models. This is inconsistent with the trends in literature, where tree-based Gradient Boosted models outperform both Logistic Regression and Neural Network models, when reported together (Table 13). In general, the models' performance achieved in this study is hardly comparable to the results found in the literature. The implemented here

models outperform only the Multilayer Perceptron utilized by Wolff et al. [45] to predict readmission to the emergency department of a pediatric hospital. Even the best performing model in this study, which is the Neural Network with an AUROC score of 0.71, cannot outperform any of the Logistic Regressions or Gradient Boosted models reported in the literature. However, it should be noted that the related studies, to which the performance of this study was compared to, are very different and thus any comparison is biased. First of all, these studies cover completely different patient population, namely adult patients. Moreover, most of the studies also use multiple database sources to enrich and enlarge their datasets, whereas in this study only one data source is used. Furthermore, the readmission rate for the adult patient population is significantly higher compared to the ones of pediatric and neonatal populations. Hence the class imbalance that the reviewed literature dealt with is not as prominent as the one faced in this study.

# 9 Conclusion

This study is concluded by answering the research questions as stated in Section 1.2.

**RQ1: To what extent one can predict readmission of neonates to an intensive care unit using machine learning algorithms?**

Test results showed that the Neural Network model developed in this study can achieve an AUROC score of 0.71, which is an acceptable value for AUC in general. Although comparisons of the performances with literature indicates that the data and model developed in this study are subject to improvement.

**RQ1.1: Given the class imbalance dataset, does class balancing during data preprocessing improve the classification performance?**

The imbalance correction technique used is ADASYN. This oversampling method did not improve the AUROC score for any of the implemented classification algorithms. In fact, it had a degrading effect on Logistic Regression for all evaluation measures. While it resulted in decreased AUROC score for the Gradient Boosted Decision Trees, it showed improvement in the recall and precision scores. In the case of Neural Network, the AUROC score was neither improved nor worsen, however, the harmony between recall and precision (F-score) was enhanced. Furthermore, the improvement of the Neural Network performance by ADASYN is not much higher than no handling. Therefore, it is acceptable to not apply class imbalance handling during data preprocessing.

**RQ2: Which machine learning algorithm performs best in the classification task?**

Neural Network, with ADASYN, outperformed the other candidates with AUROC of 0.71, even though this score was not statistically distinguishable from the AUCs of Logistic Regression and Gradient Boosted Decision Trees. Surprisingly the worst performing model is Gradient Boosted Decision Trees, achieving AUROC of 0.65.

**RQ2.1: Which features, used in this study, have the greatest impact on the output of the model from RQ2?**

The features' impact on the output of the Neural Network model are presented in Figure 19. The top five most powerful indicators of readmission are conditions originating in the perinatal period (*icdP00-P96*), age of the newborn (*agedays*), average respiratory rate (*resp_rate_mean*), highest heart rate (*heart_rate_max*), and the lowest respiratory rate (*resp_rate_min*).

# A Appx: ICD10 Categories

| | |
|---|---|
| A00-B99 | Certain infectious and parasitic diseases |
| C00-D48 | Neoplasms |
| D50-D89 | Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism |
| E00-E90 | Endocrine, nutritional and metabolic diseases |
| F00-F90 | Mental, Behavioral and Neurodevelopmental disorders |
| G00-G90 | Diseases of the nervous system |
| H00-H59 | Diseases of the eye and adnexa |
| H60-H95 | Diseases of the ear and mastoid process |
| I00-I99 | Diseases of the circulatory system |
| J00-J99 | Diseases of the respiratory system |
| K00-K93 | Diseases of the digestive system |
| L00-L99 | Diseases of the skin and subcutaneous tissue |
| M00-M99 | Diseases of the musculoskeletal system and connective tissue |
| N00-N99 | Diseases of the genitourinary system |
| O00-O99 | Pregnancy, childbirth and the puerperium |
| P00-P96 | Certain conditions originating in the perinatal period |
| Q00-Q99 | Congenital malformations, deformations and chromosomal abnormalities |
| R00-R99 | Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified |
| S00-T98 | Injury, poisoning and certain other consequences of external causes |
| V01-Y98 | External causes of morbidity |
| Z00-Z99 | Factors influencing health status and contact with health services |

