Utrecht University

Department of Information and Computing Science

Applied Data Science master thesis

Detecting deterioration in patients with congenital heart diseases at the pediatric intensive care unit

Supervising team:

dr. Joppe Nijman, MD PhD

prof. dr. Arno P.J.M. Siebes, PhD

Ruben S. Zoodsma, BSc

drs. Erik Koomen, MD

Candidate:

Daniel van der Mee Mendes

In cooperation with:

Wilhelmina Childrens Hospital part of UMC Utrecht

July 7, 2023

Acknowledgement

I would like to express my sincere gratitude to dr. Joppe Nijman, MD PhD and Ruben S. Zoodsma, BSc for their guidance and feedback over the last months. Without them this study would not have been at the level it is now.

Furthermore, I would like to thank prof. dr. Arno P.J.M. Siebes, PhD for his support and time during this period.

Finally, I would like to thank the pediatric intensive care unit of the Wilhelmina Childrens Hospital part of UMC Utrecht for the opportunity of doing my master's thesis at their department and for their feedback. From this department, I would like to thank drs. Erik Koomen, MD in special for his feedback and effort to make me more familiar with the workflow at the hospital.

Abstract

Critical congenital heart disease (cCHD) is present in two to three of every 1,000 newborns. Children diagnosed with cCHD are admitted to the pediatric intensive care unit (PICU) and closely monitored to ensure the highest possible quality of healthcare. During this period large quantities of continuous data streams are collected. The aim of this study is to analyze these large quantities of data using machine learning techniques such as random forest and boosting to provide insights in detecting deterioration by classifying periods as stable and unstable. The data consisted of 86 patients with information on five vital signs: heart rate, respiratory rate, invasive mean blood pressure, oxygen saturation and regional cerebral oxygen saturation. Pre-processing steps were necessary to transform the data and generate artificial labels for model training. Using the pre-processed data, hyperparameter tuning was performed, and final models were created. Based on these models, classifications were made on a left-out dataset consisting of nine patients. These model classifications are compared with a clinical classification established by a medical expert. The findings revealed an accuracy range of 64.4% to 87.1%, a sensitivity range of 66.0% to 97.3%, and a specificity range of 23.1% to 94.3%. These numbers demonstrate generally favorable accuracy and sensitivity scores. However, some models had very low specificity scores, indicating large amounts of true unstable periods were not classified as unstable. Relying on such a model would result in missing many critical situations at the PICU. Nevertheless, some models showed great potential. This study highlights that machine learning techniques such as random forest and boosting can be used to provide insights in detecting deterioration in patients. Consequently, it is recommended to explore the best performing models further and assess if further improvements are possible. Additionally, it is important to analyze the reasons behind certain incorrect classifications to enhance the understanding of the model. Finally, choosing a best model depends on the task at hand and should be considered carefully.

Contents

1. Introduction

Critical congenital heart disease (cCHD) is present in two to three of every 1,000 newborns where congenital heart disease is the most frequently occurring congenital disease in newborns [1] [2]. It is categorized as cCHD if cardiac intervention is required for survival within a newborn's first year [3]. Over the last years mortality rates have dropped massively for these infants and nearly 85% reaches adulthood. Nevertheless, these infants are vulnerable for complications such as brain injury and delayed motor development. Therefore, it is necessary to focus not only on one specific disease, but also and especially on co-morbidity. To prevent these co-morbidities, monitoring is essential. Children diagnosed with cCHD are admitted to the pediatric intensive care unit (PICU) and closely monitored to ensure optimal healthcare and minimize the likelihood of complications, thereby improving their quality of life. During this period large quantities of lab, image, discontinuous and continuous data streams are collected and used by nurses and doctors for clinical assessment [3]. Nevertheless, it can be difficult to get a good overview of all available data at ones and to use all information in the best possible way for clinical assessment. Therefore, it would be helpful if certain models could analyze at least parts of all available data.

Machine learning is still new to healthcare and few examples exist of an implementation within the clinical workflow of departments working with adult patients [4]. Moreover, no study describes an implementation at the PICU clinical workflow [3]. Nevertheless, it gives a lot of opportunities and enables an in-depth analysis of all the data collected of patients at the PICU [5]. Furthermore, AI and machine learning have the ability to change how healthcare is practiced under the consideration certain aspects are carefully examined [6]. For instance, while a classification model may yield high sensitivity and accuracy scores, its specificity could be much lower. This specificity might be

critical for the implementation of a model, thus not examining all aspects carefully can result in unfavorable outcomes. Studies showed that machine learning can be used to make accurate predictions in healthcare [4] [5] [7]–[11]. The majority of these studies focus on support vector machine (SVM), random forest and boosting models, which are able to outperform more traditional models. To highlight a few studies, Pirneskoski et al. applied a random forest model to predict one-day mortality rates, surpassing the performance of the prehospital national early warning score. In addition, Clifton et al. demonstrated that their SVM model was the superior classifier in identifying deterioration in patients at the emergency department. In short, several studies indicate potential in applying machine learning techniques to provide better insights compared to traditional methods focused on critical situations such as detecting deterioration.

Studies also address how these new developments should be implemented in the workflow to reduce workload but also describe the challenges of the actual implementation [4] [5]. These challenges can be mitigated through good communication between nurses, doctors, and the data specialists creating the models. With these developments, machine learning can be combined with expert opinions to improve the quality of healthcare while reducing workload.

The aim of this study is to analyze data from infants (< 1 year) at the PICU with cCHD by using machine learning techniques such as random forest and boosting. The implementation of machine learning models can provide better insights in detecting deterioration of patients at specific time points. A good use of such a prediction model can potentially identify unstable periods in patients very quickly, reducing the workload of nurses and doctors, both of which can improve the quality of healthcare. This would also be beneficial for the prevention of co-morbidities, as quick intervention is possible. Furthermore, a well-performing model that accurately classifies stable periods can also provide significant benefits. A patient being stable indicates that no intervention is required at that moment in time, allowing the family to spend uninterrupted time with their child. Moreover, insights into stable periods among patients

can be valuable when dealing with another patient in critical condition. The nurses and doctors can directly see if there are any other patients that require intervention. If all patients are stable except for the critical patient, all necessary help can be allocated to the critical patient.

The aim is translated into the following research question:
How well can machine learning algorithms such as random forest and boosting classify deterioration in infants with cCHD considering their vital parameters compared to the already implemented support vector machine algorithm?

2. Data

This chapter describes what data is used and which pre-processing steps have been taken to prepare the data for further analysis. All coding for this study was done using R in a digital research environment created by the hospital. The code used for this study can be found on the website of the Pediatric (Cardiac) Critical Care and Data Science research group from the University Medical Center Utrecht, The Netherlands [12].

2.1 Data overview

The data used for this study was collected at the PICU of the Wilhelmina Childrens Hospital, part of UMC Utrecht, The Netherlands. The dataset consisted of 86 patients ($<$ 1 year) all diagnosed with cCHD where each patient has its own patient ID and timeline starting from zero when admitted to the PICU. However, the anatomy and physiology of cCHD patients can be quite different, resulting in a heterogeneous patient group, all spending a different amount of time at the hospital. The time element is added as sequential observations, for example a patient can have 5000 observations all at different times. The dataset consisted of information on five collected vital signs: heart rate, respiratory rate, oxygen saturation, invasive mean blood pressure, and regional cerebral oxygen saturation. Furthermore, information is available on whether the patients received mechanical ventilation.

The data was provided fully anonymized in the research environment by assigning patients an integer number, for instance number one for the first patient. Furthermore, date and time information was replaced by an incremental change of 1 every minute starting at zero. Without a name and specific time and date information, it was not possible in this study to identify a patient with the resources available. Due to the use of fully anonymized data, the con-

sent of the patient's caregivers was waived by the Medical Ethical Research Committee of the UMC Utrecht (no. 22-822).

2.2 Data preparation

The data described in the previous section requires pre-processing steps to make it suitable for training machine learning algorithms. The data does not provide any labels on the patient being stable or unstable at any given moment in time, thus should be added by using adequate methods. This emphasizes the great importance of the preparation phase. Therefore, data has to be cleaned, modified and analyzed to ensure the quality of the models used. All these steps were already documented by previous work and used for their model, but were revisited to make sure they would work for the models in this study [3]. As stated in previous work, no golden standard is present to determine artificial labels for indicating deterioration, but a best combination of steps was chosen and used. Nevertheless, it is possible that some steps may be adapted to new findings.

With the data in the digital environment, the first cleaning steps can be taken. First, each patient is analyzed if they are being ventilated by considering an end-tidal carbon dioxide (EtCO2) above zero at each moment in time. The EtCO2 is equal to the amount of carbon dioxide present in the expiratory air of a ventilated patient which is a standard to use at the hospital to determine the efficacy of the mechanical ventilation. In other words, the availability of this value indicates that the patient is being ventilated. For times when patients are ventilated and the respiratory rate is below 5, the ventilation respiratory rate is used instead of the normal respiratory rate. This replacement is executed to describe the respiratory rate as best as possible as the normal respiratory rate can be incorrect when being ventilated. Infants can also have large fluctuations in their respiratory rate, thus a five-minute moving average is used. Secondly, regional cerebral oxygen saturation is collected in both hemispheres. If only one value was available, this value was used as the vital sign value. Otherwise a mean between both hemispheres was calculated and used. Third, as the

artificial label creation method does not allow missing values, only complete data was used and patients with less than 12 hours of complete data were removed. Within this study, no patients were removed as all patients consisted of more than 12 hours of data. The remaining patients can be used for analysis and observations are sorted by time per patient. Furthermore, a separation is made between patient with a mean oxygen saturation during admission below 90% (low group) or equal and above 90% (high group). This separation is created to account for underlying varying physiology (heterogeneity) in cCHD patients [3]. Splitting the data in two groups resulted in 26 patients in the low group and 60 patients in the high group. Besides the separation on oxygen saturation, equal steps were taking on the complete dataset to determine if accurate models could be established without accounting for the underlying varying physiology.

The data is now ready to start the process of creating artificial stable and unstable labels. Each of the following steps were applied on the low group, high group and total data. First, the five vital signs were standardized to a mean of zero and a standard deviation of 1. The standardized columns were used to create a five-by-five covariance matrix. This covariance is required for the Mahalanobis calculation. The Mahalanobis is a method that is used to perform dimension reduction, in this situation a reduction to one principal component [13]. In other words, the five vital sign values per time point are reduced to one value by taking into account the correlations between the vital signs used and a mean. Non-complying correlations of vital signs and vital sign values far from the mean will result in a high Mahalanobis. The Mahalanobis values, at times a patient is ventilated, are multiplied by 1.2. This multiplication is performed, because ventilated patients have reduced variance in their vital signs due to the controlled setting where their respiratory rate is regulated by a machine. With Mahalanobis values present for each time point a baseline can be established by taking the median of all the Mahalanobis values available until that specific moment in time.