Table 14: ICD10 Categories

# B Appx: Congenital conditions & Birth asphyxia varieties

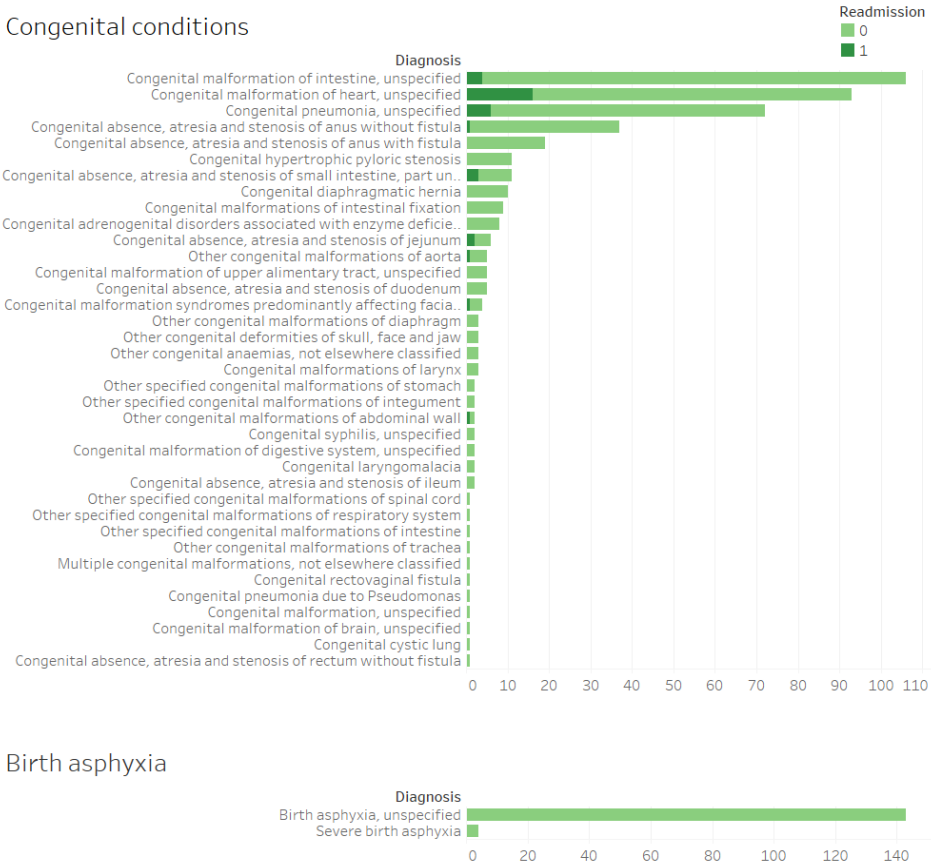



Figure 20: Variety of Congenital conditions & Birth asphyxia

# C Appx: PIC database overview of tables

| Table name | Description | Rows |
|---|---|---|
| ADMISSIONS | Every unique hospitalization for each patient in the database (defines HADM_ID) | 13,449 |
| CHARTEVENTS | All charted observations for the patients from the hospital database, not including lab events. | 2,278,978 |
| D_ICD_DIAGNOSES | Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-10) and International Classification of Diseases for Oncology (ICD-O-3) codes relating to diagnoses. | 25,378 |
| D_ITEMS | Dictionary of local codes ('ITEMIDs') appearing in the PIC database, except those that relate to laboratory tests. | 466 |
| D_LABITEMS | Dictionary of local codes ('ITEMIDs') appearing in the PIC database that relate to laboratory tests. | 832 |
| DIAGNOSES_ICD | Hospital assigned diagnoses, coded using the ICD-10 English code and the ICD-O-3 Chinese code. | 13,365 |
| EMR_SYMPTOMS | Structured symptoms extracted from notes, including nursing and physician notes, discharge summaries and so on. | 402,142 |
| ICUSTAYS | Every unique ICU stay in the database (defines ICUSTAY_ID). | 13,941 |
| INPUTEVENTS | Calculated fluid input data every morning for each patient in ICU. | 26,884 |
| LABEVENTS | Laboratory measurements for patients from the hospital database. | 10,094,117 |
| MICROBIOLOGYEVENTS | Microbiology culture results and antibiotic sensitivities from the hospital database. | 183,869 |
| OR_EXAM_REPORTS | Contains all exams performed during the patient's stay. | 183,809 |
| OUTPUTEVENTS | Output information for patients from the hospital database. | 39,891 |
| PATIENTS | Every unique patient in the database (defines SUBJECT_ID). | 12,881 |
| PRESCRIPTIONS | Medications ordered for a given patient | 1,256,591 |
| SURGERY_VITAL_SIGNS | Vital signs recorded every 5 minutes during the surgery | 1,216,011 |

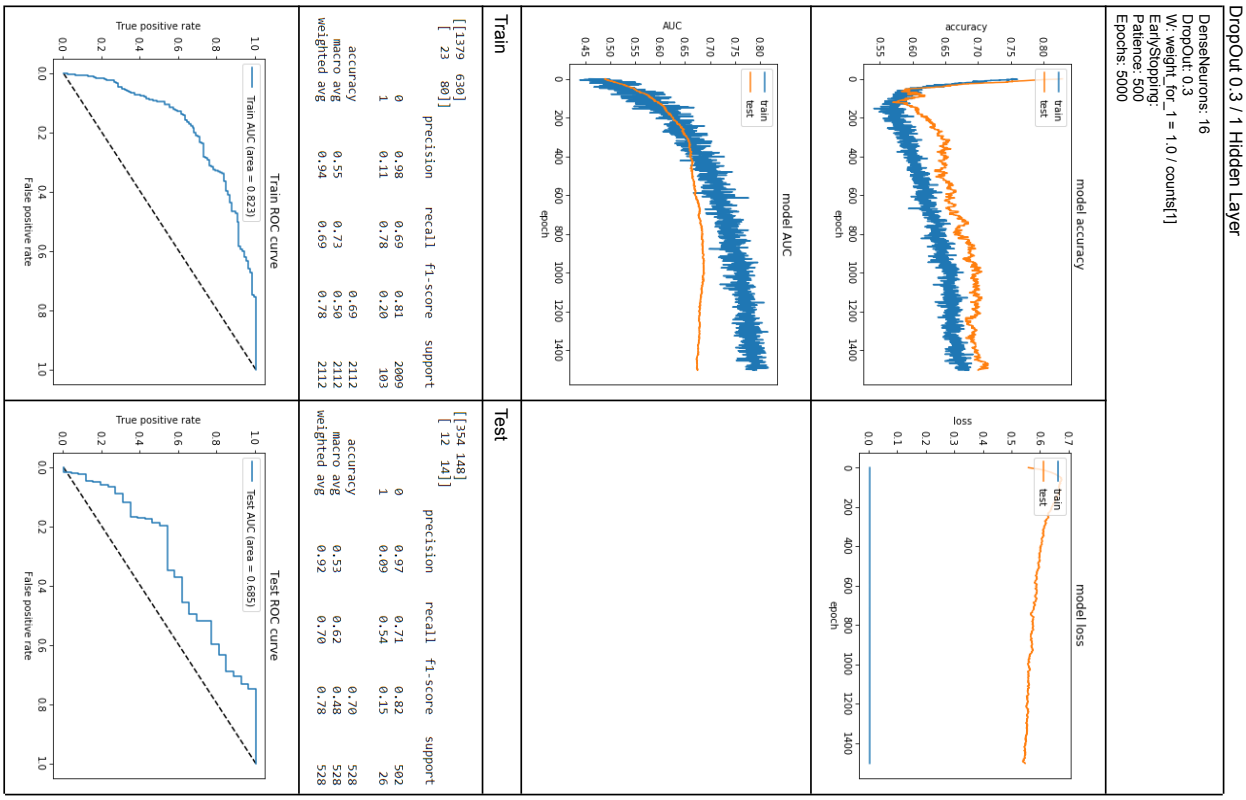
Table 15: An overview of the data tables in the PIC database [17]