The baseline and the Mahalanobis are the necessary information sources to get an understanding on which measures were an indication of deterioration. First, the baseline is subtracted from the Mahalanobis. Secondly, all these differences of the patients are sorted and the highest 20% differences above zero are given the label unstable where all other observations get the label stable. A negative difference would indicate that the baseline is above the Mahalanobis, meaning that a patient is improving and stable. To elaborate, the baseline is a balanced line which determines the general trend of a patient over time considering the median of the Mahalanobis values. As mentioned before, high Mahalanobis values occur if the vital signs are far from the mean and are not following the correlations of the vital signs. A Mahalanobis below this line would indicate that the vital signs are closer to the mean and following the correlations, indicating an improvement compared to the general trend. Therefore, it is preferred to see patients with a Mahalanobis close but preferably below the baseline. However, a patient that is always stable will most likely not be at the PICU. Furthermore, the artificial labels do not only contain an 80/20 split classification, but also a 90/10 split classification. In other words, the data consisted of one additional column where 20% of the observations are unstable and one column where 10% of the observations are unstable. Finally, the first hour of each patient was removed from the data, because it is not used to train the model. This first hour often consists of fluctuations due to the admission period and processes that occur at that period (e.g. starting of certain medication, placement of lines). Using this part for classification could influence the overall performance. With the artificial labels present visualisations of the distributions of the vital signs per group separated on stable and unstable periods can be created (Appendix A). These visualisations provide additional insights in the data. For instance, the respiratory rate distribution indicates that ventilated patients are overrepresented in the stable group. This finding supports the decision to multiply the Mahalanobis by 1.2.

The data is fully prepared and cleaned, but still needs to be split into a training and test set. The training set will be used to train the model where the test set is used to evaluate the performance. For this study the dataset is not split on data, but on the number of patients. In the low group 20 out of 26 patients were selected to by in the training set. The high group training set consisted of 45 out of 60 patients. At last, the training set without the group separation consisted of 65 out of 86 patients. Although this split does not result in a data split of exactly 80/20%, it is still within the recommended boundaries of splitting a dataset. Splitting on patients rather than data is preferred for this study to ensure that the model is always evaluated on new unseen patients. Splitting the dataset on the data itself would result in patient information being present in the training and test set. The model would be able to learn the patterns of each patient, resulting in an overestimation of true performance. Applying the same model to new unseen data would most likely result in a lower performance. This approach is chosen to have more realistic performance metrics during the model training process compared to the final model performance. The splitting of patients is done randomly.

The training set is also used to perform 5-fold cross-validation (CV) for hyperparameter tuning. Therefore, another patient split is performed to create 5 folds from the training data. The low group ended up with 5 folds with each containing 4 patients where the high group folds contained 9 patients each. The total data ended up with 5 folds containing 13 patients. With the cleaning, modifying, analyzing and patient splitting steps described in this section, the data is fully prepared to perform hyperparameter tuning, training and evaluating the models. To summarize, Figure 2.1 illustrates the flow of data as described above.

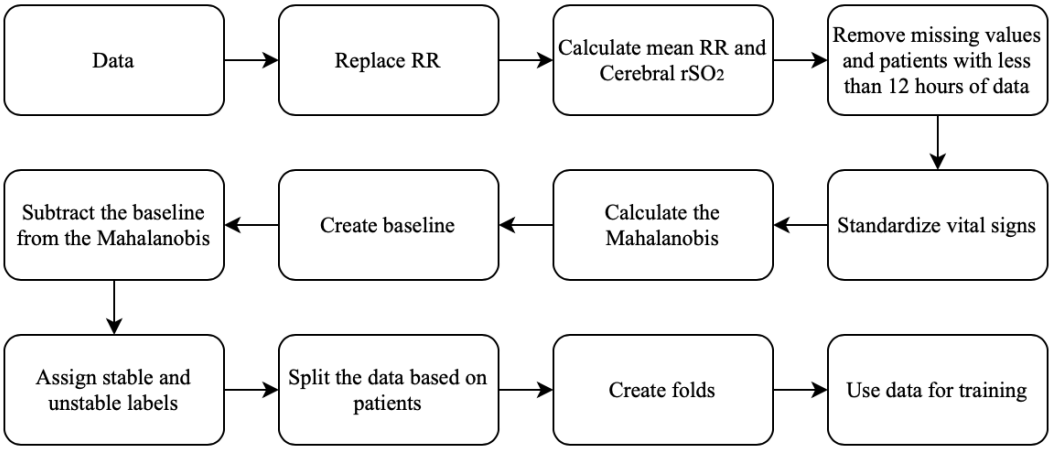


Figure 2.1: Data preparation flow chart. RR: respiratory rate, Cerebral rSO2: regional cerebral oxygen saturation

3. Methods

To answer the research question, four steps are required, each with its own methods. These steps involve training classification algorithms, comparing the model performance based on artificial labels, smoothing of final classifications and comparing the model performance based on clinical labels. The two methods used for classification are random forest and boosting. Both methods are selected, because studies showed that they can accurately make predictions in healthcare [7] [9] [11]. Furthermore, the properties of both models match the expectations stated by the hospital. They strongly prefer a model that can accurately make classifications without losing too much on interpretability. The accurate classifications are required to ensure a good performance when such a model is used in practice. Especially in healthcare, missing critical situations could have negative consequences for that particular patient. Nevertheless, interpretability might be as important as accuracy. Healthcare is a field in society that highly relies on the expertise of medical experts such as nurses and doctors. The classification or decision of a data-driven machine learning technique might not always be equal to the clinical opinion or decision. This disagreement might be resolved if the medical expert understands why the algorithm makes a certain decision. Furthermore, machine learning techniques will most likely be used to provide additional insights and help the nurses and doctors, not to replace them. Therefore, it is important that these techniques are able to explain in some sense what is happening and why certain decisions are made.

Besides random forest and boosting, other methods are able to make classifications as well. Logistic regression is one of the simplest classification algorithms. Logistic regression is strong in terms of interpretability, as you get estimates for each variable, but may lack the ability to achieve high accuracies, or at least will most likely not outperform the random forest and boosting techniques.

For instance, logistic regression can have difficulties with correlated variables as is the case in this study. Furthermore, it can be sensitive to outliers and lacks robustness [14]. Deep learning can also be used for classification. These type of models are normally very strong considering the accuracy, but do not provide much in terms of interpretability. Moreover, deep learning requires large amounts of data to achieve a high accuracy which is not the case for this study [15] [16]. Therefore, a balance between accuracy and interpretability is created. Both random forest and boosting techniques consist of this balance and are somewhat similar to the already implemented support vector machine model. These models also have the ability to reduce overfitting and deal with correlated variables [16] [17].

3.1 Random Forest

Random forest can be classified as an ensemble technique that aims to construct a strong learner by combining multiple weak learners. The strong learner is an aggregated tree, based on a chosen number of simple decision trees, which gives a prediction considering all the information from the single trees. Nevertheless, the simple decision trees are adjusted to increase accuracy and mitigate the risk of overfitting. Each tree uses a bootstrap sample of the data, meaning that repeated samples are taken from the training set. This process is referred to as bagging. Furthermore, the variables to determine the best split at each node are not always equal. Depending on the "mtry" parameter a given number of variables are considered in each split. The variables are chosen randomly at each node [16]. This additional step is what distinguishes bagging from random forest and usually leads to better performance.

Machine learning techniques often have the opportunity to tune the model's hyperparameters considering the available data and research objectives. In this study, hyperparameter tuning is conducted by applying a 5-fold cross-validation (CV) grid search on the training data ensuring an equal distribution of patients across each fold. For instance, the low group has 20 patients in the training data, resulting in 4 patients in each fold. Table 3.1 describes

each hyperparameter considered with its input for the grid search. The grid search explores 27 different models, each representing a unique combination of hyperparameters. The performance metric used for evaluation is the accuracy, and the combination of hyperparameters with the highest accuracy is used to train the final model.

Hyperparameter	**Description**	**Possible values**
mtry	Number of random variables looked at, at each split	2, 3, 4
num.trees	Number of trees created	100, 250, 500
min.node.size	Minimum number of observations in each end node	1, 5, 10

Table 3.1: Hyperparameter tuning, random forest

3.2 Boosting

Boosting, similar to random forest, can be classified as an ensemble technique. Boosting also creates multiple trees and generates a prediction incorporating the information of all constructed trees. However, there are notable distinctions between random forest and boosting. Unlike random forest, boosting does not use a bootstrap sample for each tree. Additionally, the trees in random forest are independent, whereas in boosting, each tree is built based on the information of previously grown trees. The error of previously grown trees is memorized and accounted for in the next tree. This process is repeated as long as desired, resulting in a final tree [16].

Hyperparameter tuning is also an important element of boosting and potentially even more so than random forest. Boosting offers a wider range of hyperparameter options and has a greater influence on the model's performance compared to random forest. The "colsample_bytree" parameter is similar to the "mtry" parameter of the random forest model and describes a proportion of variables considered in each tree. Another hyperparameter is "eta", which determines the learning speed of the model. A higher eta value results in faster

weight updates and often requires fewer trees, while a lower eta value is more conservative but requires a larger number of trees to achieve good performance. Choosing a high value for eta should be carefully considered due to the risk of overfitting. The number of trees created is determined by the "nrounds" parameter. Lastly, the "alpha" parameter is used to shrink variables towards zero, with zero indicating that a variable does not contribute to the classification. A higher "alpha" value indicates greater shrinkage, due to a higher penalty. The literature used within this study did not specify a direction for the hyperparameter values. Therefore, a wide range of options are included to determine which combination yields the best overall performance considering the relative limited dataset. Table 3.2 provides an overview of each considered hyperparameter with its input for the grid search for the low group model. The grid search explores 162 different models.

Hyperparameter	**Description**	**Possible values**
maxdepth	The depth or layers within a tree	3, 4, 5
eta	How fast the model is trained	0.01, 0.1, 0.3
colsample_bytree	Proportion of variables selected for each tree	0.6, 1
alpha	Shrinking parameter for less important variables	0.01, 0.1, 1
nrounds	Number of trees created	100, 150, 200

Table 3.2: Hyperparameter tuning, boosting

The low model is tuned based on an extensive grid search. To reduce computation time the high group and the model without group separation are tuned based on a narrower range of hyperparameters or possible values. This is done only when certain values were not close to the best option of the low group. It is assumed that good hyperparameter options work similarly for the remaining models. In other words, a selection of the best possible values is made to increase computational efficiency.

3.3 Model performance comparison, artificial

The random forest and boosting output can be analyzed by comparing their confusion matrices based on the artificial labels used for training and testing. Each model gives a confusion matrix, which is the core for calculating different performance metrics such as: accuracy, error rate, and F1-score. Normally, it would be useful to compare different metrics because a high accuracy can be achieved with a high sensitivity but very low specificity. Nevertheless, this comparison is used to gain some insights into which model is better at classifying the given artificial labels, but does not provide information on the clinical relevance, thus only the accuracy is considered. Each model is compared to the same model, but from the other machine learning technique. In other words, the random forest low group model is compared with the boosting low group model. This comparison indicates which model has a higher accuracy and whether this accuracy is significantly different.

3.4 Model performance comparison, clinical

The artificial performance comparison gives some insights in which machine learning technique is better per model on classifying artificial labels, but cannot give any indication on how well it performs compared to the opinion of a clinical expert. The models in the artificial performance comparison section were trained and tested on the 86 patients. The final models for the clinical performance comparison are trained on all available data from the 86 patients and tested on a left out dataset of nine patients. Both the random forest and boosting models provide classifications with stable and unstable labels. However, it is highly likely that unstable labels frequently arise during stable phases and vice versa. These sudden fluctuations might occur during bedside activity such as some quick checks, a parent being present or the withdrawal of blood. A clinical expert would not indicate such a period as unstable, resulting in a model that does not capture true unstable periods and reducing its clinical relevance. Therefore, it is recommended to smooth out these phases and transition from one state to another only if, within the last five minutes,

at least four instances belong to the other state. For example, if a patient is stable at time 10, they will only be considered unstable if, between time 10 and 15, four moments of instability are classified. Another method as mentioned in previous work, is to classify these sudden fluctuations as unreliable or measurement error [3]. The method from that study can also be applied to this data, but is out of the scope for this study. Smoothing of phases will probably also reduce the number of times an alarm is given to nurses and doctors and most likely reduce the number of false alarms. Studies have indicated that 72% to 99% of clinical alarms are false, leading to alarm fatigue [18]. Alarm fatigue is sensory overload when nurses and doctors are exposed to a high number of alarms, which can negatively affect the response when an alarm is given in a real critical situation. Possibly preventing this by smoothing can therefore, improve the quality of healthcare. Smoothing of classification labels is applied to each patient. The window for a possible change in state only considers past events. In case a change in state is occurring the labels are shifted back for four minutes in time to accurately capture the exact moment the change began. In short, stable and unstable moments are smoothed over time considering a rolling window where the final results are shifted back in time by four minutes to capture the real change in state as closely as possible. Nevertheless, when dealing with a gradual change in deterioration or small periods of unstable moments it might be useful to have a short delay of having a unstable phase, thus not shift the period back by four minutes also considering alarm fatigue. It is important to mention that having a delay on deterioration requires ethical and legal considerations. However, this study assumes that the real clinical deterioration is best captured with this short shift back in time.

The smoothed classifications of every model are compared to clinical classifications established by a medical expert on a random window of 24 hours for each patient within the left out dataset. The clinical classification is established by considering the five vital signs included in this study. The clinical performance is determined by creating a confusion matrix where the model classifications are the predictions and the clinical classification, the reference or true labels. Each model gives a confusion matrix which are the core for cal-

culating different performance metrics such as: accuracy, sensitivity, specificity and F1-score. For comparing the models multiple performance metrics are considered. It is important to include metrics such as sensitivity and specificity in the model comparison, due to data imbalance. For example, classifying all data as stable when 90% of the data is stable will still result in an accuracy of 90%. Nevertheless, the specificity would be zero. Therefore, the selection of the best model and machine learning technique is based on an evaluation of different performance metrics. This process of assessing the clinical performance and relevance is part of the internal validation, where the model performance is tested on a left out dataset with the same type of patients and data.

4. Results

This chapter describes the results of the study and consists of three sections. The first section focuses on the hyperparameter tuning and which values resulted in the highest accuracy. Secondly, each model within the random forest approach is compared to the same model of the boosting approach. For instance, the random forest low group model is compared to the boosting low group model. Since these models use the same data, a comparison is made on which machine learning technique can achieve a better performance considering the artificial labels. At last, the performance of the different models is evaluated by comparing the classifications with the clinical labels determined by a medical expert. This final section determines how well the models can be used in practice and their clinical relevance.

4.1 Hyperparameter tuning

Hyperparameter tuning is performed for both the low and high group models as well as for the total models with an 80/20 split. The total models with a 90/10 split are trained based on the hyperparameter values of the 80/20 split models. The 90/10 split models were added later to the study. Due to time and the hyperparameter findings of the other models, it was decided to not perform the whole tuning process for the 90/10 split models as well.

The hyperparameter tuning of the random forest models explored three parameters: mtry, num.trees, and min.node.size. The best values for each hyperparameter per model can be found in Table 4.1. Table 4.1 demonstrates that the low group model which contains the least amount of data requires fewer trees to achieve the best mean 5-fold CV accuracy. However, the increase in data from the high group model to the 80/20 split model did not result in any differences. Besides the number of trees, the other hyperparameters also

differed between the low group model and the other models. The other models gave a better performance considering only two variables per split instead of four and used a higher min.node.size preventing the model from overfitting.

Model	**Hyperparameters**		
	mtry	**num.trees**	**min.node.size**
Low group	4	100	1
High group	2	250	10
Total 80/20 split	2	250	10
Total 90/10 split	2	250	10

Table 4.1: Best hyperparameter values, random forest models

The hyperparameter tuning of the boosting models explored five parameters: maxdepth, eta, colsample_bytree, alpha and nrounds. The best values for each hyperparameter per model can be found in Table 4.2. Table 4.2 illustrates that the best hyperparameter values are similar. However, both low group and high group have an eta value of 0.01, while the 80/20 split model has an eta value of 0.1. In other words, the low and high models have a lower learning rate, thus are more conservative during the training process. The colsample_bytree indicates that the best performance is achieved by considering a proportion of variables for each tree instead of all five of the vital signs. At last, the low alpha value as best hyperparameter value shows that giving a penalty to certain variables does not help the classification performance.

Model	**Hyperparameters**				
	maxdepth	**eta**	**colsample_-bytree**	**alpha**	**nrounds**
Low group	4	0.01	0.6	0.01	100
High group	4	0.01	0.6	0.01	150
Total 80/20 split	4	0.1	0.6	0.01	150
Total 90/10 split	4	0.1	0.6	0.01	150

Table 4.2: Best hyperparameter values, boosting models

4.2 Model performance comparison, artificial

With the determined best values, models are created with the training data and evaluated based on the test data (artificial performance). It is important to mention that for the low and high group models of the boosting technique, the hyperparameter best values are slightly modified. The best eta value determined by the hyperparameter tuning was 0.01 for these models. Nevertheless, this resulted in an accuracy of around 3% lower compared to using an eta value of 0.1. An eta value of 0.1 was the best value for the total models. Therefore, the eta value is slightly changed for the low and high group models to increase the performance. Table 4.3 contains the accuracy of all models where each specific model is compared between the random forest and boosting techniques, as it uses the same data for training and testing. The chi-square test shows that for all comparisons except for the 90/10 split the boosting model has a significant better accuracy. In other words, the boosting models are often better at classifying stable and unstable periods. A complete overview of the model performance including other metrics can be found in Appendix B. These performance metrics demonstrated in the appendix indicate that the sensitivity and precision are quite high (>= 83%). Nevertheless, the negative predictive value has a range of about 20% to 60%. This indicates that not too many unstable classifications are indeed unstable determined by the artificial labeling.

Model	**Accuracy**		**Chi-square**
	Random forest	**Boosting**	**p-value**
Low group	82.2%	83.1%	< 0.01
High group	85.6%	86.7%	< 0.001
Total 80/20 split	83.1%	84.8%	< 0.001
Total 90/10 split	91.1%	91.3%	0.27

Table 4.3: Accuracy comparison, artificial labels

4.3 Model performance comparison, clinical

The artificial model comparison determined how well the artificial labels are classified. However, it is not possible to capture the clinical performance and relevance with those results. The models presented in the previous section are all retrained on the complete dataset to make a final classification on a left out dataset. The variable importance plots and partial dependence plots of these final models can all be found in Appendix C. The visualisations indicate that in general respiratory rate is the most important vital sign followed by heart rate, invasive mean blood pressure, oxygen saturation, and regional cerebral oxygen saturation. Nevertheless, there were some models where oxygen saturation was the second most important variable and invasive mean blood pressure the least important variable. Overall, the respiratory rate and the least important vital sign had an importance value of approximately 0.3 and 0.13, respectively. Generally, this indicates that all vital signs contribute to classifying stable and unstable periods.

The new dataset contains nine patients where patient numbers 1, 3 and 8 fall within the high group and the others in the low group. This left out dataset does not consist of true stable and unstable labels. Therefore, the classification is compared to the expert's evaluation. The classification that is compared to is smoothed to avoid sudden fluctuations which are commonly present in patients at the PICU. For instance, five unstable observations spread over an hour of stable observations should all be classified as stable. An example of the vital signs distributed over time, the classifications made by the models and the clinical classification can be found in Figure 4.1. The clinical comparison can be divided into three categories: Models that consider all patients, models that only consider patients within the low group and models that only consider patients within the high group. The comparison focuses on four metrics: sensitivity, specificity, accuracy and the F1-score. A full overview of each model's confusion matrix and performance metrics can be found in Appendix D.

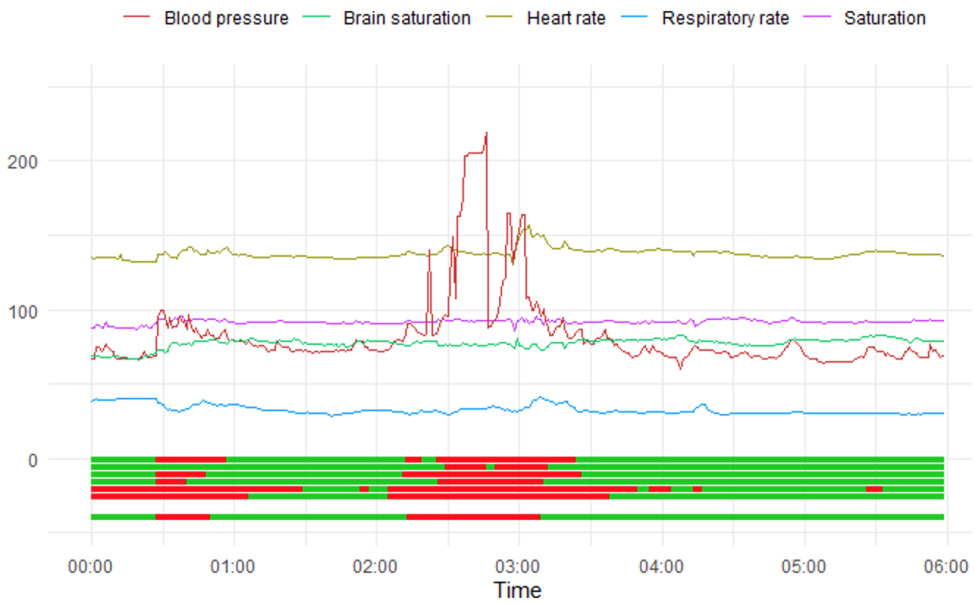


Figure 4.1: Window of six hours with smoothed classifications and vital signs. The red and green bars indicate unstable and stable periods, respectively.