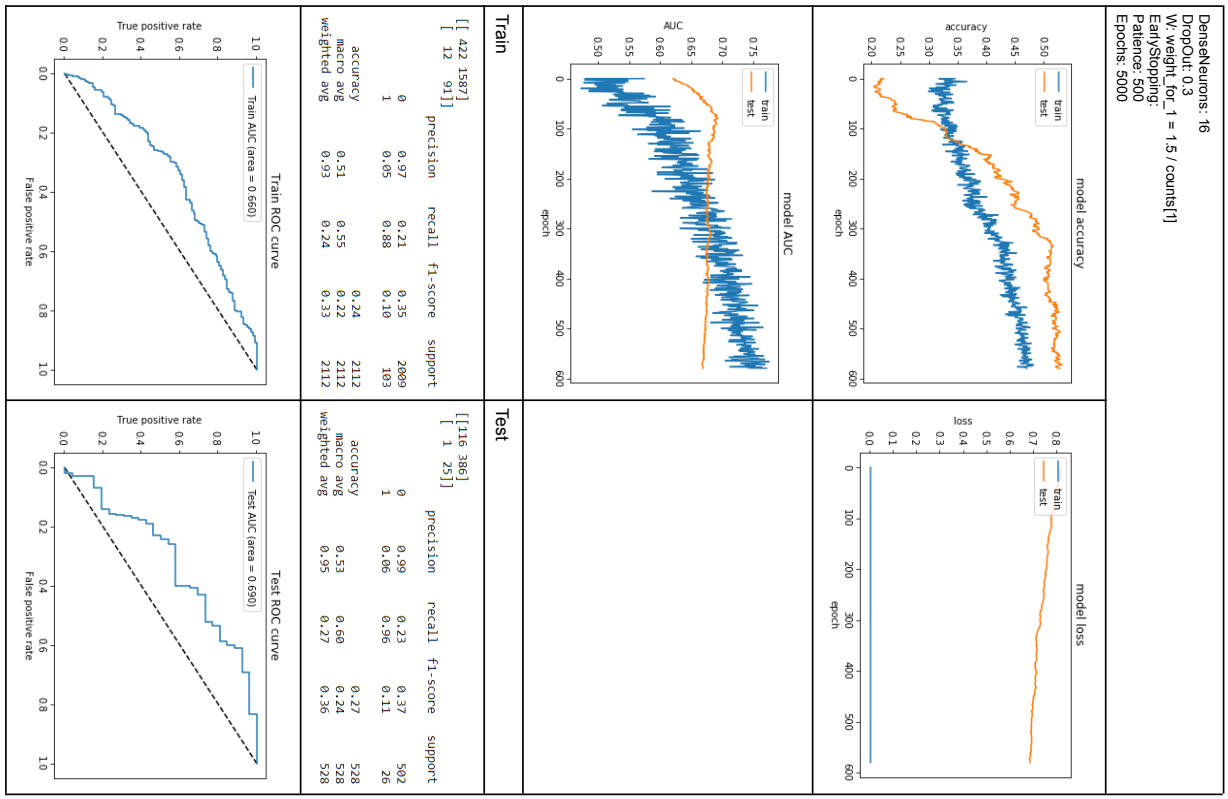
# D Appx. Initial GridSearch for GBDT

| | |
|---|---|
| GridSearch = {<br>'learning_rate':[0.1,0.01,0.001],<br>'n_estimators':[500,1000]<br>} | Val AUC: 0.5893851574042506 Best GS AUC: 0.6143509666454218 Best Params: {'learning_rate': 0.01, 'n_estimators': 500}<br>Val AUC: 0.6870098874872144 Best GS AUC: 0.6084643438849939 Best Params: {'learning_rate': 0.001, 'n_estimators': 1000}<br>Val AUC: 0.5099443118536197 Best GS AUC: 0.6203880744989732 Best Params: {'learning_rate': 0.01, 'n_estimators': 1000}<br>Val AUC: 0.701174423662462 Best GS AUC: 0.5747471668768084 Best Params: {'learning_rate': 0.001, 'n_estimators': 1000}<br>Val AUC: 0.5797085689430187 Best GS AUC: 0.6383411030859325 Best Params: {'learning_rate': 0.001, 'n_estimators': 1000}<br>Val AUC: 0.6873640713353633 Best GS AUC: 0.6013859586892716 Best Params: {'learning_rate': 0.01, 'n_estimators': 500} |
| GridSearch = {<br>'learning_rate':[0.001],<br>'n_estimators':[1000, 1500]<br>} | Val AUC: 0.6235367655415389 Best GS AUC: 0.6147935698604915 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.681100125014206 Best GS AUC: 0.6096682246299837 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.5825093760654619 Best GS AUC: 0.6162541604702216 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.7123205741626795 Best GS AUC: 0.5916534271647991 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.6134188777729448 Best GS AUC: 0.6468553965500257 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.6784471509351893 Best GS AUC: 0.6293390988236767 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500} |
| GridSearch = {<br>'learning_rate':[0.001],<br>'n_estimators':[1500, 1750]<br>} | Val AUC: 0.6358677122400275 Best GS AUC: 0.6175554139225268 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6822934424366405 Best GS AUC: 0.6097124849514907 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.5825093760654619 Best GS AUC: 0.616218752213016 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.7099282296650717 Best GS AUC: 0.592543477410218 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6134188777729448 Best GS AUC: 0.6468553965500257 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.6728468899521531 Best GS AUC: 0.6306440012185182 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750} |
| GridSearch = {<br>'learning_rate':[0.001],<br>'n_estimators':[1750, 2000]<br>} | Val AUC: 0.6357540629617002 Best GS AUC: 0.617537709793924 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6766109785202864 Best GS AUC: 0.6108101409248636 Best Params: {'learning_rate': 0.001, 'n_estimators': 2000}<br>Val AUC: 0.5788725991589954 Best GS AUC: 0.6160328588626867 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.7099282296650717 Best GS AUC: 0.5925061316269007 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6171705089169204 Best GS AUC: 0.6435890413502326 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6728468899521531 Best GS AUC: 0.6307325218615321 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750} |
| GridSearch = {<br>'learning_rate':[0.001],<br>'n_estimators':[1550, 1750]<br>} | Val AUC: 0.6358677122400274 Best GS AUC: 0.6175731180511296 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6832594613024208 Best GS AUC: 0.6109694780822887 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550}<br>Val AUC: 0.5852369587453119 Best GS AUC: 0.6159266340910701 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550}<br>Val AUC: 0.7099282296650717 Best GS AUC: 0.5925244815975407 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750}<br>Val AUC: 0.6116789908655936 Best GS AUC: 0.6454884367921857 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550}<br>Val AUC: 0.6728468899521531 Best GS AUC: 0.6307148177329293 Best Params: {'learning_rate': 0.001, 'n_estimators': 1750} |
| GridSearch = {<br>'learning_rate':[0.001],<br>'n_estimators':[1500, 1550]<br>} | Val AUC: 0.6243323104898283 Best GS AUC: 0.6152804333970682 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550}<br>Val AUC: 0.6832594613024208 Best GS AUC: 0.6109340698250831 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550}<br>Val AUC: 0.5825093760654619 Best GS AUC: 0.6162010480844133 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.7123205741626795 Best GS AUC: 0.591690127106079 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.6135276207046543 Best GS AUC: 0.6469085089358341 Best Params: {'learning_rate': 0.001, 'n_estimators': 1500}<br>Val AUC: 0.6785558938668987 Best GS AUC: 0.6305675982354587 Best Params: {'learning_rate': 0.001, 'n_estimators': 1550} |