Table 4.4 illustrates the clinical performance considering all patients and contains several findings. First, the sensitivity and specificity metrics are closer to each other for the 80/20 split compared to the 90/10 split. In other words, the percentage of true stable observations classified as stable is similar to the percentage of true unstable observations classified as unstable. The 90/10 split models come close to classifying all true stable observations as stable, but classify less than half true unstable observations as unstable. The accuracy shows that the different techniques perform similarly considering the same split, but the 90/10 models are better compared to the 80/20 models. Nevertheless, the F1-scores indicate that the boosting models are performing better with very little difference between the 80/20 and 90/10 split.

Model	**Sensitivity**	**Specificity**	**Accuracy**	**F1**
Random forest 80/20	77.4%	72.9%	76.5%	84.0%
Random forest 90/10	94.8%	48.2%	85.4%	83.7%
Boosting 80/20	78.6%	63.7%	75.5%	91.2%
Boosting 90/10	95.7%	41.4%	85.0%	91.1%

Table 4.4: Clinical performance, all patients

Table 4.5 demonstrates the clinical performance considering only patients within the low group. The different models show quite some variation in the sensitivity and specificity values. The table indicates that from both specific low models, the boosting technique is performing better than the random forest. The boosting model was able to increase the sensitivity and keep the decrease of specificity to a minimum. The boosting model also resulted in the highest accuracy excluding the 90/10 split options. The 90/10 split options resulted in a higher accuracy, but achieved this by mostly classifying all true stable observations as stable. Very few true unstable observations were classified as unstable.

Model	**Sensitivity**	**Specificity**	**Accuracy**	**F1**
Random forest low	66.0%	57.4%	64.4%	75.3%
Boosting low	82.7%	57.2%	78.2%	86.1%
Random forest 80/20	82.2%	58.2%	77.9%	85.9%
Random forest 90/10	97.0%	34.2%	85.7%	91.8%
Boosting 80/20	81.8%	44.2%	75.0%	84.3%
Boosting 90/10	97.3%	23.1%	83.9%	90.8%

Table 4.5: Clinical performance, low group

Table 4.6 demonstrates the clinical performance considering only patients within the high group. Again variations can be detected in the sensitivity and specificity. However, the metric values are in general better compared to the low group and the total models. Furthermore, the high group models are the only models to achieve a higher specificity score compared to the sensitivity scores excluding the 90/10 split models. The increase in specificity compared to other group models did not decrease the sensitivity to much as well. In general, it seems that the high models are better at creating a good balance in correctly classifying both classes. The highest accuracies are achieved, equal to other groups, by the 90/10 split models. Nevertheless, the boosting model specific for the high group has an accuracy above 80%, with good numbers for both the sensitivity and specificity. The highest F1-scores are for the 90/10 split models, but followed by the specific high group boosting model.

Model	**Sensitivity**	**Specificity**	**Accuracy**	**F1**
Random forest high	71.1%	87.4%	75.1%	81.2%
Boosting high	77.5%	88.2%	80.1%	85.5%
Random forest 80/20	67.0%	94.3%	73.7%	79.4%
Random forest 90/10	90.0%	68.9%	84.8%	90.0%
Boosting 80/20	71.5%	92.1%	76.6%	82.1%
Boosting 90/10	93.2%	68.3%	87.1%	91.6%

Table 4.6: Clinical performance, high group

5. Discussion and Conclusions

This chapter discusses the findings of the study and compares them to related work. The discussion of the findings is followed by the limitations and recommendations. With all the information, conclusions are formulated and the research question is answered.

5.1 Discussion

Study findings

Within this study, eight different models were developed and tested on both their artificial and clinical performance. These eight models were trained using both random forest and boosting techniques, resulting in four models each. The artificial performance describes the performance of the models when trained and tested on the artificially created labels, while the clinical performance indicates how well the trained models on artificially created labels can classify stable and unstable labels compared to a clinical classification by a medical expert. The results of the artificial performance indicated that the boosting models were better at classification, with a significant difference in three out of four comparisons. This difference only concludes that the boosting models in general can classify the artificial labels slightly more accurate. However, these numbers do not give any indication on how well the classifications are compared to a true clinical classification.

The clinical performance comparison indicated that in general accurate classifications can be made by the different models. The comparison also showed a wide range of variation in sensitivity and specificity values within and between groups. The highest accuracies were achieved by both the boosting and random forest 90/10 models [83.9%-87.1%]. However, the sensitivity values [90%-97.3%] were much higher compared to the specificity values [23.1%-68.9%].

This is probably related to the imbalanced dataset. The model with a specificity of 23.1% would not capture a large number of unstable periods. A nurse or doctor relying only on such a model would miss many critical situations that could have dangerous consequences. Luckily, monitors around the bed of a patients capture high and low vital sign values, but using a model like this should closely be considered. Nevertheless, a low specificity value is often associated with a high sensitivity value. In case the aim of a model is to capture only stable periods as well as possible, a 90/10 model is very useful. For instance, a hospital that relies entirely on monitor alarms (e.g. an alarm is triggered when the heart rate exceeds a certain threshold) and only wants to know when a patient is stable may want to use such a model. Nevertheless, in general it would probably be more valuable to accurately classify both stable and unstable periods, or at least not miss any unstable periods.

This study also compared group specific models with models trained on all data and applied to the same specific group. The findings demonstrate that the specific random forest model did not outperform the total models. The boosting specific group models, on the other hand, were able to perform better than the 80/20 split models based on accuracy. The boosting low model was also able to perform better considering the sensitivity and specificity, while the high group was better considering the sensitivity. This indicates that it is preferred to have a group specific boosting model compared to a group specific random forest model and a total model with an 80/20 split. The specific group models also showed more balanced sensitivity and specificity values, which can be a possible requirement for future adaptation of machine learning models in healthcare.

The findings of this study can also be compared to previous work including a SVM model, since one of the goals of this study was to provide insights in its performance compared to other models [3]. It should be mentioned that it is difficult to compare both studies as the performance is determined based on a combination of three models. The three models are a SVM model, a measurement error detection model, and a baseline variation model. Both the

SVM model and baseline variation model classify observations as stable or unstable. The measurement error detection model determines whether one or more observations were collected during a measurement error and gives those observations a corresponding label. In other words, it is not possible to compare the outcomes directly, but might give an indication of what is possible. The SVM model resulted for both groups in sensitivity and specificity scores of 93% and 77%, respectively. The findings of this study indicate lower metric scores considering the best models. The sensitivity score could be matched with the 90/10 models on all patients, but have much lower specificity scores. Nevertheless, the remaining unstable periods might be captured by the baseline variation model stated in the previous work. The SVM model specific for the low group resulted in a sensitivity and specificity of 92% and 63%, respectively. The specific boosting low group and random forest 80/20 model are the models closest to the SVM according to the metrics. The sensitivity is about 9.5% lower where the specificity is about 5% lower. These percentages might also go up when including the additional models of the previous work. At last, the SVM specific for the high group resulted in a sensitivity and specificity of 93% and 83%, respectively. The specific boosting high group model would outperform the SVM on the specificity by 5%, but perform less on the sensitivity with 16%. The boosting 90/10 model would perform similar on the sensitivity, but score 15% less on the specificity. To summarize, the combination of models in previous work has better metric values compared to the models in this study. Nevertheless, some models in this study have the potential to come close to or perhaps exceed the metrics when combined with the other models mentioned in the previous work under the assumption that this would improve the performance.

Related work

A study by Kanbar et al. also developed a machine learning model for identifying epilepsy surgical candidates and achieved a sensitivity of 77%. Even though the specific application of the model is different, the sensitivity of the created models in this study were often equal or higher. Unfortunately, this study did not describe their specificity value. The same study also determined

that their model was performing similarly to board-certified neurologist. This indicates that even professionals make mistakes and models will most likely also make mistakes. In other words, a model with a metric below 100% is not immediately unusable in practice. Another study by Ruiz et al. aimed to predict deterioration in advance (e.g. four hours). Their model, using extreme gradient boosting, resulted in a sensitivity of 88% and a specificity of 86%. The application of the model is again a bit different, but has higher metric values compared to the best models of this study. This indicates that there are probably still areas to improve on. Nevertheless, some models of this study were not to far in achieving similar results. It should be mentioned that the study by Ruiz et al. consisted of 488 patients and over 1,000 possible variables. The difference in these numbers compared to this study can have a big impact on the classification metrics and improve the models.

A study conducted by Pirneskoski et al. aimed to compare a random forest model to the national early warning system (NEWS) based on the mortality prediction accuracy. The random forest outperformed the NEWS model indicating that machine learning techniques are capable of outperforming more traditional methods. The sensitivity and specificity of the study were not reported, but could be inferred from the area under the curve plot. The random forest model created by Pirneskoski et al. was able to achieve a sensitivity and specificity of approximately 80% and 70%, respectively. The plot also indicates that a sensitivity of around 90% would result in a specificity of around 45%. These metric values are similar to the values of this study and might conclude that the created models of this study also have the ability to outperform more traditional methods. At last, two other studies explored the use of deep learning for predicting mortality a certain number of hours in advance. The area under the curve values of both studies resulted in high scores, thus were accurate in predicting mortality in advance [19] [20]. Aczon et al. predicted mortality with an area under the receiver operating characteristic curve of 94%. Kim et al. used the same metric for their models and gave a range of 89% to 97%. These metrics indicate that with enough data, deep learning models can make very accurate predictions within healthcare. These metrics

also show the best performance compared to all the studies discussed in this section. However, these deep learning models lack interpretability, which is currently essential for an implementation into the workflow.

Overall, different studies indicate their potential in using machine learning techniques to add value to the workflow within healthcare. The deep learning models showed great potential, but also lacked the ability to provide insights into what is happening within the model, thus its decision making. It appears that the models created in this study within ten weeks are comparable to other created models, but in general consisted of slightly lower sensitivity and/or specificity scores. The models used within the other studies were trained on more data and included additional variables probably resulting in these differences. Another reason for the difference might be the use of real labels, as the studies did not indicate that artificial labels were created. Nevertheless, combining the best models of this study, taking into account accuracy as well as interpretability, with the other additional models created by previous work would most likely result in a better performance [3]. This would slowly bridge the gap from being a machine learning technique to an implementation into the workflow.

Limitations

In addition to the findings and strengths of this study, several limitations should be mentioned. First, the study filters patients based on the availability of vital signs such as regional cerebral oxygen saturation which is currently unavailable as a standard of care resulting in selection bias [3]. Furthermore, the frequency of data collected is once per minute, which is the frequency at which data is written from the monitors to the patient database management system. The flow of data to the database is done through re-sampling resulting in some loss of data as some vital signs are measured more than once every minute. To summarize, decisions are made based on the usability of the applications currently available, resulting in a possible reduction in data quality.

Secondly, the study is conducted within a time period of ten weeks. These weeks gave the opportunity to explore multiple options to execute the research and add value to the field of interest. Nevertheless, the depth of the research has its limits. Even though hyperparameters were tuned and captured a wide range of possible options, it could have been executed more extensively. In the end, it is very difficult to state that the hyperparameter values of the final models are indeed the best options, since many options were not considered. Furthermore, the models from the hyperparameter tuning have been compared based on the accuracy. The findings of this study indicated that the accuracy can change depending on which model is considered. Moreover, an equal accuracy between two models can have different sensitivity and specificity values. This study focused on a general classification performance, thus accuracy was used for comparison. Nevertheless, if the aim would be to train models that accurately classify unstable periods, it would be better to use the specificity to compare the models during hyperparameter tuning. In short, the metric used to compare the models during hyperparameter tuning should be carefully considered by determining the clinical aim.