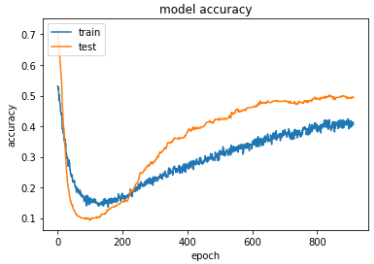
Table 16: Initial GridSearch for the GBDT focused on defining the learning rate and the number of estimators

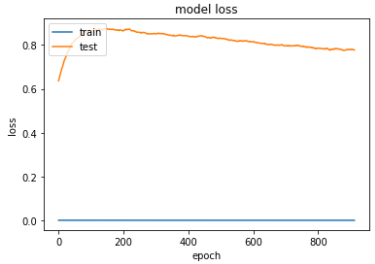
# E Appx. Tuning experiments for NN

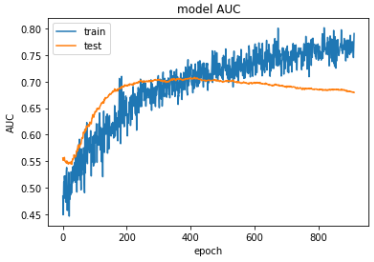


DenseNeurons: 16
DropOut: 0.3
W: weight_for_1 = 2.0 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000

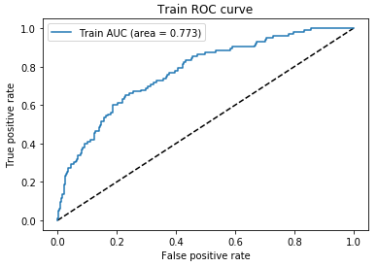


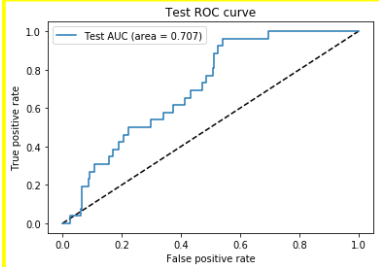

Train

```
[[ 576 1433]
 [   5   98]]
              precision    recall  f1-score   support

           0       0.99      0.29      0.44      2009
           1       0.06      0.95      0.12       103

    accuracy                           0.32      2112
   macro avg       0.53      0.62      0.28      2112
weighted avg       0.95      0.32      0.43      2112
```



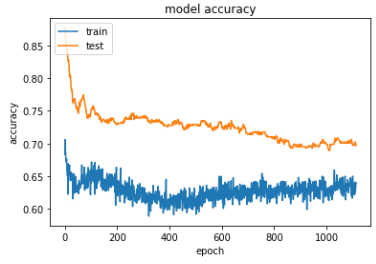

Test

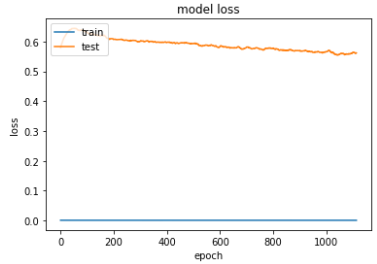
```
[[179 323]
 [  1  25]]
              precision    recall  f1-score   support

           0       0.99      0.36      0.52       502
           1       0.07      0.96      0.13        26

    accuracy                           0.39       528
   macro avg       0.53      0.66      0.33       528
weighted avg       0.95      0.39      0.51       528
```

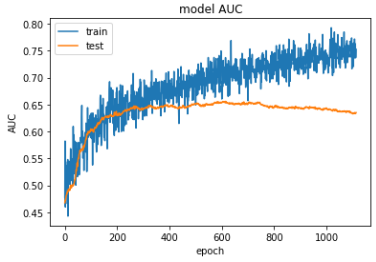



DropOut 0.4 / 1 Hidden Layer

DenseNeurons: 16
DropOut: 0.4
W: weight_for_1 = 1.0 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000

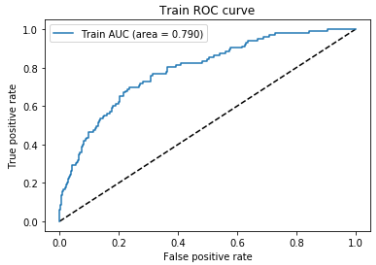


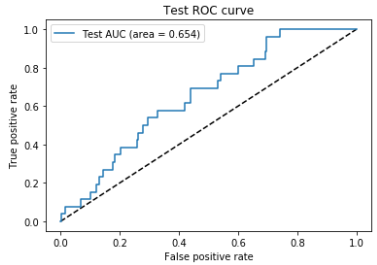

Train

```
[[1466  543]
 [  30   73]]
              precision    recall  f1-score   support

           0       0.98      0.73      0.84      2009
           1       0.12      0.71      0.20       103

    accuracy                           0.73      2112
   macro avg       0.55      0.72      0.52      2112
weighted avg       0.94      0.73      0.81      2112
```

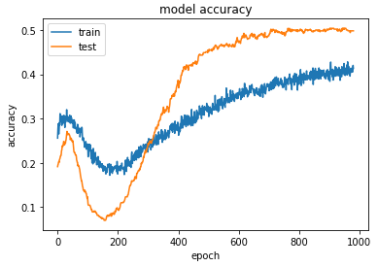


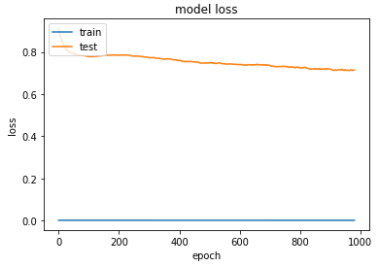

Test

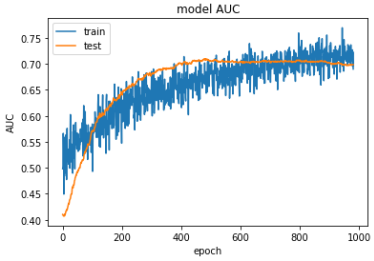
```
[[371 131]
 [ 15  11]]
              precision    recall  f1-score   support

           0       0.96      0.74      0.84       502
           1       0.08      0.42      0.13        26

    accuracy                           0.72       528
   macro avg       0.52      0.58      0.48       528
weighted avg       0.92      0.72      0.80       528
```



DenseNeurons: 16
DropOut: 0.4
W: weight_for_1 = 1.5 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000



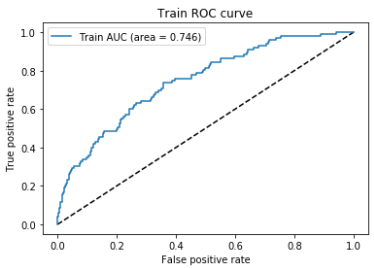



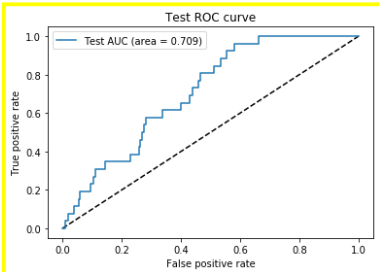

**Train**

```
[[ 701 1308]
 [   9   94]]
              precision    recall  f1-score   support

           0       0.99      0.35      0.52      2009
           1       0.07      0.91      0.12       103

    accuracy                           0.38      2112
   macro avg       0.53      0.63      0.32      2112
weighted avg       0.94      0.38      0.50      2112
```

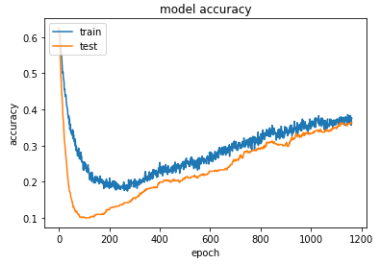
**Test**

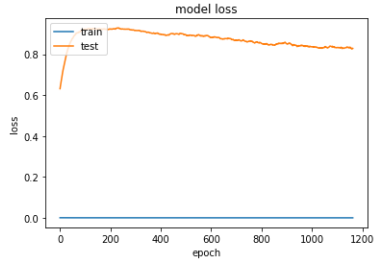
```
[[210 292]
 [  2  24]]
              precision    recall  f1-score   support

           0       0.99      0.42      0.59       502
           1       0.08      0.92      0.14        26

    accuracy                           0.44       528
   macro avg       0.53      0.67      0.36       528
weighted avg       0.95      0.44      0.57       528
```

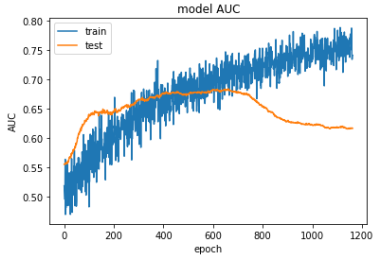



DenseNeurons: 16
DropOut: 0.4
W: weight_for_1 = 2.0 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000



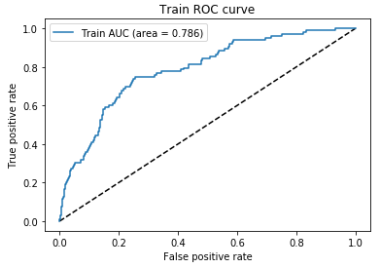



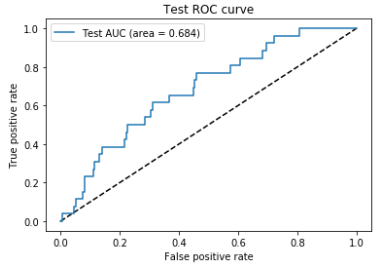

**Train**

```
[[ 371 1638]
 [   3  100]]
              precision    recall  f1-score   support

           0       0.99      0.18      0.31      2009
           1       0.06      0.97      0.11       103

    accuracy                           0.22      2112
   macro avg       0.52      0.58      0.21      2112
weighted avg       0.95      0.22      0.30      2112
```