At last, predictions were made on a left out dataset of nine patients. From these nine patients, a random window of 24 hours was selected and used to compare it to a clinical classification annotated by one medical expert. It is possible, due to the low number of true testing labels and the annotations of one medical expert, the results are less robust. For instance, the high group contained three patients. A random window with only stable or unstable observations can have a big influence on the final performance.

Recommendations

For future research, it is recommended to further explore the models and also determine if the models can be improved. A possible improvement might be achieved by performing a more extensive hyperparameter tuning. The hyperparameter tuning in this study covered a wide range of options, but small improvements might be possible by testing small changes to the best values from this study. Furthermore, the study uses a threshold of 0.5 as boundary

between stable and unstable periods. Adjusting this threshold can result in a better performance or shift the trade-off between the sensitivity and specificity. The use of percentages can also be extended by classifying the periods into more groups or only use the percentage and visualize these percentages with a color gradient scale. Especially the focus on percentages can be interesting for healthcare, where you often work with certain risks of for example being unstable.

Furthermore, it is recommended to explore the options to obtain more training data. Machine learning models often give better performance when trained on more data. The training set for the high group for example contained more data compared to the low group and resulted in a better performance on the test data. The data used also contained a class imbalance which influences the classification models. This class imbalance cannot be prevented. However, options exist that create samples from real data to increase in this case the number of observations for the unstable class. At last, testing is performed on a relatively small clinically labeled dataset. It would be recommended to have more clinical labels to give additional value and increase the reliability of the clinical comparison.

5.2 Conclusions

The aim of this study was to develop several models using the random forest and boosting techniques that would accurately detect deterioration by classifying stable and unstable periods in infants at the PICU with cCHD over time. The models indicated some promising results where certain models were performing better than others. Especially, the boosting specific low and high group models showed potential. In addition, models trained on all the data without separating on a group also resulted in good performance metrics. It can be helpful to have a general model that can be used for all patients, since you might not have the information about which group a patient belongs to.

The performance metrics of this study showed lower scores compared to the SVM model of Zoodsma et al. However, the performance of the SVM model also included additional models to classify deterioration that were not used within this study. It is assumed that the addition of those models to the random forest and boosting models would enhance the metrics and result in similar results compared to the SVM. Furthermore, the SVM model by Zoodsma et al. is not able to give percentages, while the random forest and boosting models can. These percentages make it possible to move from a dichotomous classification to percentages that indicate a risk of deterioration. In other words, there will be a difference between a 55% and a 95% probability of being unstable, which can be of great importance for clinical decision making in the clinical workflow.

Bibliography

[1] T. van der Blom, A. C. Zomer, A. H. Zwinderman, F. J. Meijboom, B. J. Bouma, and B. J. M. Mulder, "The changing epidemiology of congenital heart disease," *Nature Reviews Cardiology*, vol. 8, Nov. 2010. DOI: 10.1038/nrcardio.2010.166.

[2] R. Sun, M. Lui, L. Lu, Y. Zheng, and P. Zhang, "Congenital heart disease: Causes, diagnosis, symptoms, and treatments," *Cell Biochemistry and Biophysics*, vol. 72, Feb. 2015. DOI: 10.1007/s12013-015-0551-6.

[3] R. S. Zoodsma, R. Bosch, T. Alderliesten, *et al.*, "Continuous data-driven monitoring in critical congenital heart disease: Clinical deterioration model development," 2023.

[4] D. A. Clifton, D. Wong, L. Clifton, *et al.*, "A large-scale clinical validation of an integrated monitoring system in the emergency department," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, Jul. 2013. DOI: 10.1109/JBHI.2012.2234130.

[5] L. J. Kanbar, B. Wissel, Y. Ni, *et al.*, "Implementation of machine learning pipelines for clinical practice: Development and validation study," *JMIR Medical Informatics*, vol. 10, no. 12, Dec. 2022. DOI: 10.2196/37833.

[6] M. van Smeden, G. Heinze, B. van Calster, *et al.*, "Critical appraisal of artificial intelligence-based prediction models for cardiovascular disease," *European Heart Journal*, vol. 43, no. 31, Apr. 2022. DOI: 10.1093/eurheartj/ehac238.

[7] M. Oyeleye, T. Chen, S. Titarenko, and G. Antoniou, "A predictive analysis of heart rates using machine learning techniques," *Int J Environ Res Public Health*, vol. 19, no. 4, Feb. 2022. DOI: 10.3390/ijerph19042417.

[8] N. Shah, A. Arshad, M. B. Mazer, C. L. Carroll, S. L. Shein, and K. E. Remy, "The use of machine learning and artificial intelligence within pediatric critical care," *Pediatric Research*, vol. 93, Nov. 2022. DOI: 10.1038/s41390-022-02380-6.

[9] V. M. Ruiz, M. P. Goldsmith, L. Shi, *et al.*, "Early prediction of clinical deterioration using data-driven machine-learning modeling of electronic health records," *Journal Thorac Cardiovasc Surg.*, vol. 164, no. 1, Jul. 2022. DOI: 10.1016/j.jtcvs.2021.10.060.

[10] S. Muralitharan, W. Nelson, S. Di, *et al.*, "Machine learning-based early warning systems for clinical deterioration: Systematic scoping review," *Journal Med Internet Res.*, vol. 23, no. 2, Feb. 2022. DOI: 10.2196/25187.

[11] J. Pirneskoski, J. Tamminen, A. Kallonen, *et al.*, "Random forest machine learning method outperforms prehospital national early warning score for predicting one-day mortality: A retrospective study," *Journal Resuscitation Plus*, vol. 4, Dec. 2020. DOI: 10.1016/j.resplu.2020.100046.

[12] PICU Datalab, *Applied Data Science Thesis Code*, http://picudatalab.com/ADS_thesis/.

[13] R. D. Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, Jan. 2000. DOI: 10.1016/S0169-7439(99)00047-7.

[14] F. E. Harrell, *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer, 2015.

[15] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *ICT Discoveries*, vol. 1, Oct. 2017.

[16] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed. Springer, 2021.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, Oct. 2001. DOI: 10.1023/A:1010933404324.

[18] S. Sendelbach and M. Funk, "Alarm fatigue: A patient safety concern," *AACN Adv Crit Care*, vol. 24, no. 4, Oct. 2013. DOI: 10.1097/NCI.0b013e3182a903f9.

[19] S. Y. Kim, S. Kim, J. Cho, *et al.*, "A deep learning model for real-time mortality prediction in critically ill children," *Critical Care*, vol. 23, no. 279, Aug. 2019. DOI: 10.1186/s13054-019-2561-z.

[20] M. D. Aczon, D. R. Ledbetter, E. Laksana, L. V. Ho, and R. C. Wetzel, "Continuous prediction of mortality in the picu: A recurrent neural network model in a single-center dataset," *Pediatric Critical care medicine*, vol. 22, no. 6, Jun. 2021. DOI: 10.1097/PCC.0000000000002682.

Appendices

A. Stable and unstable data distributions

This appendix consists of all distributions of the data separated on stable and unstable periods.