**Test**

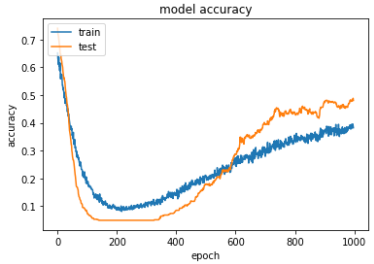
```
[[100 402]
 [  1  25]]
              precision    recall  f1-score   support

           0       0.99      0.20      0.33       502
           1       0.06      0.96      0.11        26

    accuracy                           0.24       528
   macro avg       0.52      0.58      0.22       528
weighted avg       0.94      0.24      0.32       528
```

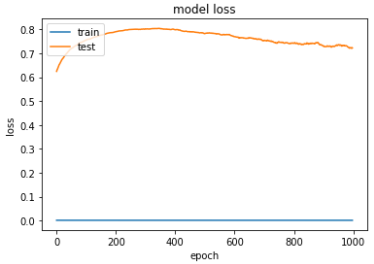


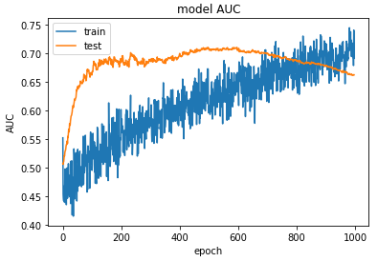



DropOut 0.4 / 2x Hidden Layers

2x(DenseNeurons: 16
DropOut: 0.4)
W: weight_for_1 = 1.5 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000



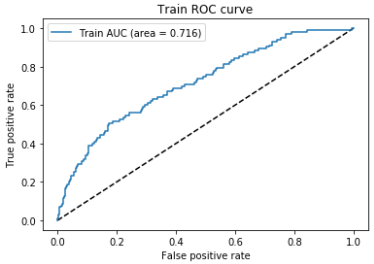



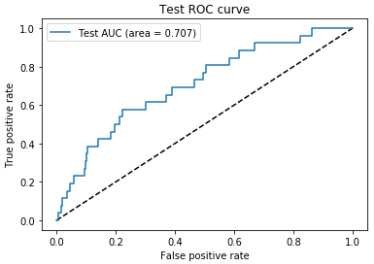

Train

```
[[ 217 1792]
 [   1  102]]
              precision    recall  f1-score   support

           0       1.00      0.11      0.19      2009
           1       0.05      0.99      0.10       103

    accuracy                           0.15      2112
   macro avg       0.52      0.55      0.15      2112
weighted avg       0.95      0.15      0.19      2112
```

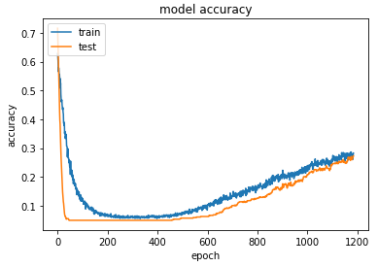
Test

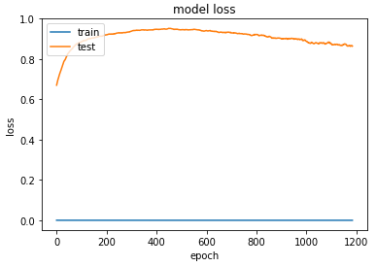
```
[[ 67 435]
 [  0  26]]
              precision    recall  f1-score   support

           0       1.00      0.13      0.24       502
           1       0.06      1.00      0.11        26

    accuracy                           0.18       528
   macro avg       0.53      0.57      0.17       528
weighted avg       0.95      0.18      0.23       528
```

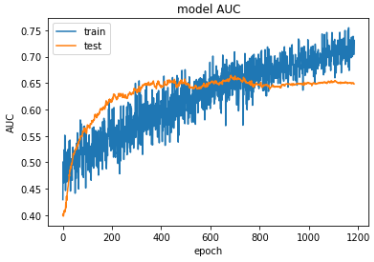



2x(DenseNeurons: 16
DropOut: 0.4)
W: weight_for_1 = 2.0 / counts[1]
EarlyStopping:
Patience: 500
Epochs: 5000



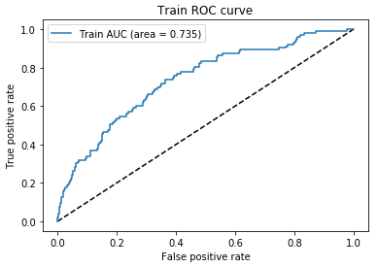



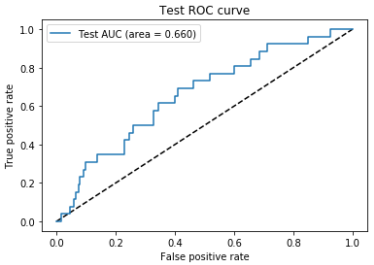

Train

```
[[  75 1934]
 [   1  102]]
              precision    recall  f1-score   support

           0       0.99      0.04      0.07      2009
           1       0.05      0.99      0.10       103

    accuracy                           0.08      2112
   macro avg       0.52      0.51      0.08      2112
weighted avg       0.94      0.08      0.07      2112
```

Test

```
[[ 22 480]
 [  0  26]]
              precision    recall  f1-score   support

           0       1.00      0.04      0.08       502
           1       0.05      1.00      0.10        26

    accuracy                           0.09       528
   macro avg       0.53      0.52      0.09       528
weighted avg       0.95      0.09      0.08       528
```



# F Appx. NLD criteria

| Test ID | Test name | Variable | Test condition |
|---|---|---|---|
| R0 | Respiratory: airway | airway | airway patent |
| R1 | Respiratory: Fio2 | fio2 | fio2<0.6 |
| R2 | Respiratory: blood oxygen | spo2 | spo2>95 (%) |
| R3 | Respiratory: bicarbonate | hco3 | hco3>19 (mmol/L) |
| R4 | Respiratory: rate | resp (rate) | 10<resp<30 (bpm) |
| C0 | Cardiovascular: blood pressure | bp (systolic) | bp>100 (mm Hg) |
| C1 | Cardiovascular: heart rate | hr | 60<hour<100 (bpm) |
| P | Pain | pain | 0<pain<1 |
| CNS | Central nervous system | gcs | gcs>14 |
| T | Temperature | temp | 36<temp<37.5 (°C) |
| B0 | Bloods: haemoglobin | haemoglobin | haemoglobin>90 (g/L) |
| B1 | Bloods: potassium | k | 3.5<k<6.0 (mmol/L) |
| B2 | Bloods: sodium | na | 130<na<150 (mmol/L) |
| B3 | Bloods: creatinine | creatinine | 59<creatinine<104 (umol/L) |
| B4 | Bloods: urea | bun | 2.5<bun<7.8 (mmol/L) |

Table 17: NLD criteria

# References


[1] J. C. Marshall, L. Bosco, N. K. Adhikari, B. Connolly, J. V. Diaz, T. Dorman, R. A. Fowler, G. Meyfroidt, S. Nakagawa, P. Pelosi, *et al.*, "What is an intensive care unit? a report of the task force of the world federation of societies of intensive and critical care medicine," *Journal of critical care*, vol. 37, pp. 270–276, 2017.

[2] U. R. Ofoma, S. Chandra, R. Kashyap, V. Herasevich, A. Ahmed, O. Gajic, B. W. Pickering, and C. J. Farmer, "Findings from the implementation of a validated readmission predictive tool in the discharge workflow of a medical intensive care unit," *Annals of the American Thoracic Society*, vol. 11, no. 5, pp. 737–743, 2014.

[3] C. J. McWilliams, D. J. Lawson, R. Santos-Rodriguez, I. D. Gilchrist, A. Champneys, T. H. Gould, M. J. Thomas, and C. P. Bourdeaux, "Towards a decision support tool for intensive care discharge: machine learning algorithm development using electronic healthcare data from mimic-iii and bristol, uk," *BMJ open*, vol. 9, no. 3, p. e025925, 2019.

[4] D.-P. Nguyen, N. Paris, and A. Parrot, "Accurate and reproducible prediction of icu readmissions," *medRxiv*, 2019.

[5] C. R. Ponzoni, T. D. Corrêa, R. R. Filho, A. Serpa Neto, M. S. Assunção, A. Pardini, and G. P. Schettino, "Readmission to the intensive care unit: incidence, risk factors, resource use, and outcomes. a retrospective cohort study," *Annals of the American Thoracic Society*, vol. 14, no. 8, pp. 1312–1319, 2017.

[6] C. K. McIlvennan, Z. J. Eapen, and L. A. Allen, "Hospital readmissions reduction program," *Circulation*, vol. 131, no. 20, pp. 1796–1803, 2015.

[7] J. Futoma, J. Morris, and J. Lucas, "A comparison of models for predicting early hospital readmissions," *Journal of biomedical informatics*, vol. 56, pp. 229–238, 2015.

[8] A. Pakbin, P. Rafi, N. Hurley, W. Schulz, M. H. Krumholz, and J. B. Mortazavi, "Prediction of icu readmissions using data at patient discharge," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4932–4935, IEEE, 2018.

[9] O. Ben-Assuli and R. Padman, "Analysing repeated hospital readmissions using data mining techniques," *Health Systems*, vol. 7, no. 2, pp. 120–134, 2018.

[10] A. M. Jorgensen, "Born in the usa–the history of neonatology in the united states: a century of caring," *NICU Currents*, vol. 1, no. 1, pp. 8–11, 2010.

[11] S. Oddie, D. Hammal, S. Richmond, and L. Parker, "Early discharge and readmission to hospital in the first month of life in the northern region of the uk during 1998: a case cohort study," *Archives of disease in childhood*, vol. 90, no. 2, pp. 119–124, 2005.

[12] P. C. Young, K. Korgenski, and K. F. Buchi, "Early readmission of newborns in a large health care system," *Pediatrics*, vol. 131, no. 5, pp. e1538–e1544, 2013.

[13] S. Agarwal, "Data mining: data mining concepts and techniques," in *2013 International Conference on Machine Intelligence and Research Advancement*, pp. 203–207, IEEE, 2013.

[14] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining (adaptive computation and machine learning)*. MIT Press, 2001.

[15] I. Yoo, P. Alafaireet, M. Marinov, K. Pena-Hernandez, R. Gopidi, J.-F. Chang, and L. Hua, "Data mining in healthcare and biomedicine: a survey of the literature," *Journal of medical systems*, vol. 36, no. 4, pp. 2431–2448, 2012.

[16] R. S. Brid, "Decision trees-a simple way to visualize a decision." URL: `https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb`, Oct 2018. (Accessed: 20-02-2020).

[17] X. Zeng, G. Yu, Y. Lu, L. Tan, X. Wu, S. Shi, H. Duan, Q. Shu, and H. Li, "Pic, a paediatric-specific intensive care database," *Scientific Data*, vol. 7, no. 1, pp. 1–8, 2020.

[18] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[19] J. J. Berman, "Confidentiality issues for medical data miners," *Artificial intelligence in medicine*, vol. 26, no. 1-2, pp. 25–36, 2002.

[20] H. C. Koh, G. Tan, *et al.*, "Data mining applications in healthcare," *Journal of healthcare information management*, vol. 19, no. 2, p. 65, 2011.

[21] R. Ichise and M. Numao, "Learning first-order rules to handle medical data," *NII journal*, vol. 3, no. 2, pp. 9–14, 2001.

[22] G. Gillespie, "There's gold in them thar'databases.," *Health data management*, vol. 8, no. 11, pp. 40–4, 2000.