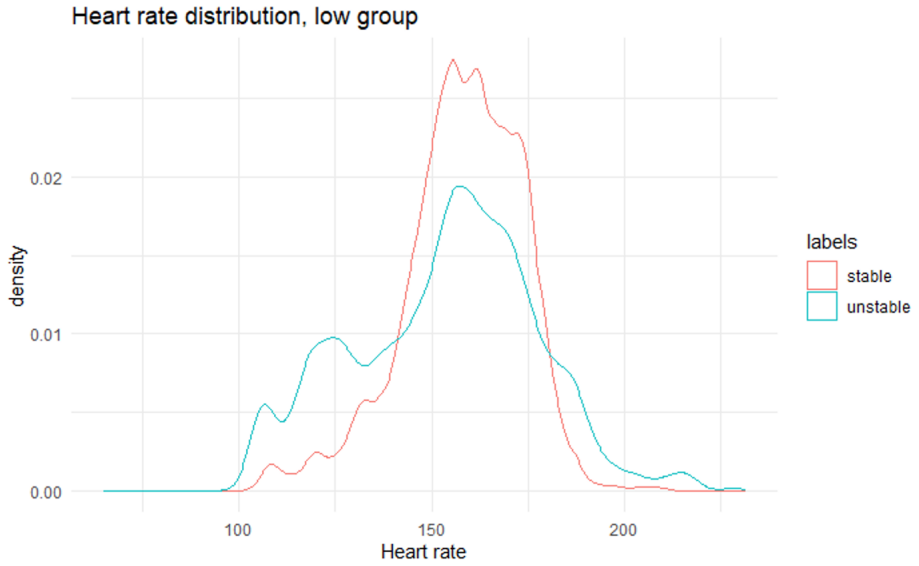


Figure A.1: Heart rate distribution, low group

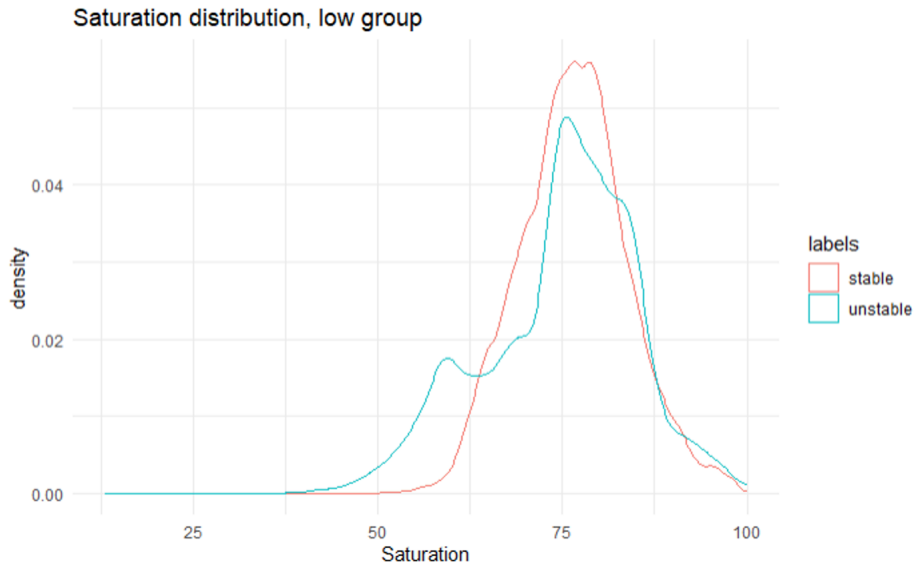


Figure A.2: Saturation distribution, low group

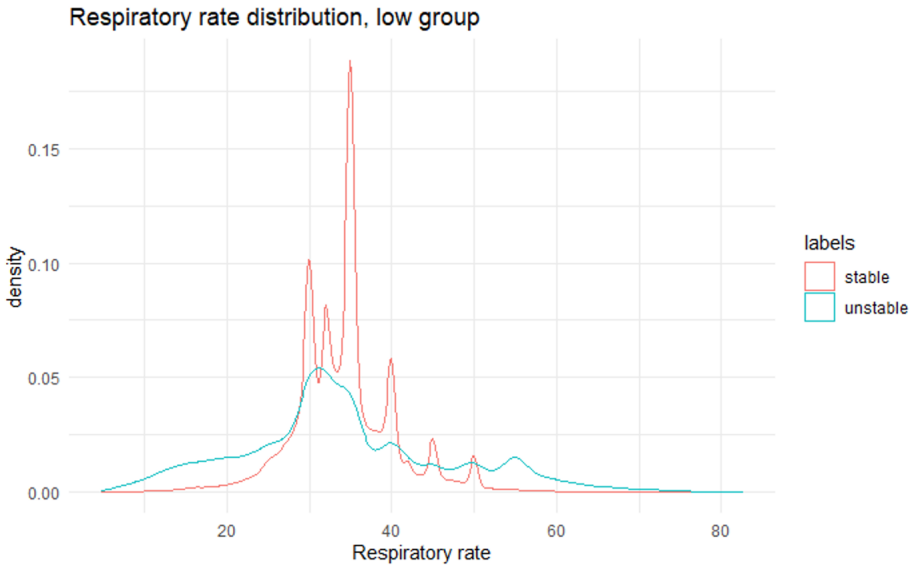


Figure A.3: Respiratory rate distribution, low group

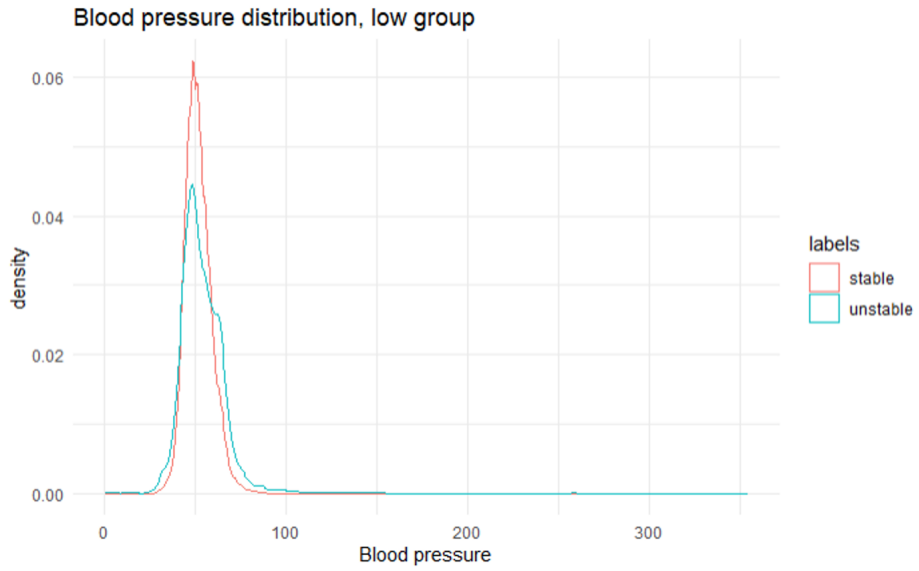


Figure A.4: Blood pressure distribution, low group

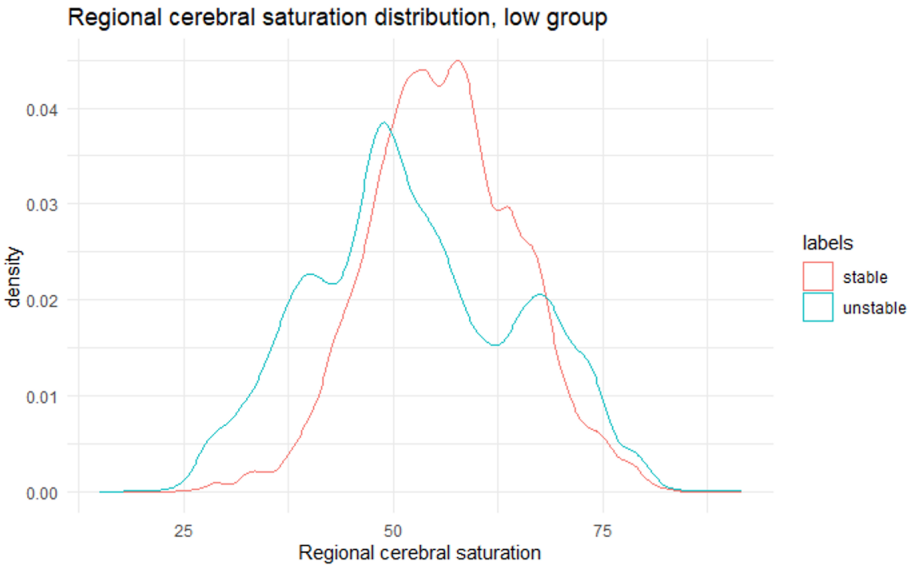


Figure A.5: Regional cerebral saturation distribution, low group

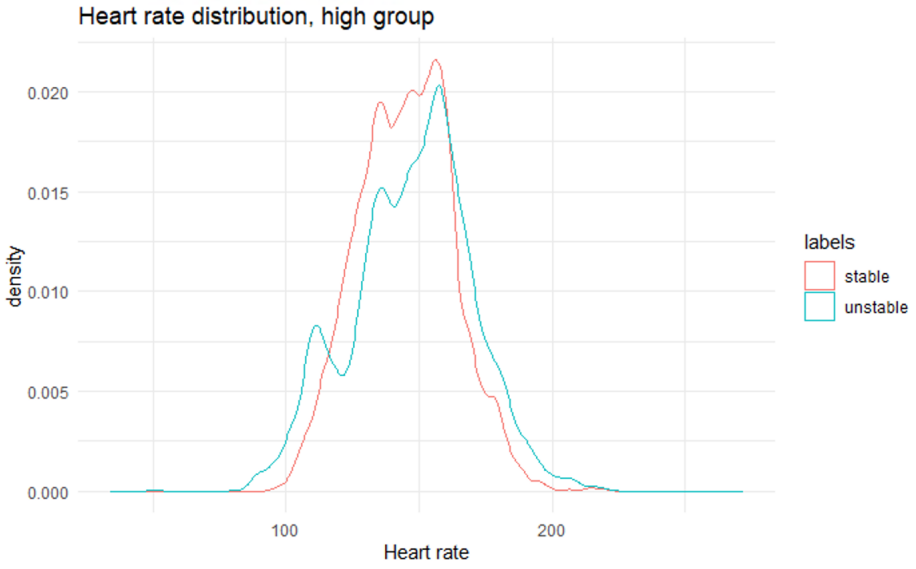


Figure A.6: Heart rate distribution, high group

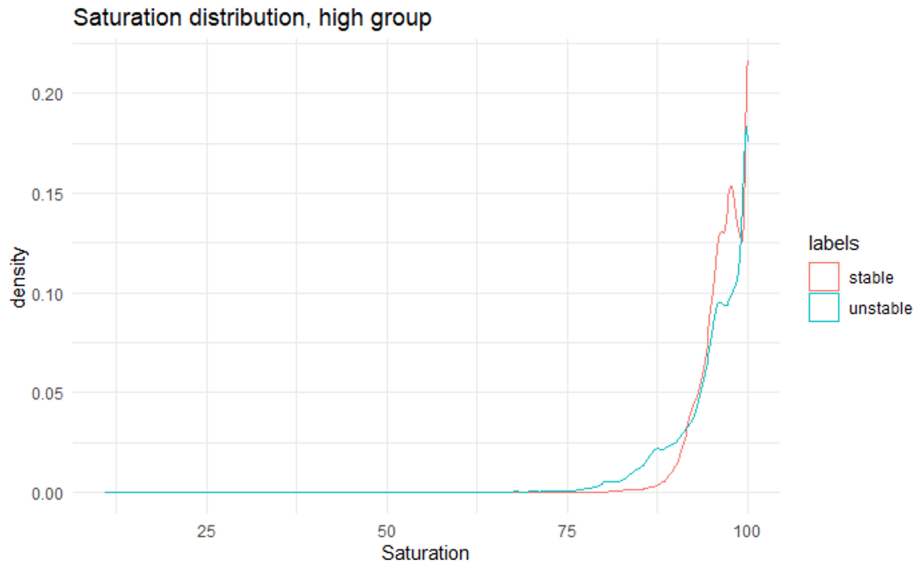


Figure A.7: Saturation distribution, high group

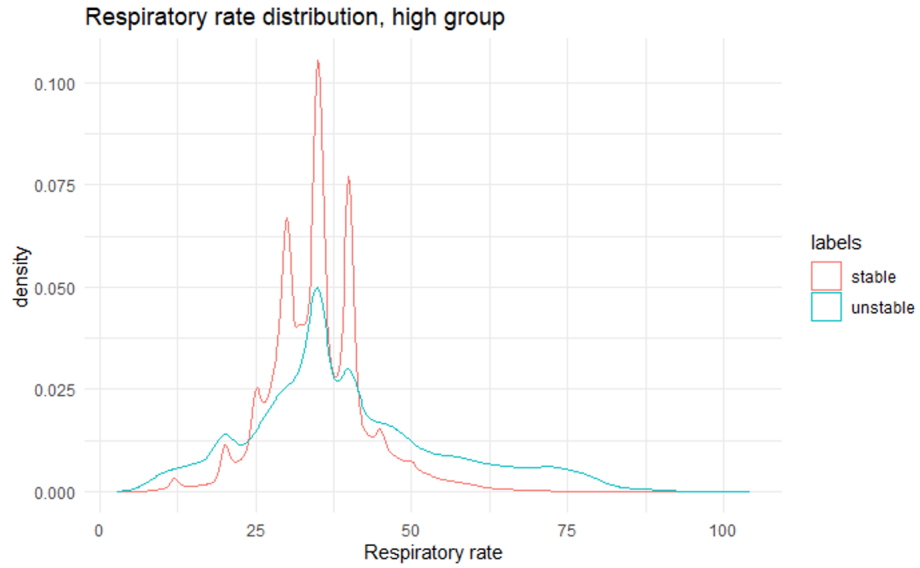


Figure A.8: Respiratory rate distribution, high group

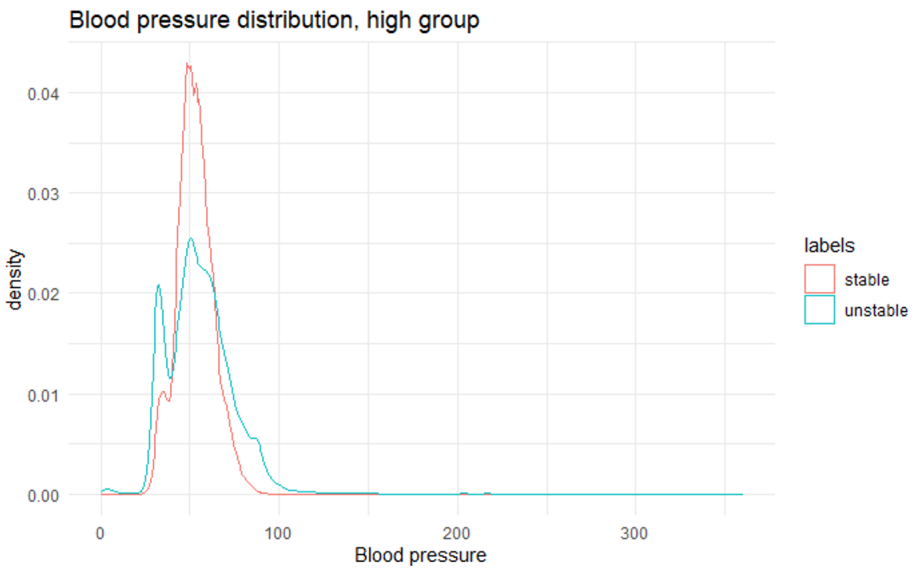


Figure A.9: Blood pressure distribution, high group

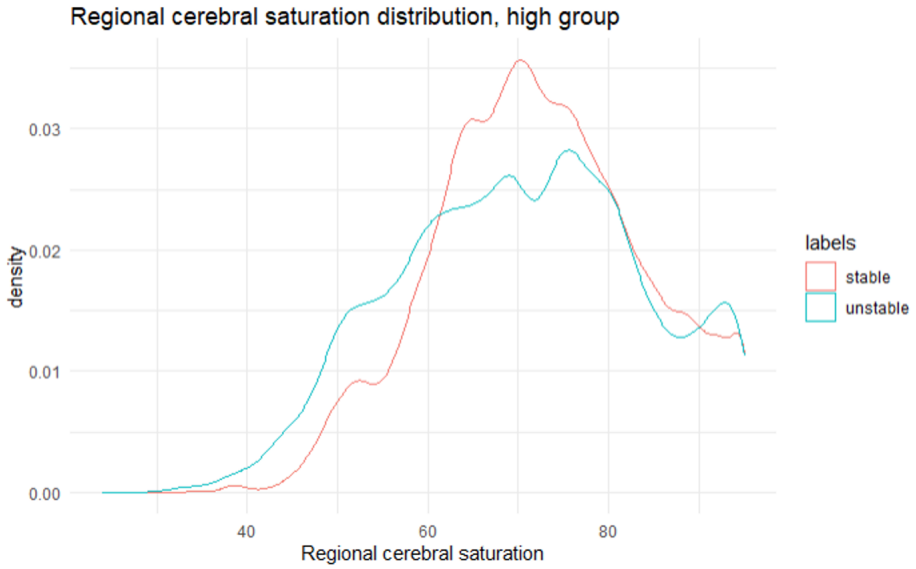


Figure A.10: Regional cerebral saturation distribution, high group

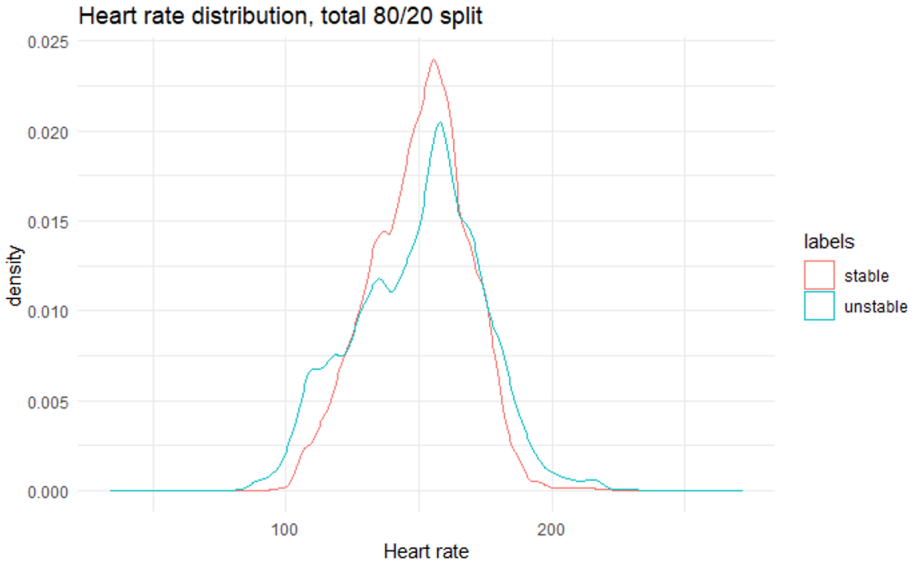


Figure A.11: Heart rate distribution, total 80/20 split

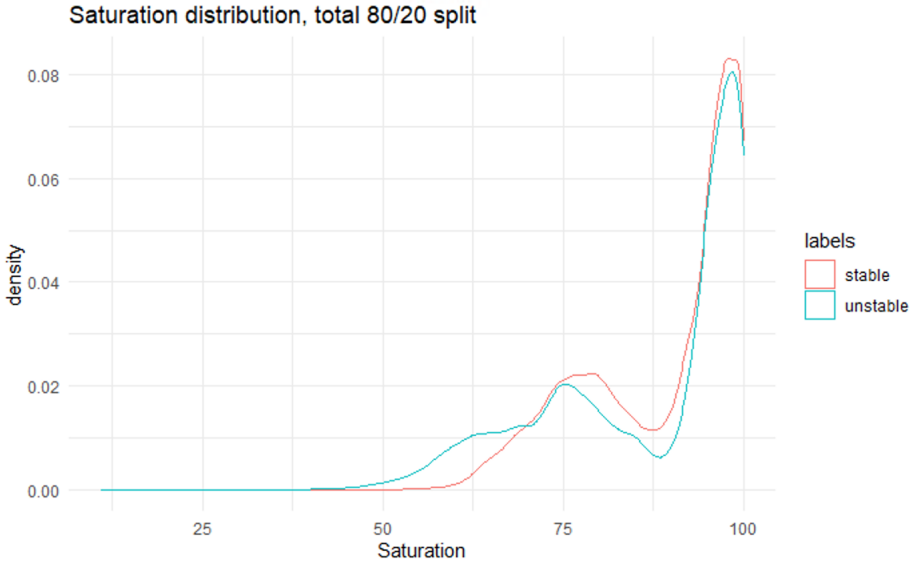


Figure A.12: Saturation distribution, total 80/20 split

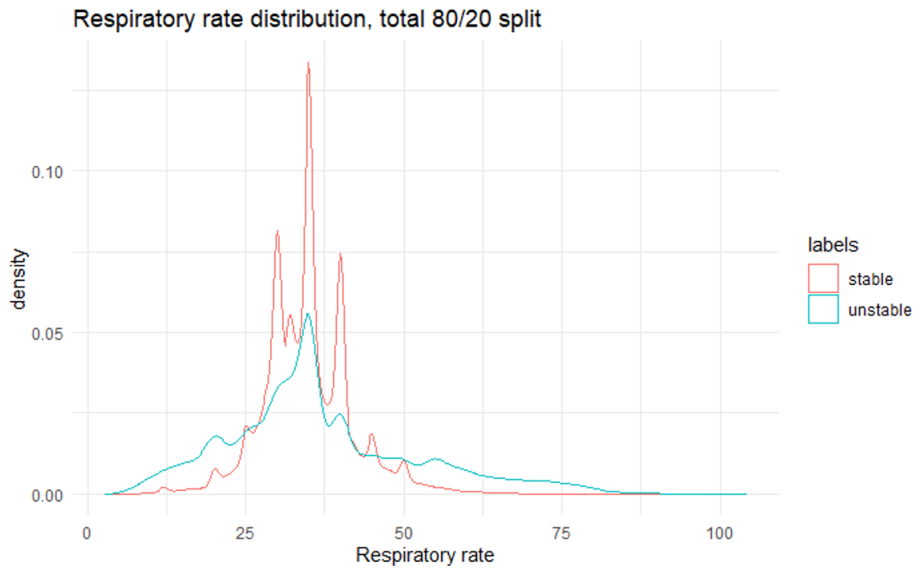