[23] F. Cismondi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. Sousa, and S. N. Finkelstein, "Missing data in medical databases: Impute, delete or classify?," *Artificial intelligence in medicine*, vol. 58, no. 1, pp. 63–72, 2013.

[24] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 1987.

[25] P. D. Allison, *Missing data*, vol. 136. Sage publications, 2001.

[26] H. Kang, "The prevention and handling of the missing data," *Korean journal of anesthesiology*, vol. 64, no. 5, p. 402, 2013.

[27] A. Batista and C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, vol. 17, p. 519–533, May 2003.

[28] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?," in *Proceedings of the 1st international workshop on Utility-based data mining*, pp. 69–77, 2005.

[29] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328, IEEE, 2008.

[30] A. Ali, S. M. Shamsuddin, A. L. Ralescu, *et al.*, "Classification with class imbalance problem: a review," *Int. J. Advance Soft Compu. Appl*, vol. 7, no. 3, pp. 176–204, 2015.

[31] G. Knight, "Nurse-led discharge from high dependency unit.," *Nursing in critical care*, vol. 8, no. 2, pp. 56–61, 2003.

[32] J. C. Rojas, K. A. Carey, D. P. Edelson, L. R. Venable, M. D. Howell, and M. M. Churpek, "Predicting intensive care unit readmission with machine learning using electronic health record data," *Annals of the American Thoracic Society*, vol. 15, no. 7, pp. 846–853, 2018.

[33] O. Gajic, M. Malinchoc, T. B. Comfere, M. R. Harris, A. Achouiti, M. Yilmaz, M. J. Schultz, R. D. Hubmayr, B. Afessa, and J. C. Farmer, "The stability and workload index for transfer score predicts unplanned intensive care unit patient readmission: initial development and validation," *Critical care medicine*, vol. 36, no. 3, pp. 676–682, 2008.

[34] C. Stenhouse, S. Coates, M. Tivey, P. Allsop, and T. Parker, "Prospective evaluation of a modified early warning score to aid earlier detection of patients developing critical illness on a general surgical ward," *British Journal of Anaesthesia*, vol. 84, no. 5, p. 663P, 2000.

[35] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

[36] B. Zheng, J. Zhang, S. W. Yoon, S. S. Lam, M. Khasawneh, and S. Poranki, "Predictive modeling of hospital readmissions using metaheuristics and data mining," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7110–7120, 2015.

[37] T. Desautels, R. Das, J. Calvert, M. Trivedi, C. Summers, D. J. Wales, and A. Ercole, "Prediction of early unplanned intensive care unit readmission in a uk tertiary care hospital: a cross-sectional machine learning approach," *BMJ open*, vol. 7, no. 9, p. e017199, 2017.

[38] S. Linton, C. Grant, J. Pellegrini, and A. Davidson, "The development of a clinical markers score to predict readmission to paediatric intensive care," *Intensive and critical care nursing*, vol. 25, no. 6, pp. 283–293, 2009.

[39] P. J. Fite, L. Stoppelbein, and L. Greening, "Predicting readmission to a child psychiatric inpatient unit: The impact of parenting styles," *Journal of Child and family Studies*, vol. 18, no. 5, pp. 621–629, 2009.

[40] S. Burokienė, I. Kairienė, M. Stričkа, L. Labanauskas, R. Čerkauskienė, J. Raistenskis, E. Burokaitė, and V. Usonis, "Unscheduled return visits to a pediatric emergency department," *Medicina*, vol. 53, no. 1, pp. 66–71, 2017.

[41] A. M. Bernard and A. S. Czaja, "Unplanned pediatric intensive care unit readmissions: A single-center experience," *Journal of critical care*, vol. 28, no. 5, pp. 625–633, 2013.

[42] K. A. Auger, E. L. Mueller, S. H. Weinberg, C. S. Forster, A. Shah, C. Wolski, G. Mussman, A. J. Ipsaro, and M. M. Davis, "A validated method for identifying unplanned pediatric readmission," *The Journal of pediatrics*, vol. 170, pp. 105–112, 2016.

[43] I. Bergese, S. Frigerio, M. Clari, E. Castagno, A. De Clemente, E. Ponticelli, E. Scavino, and P. Berchialla, "An innovative model to predict pediatric emergency department return visits," *Pediatric emergency care*, vol. 35, no. 3, pp. 231–236, 2019.

[44] A. Artetxe, B. Ayerdi, M. Graña, and S. Rios, "Using anticipative hybrid extreme rotation forest to predict emergency service readmission risk," *Journal of Computational Science*, vol. 20, pp. 154–161, 2017.

[45] P. Wolff, M. Graña, S. A. Ríos, and M. B. Yarza, "Machine learning readmission risk modeling: A pediatric case study," *BioMed research international*, vol. 2019, 2019.

[46] M. Jovanovic, S. Radovanovic, M. Vukicevic, S. Van Poucke, and B. Delibasic, "Building interpretable predictive models for pediatric hospital readmission using tree-lasso logistic regression," *Artificial intelligence in medicine*, vol. 72, pp. 12–21, 2016.

[47] S. Radovanovic, M. Vukicevic, A. Kovacevic, G. Stiglic, and Z. Obradovic, "Domain knowledge based hierarchical feature selection for 30-day hospital readmission prediction," in *Conference on Artificial Intelligence in Medicine in Europe*, pp. 96–100, Springer, 2015.

[48] W. Barfield, S. Manning, C. Kroelinger, J. Martin, D. Barradas, *et al.*, "Neonatal intensive-care unit admission of infants with very low birth weight-19 states, 2006.," *Morbidity and Mortality Weekly Report*, vol. 59, no. 44, pp. 1444–1447, 2010.

[49] M. M. Rahman and D. N. Davis, "Machine learning-based missing value imputation method for clinical datasets," in *IAENG transactions on engineering technologies*, pp. 245–257, Springer, 2013.