Figure A.13: Respiratory rate distribution, total 80/20 split

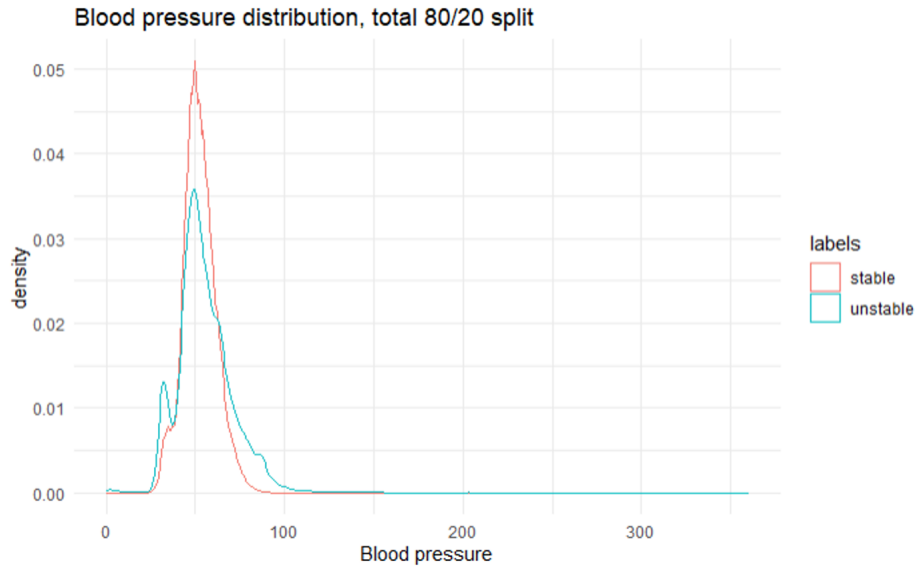


Figure A.14: Blood pressure distribution, total 80/20 split

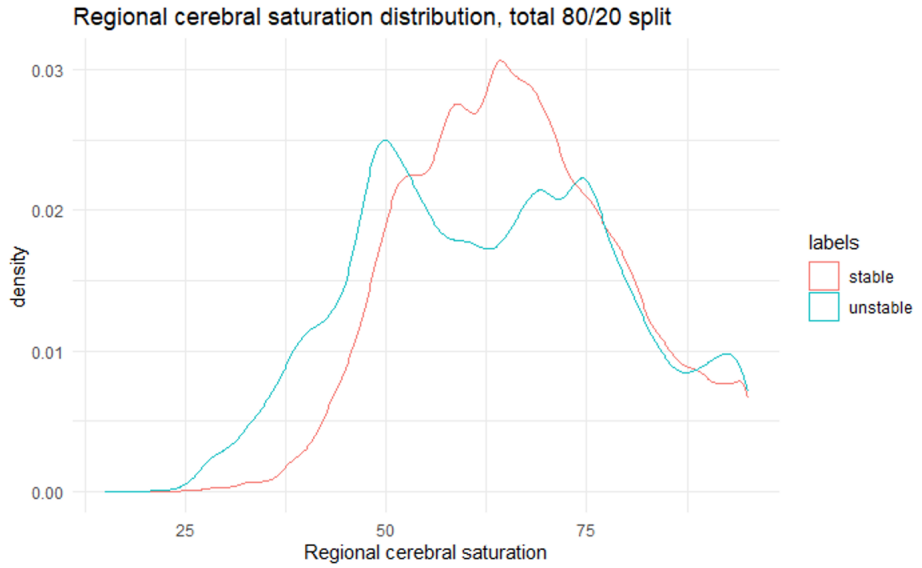


Figure A.15: Regional cerebral saturation distribution, total 80/20 split

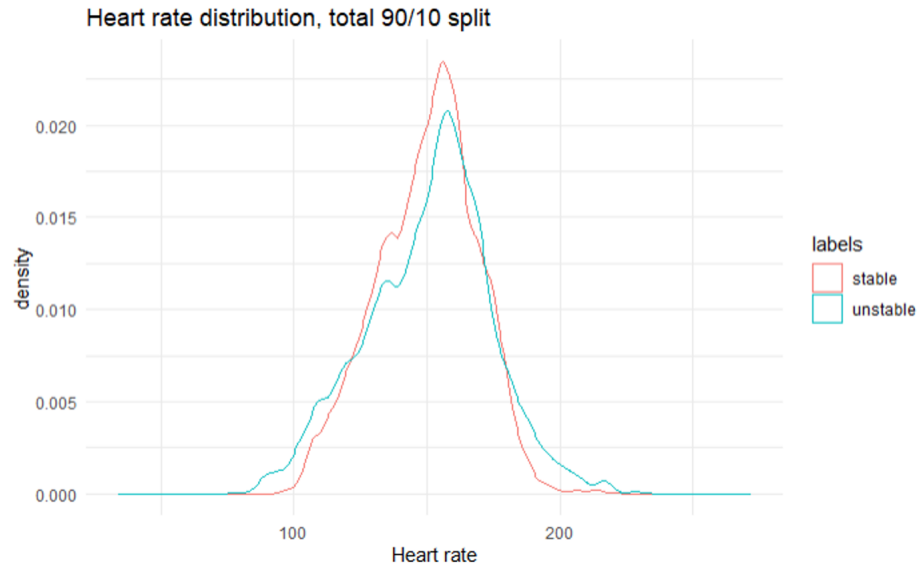


Figure A.16: Heart rate distribution, total 90/10 split

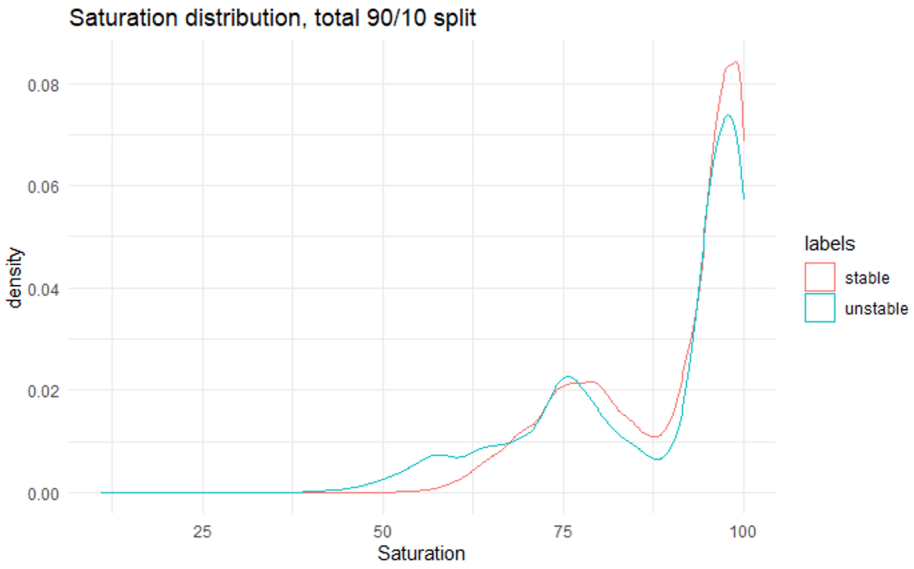


Figure A.17: Saturation distribution, total 90/10 split

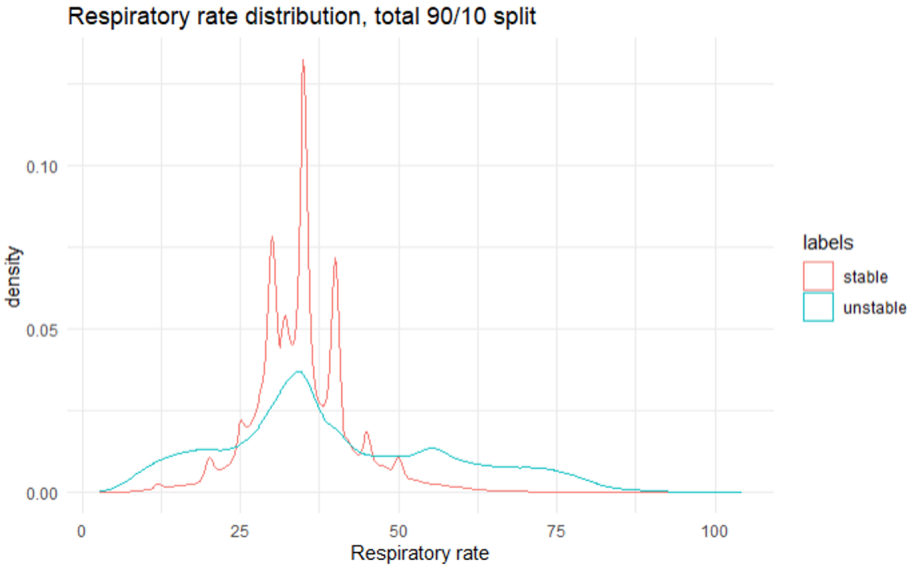


Figure A.18: Respiratory rate distribution, total 90/10 split

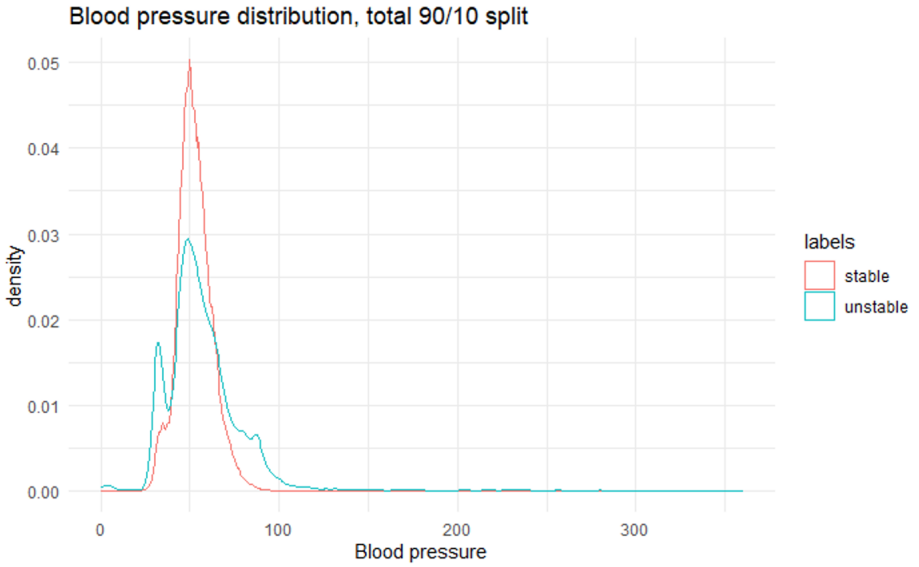


Figure A.19: Blood pressure distribution, total 90/10 split

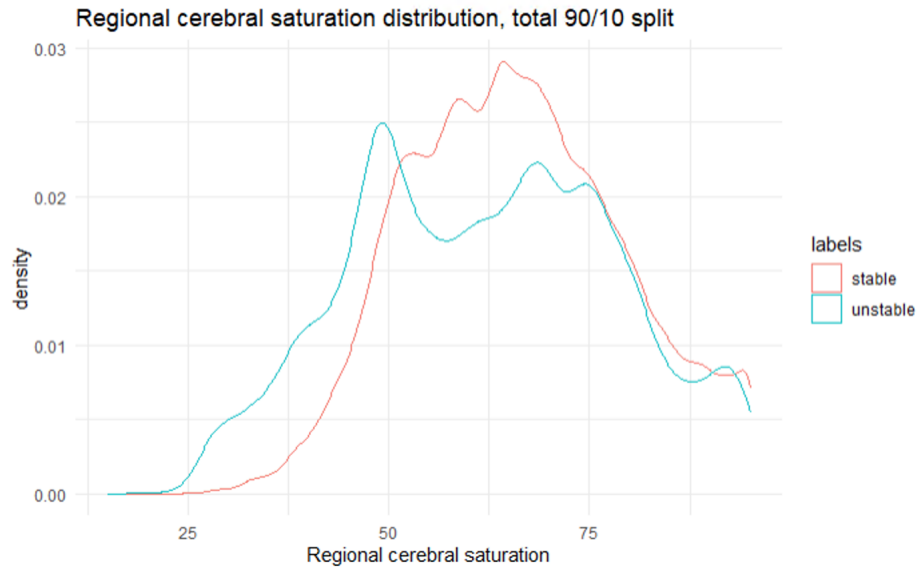


Figure A.20: Regional cerebral saturation distribution, total 90/10 split

B. Artificial performance

This appendix visualises the confusion matrices and their performance metrics for the artificial performance.

Random forest

This section contains the confusion matrices and their performance metrics for the random forest models.

```
          Reference
Prediction stable unstable
  stable    26385      744
  unstable   5324     1670

               Accuracy : 0.8222
                 95% CI : (0.8181, 0.8262)
    No Information Rate : 0.9293
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2792

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8321
            Specificity : 0.6918
         Pos Pred Value : 0.9726
         Neg Pred Value : 0.2388
             Prevalence : 0.9293
         Detection Rate : 0.7732
   Detection Prevalence : 0.7950
      Balanced Accuracy : 0.7619

       'Positive' Class : stable
```

Figure B.1: Performance metrics, low group

```
          Reference
Prediction stable unstable
  stable    32375     2612
  unstable   3430     3408

               Accuracy : 0.8555
                 95% CI : (0.8521, 0.8589)
    No Information Rate : 0.8561
    P-Value [Acc > NIR] : 0.6236

                  Kappa : 0.4452

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9042
            Specificity : 0.5661
         Pos Pred Value : 0.9253
         Neg Pred Value : 0.4984
             Prevalence : 0.8561
         Detection Rate : 0.7741
   Detection Prevalence : 0.8365
      Balanced Accuracy : 0.7352

       'Positive' Class : stable
```

Figure B.2: Performance metrics, high group

```
          Reference
Prediction stable unstable
  stable    52184     6369
  unstable   6006     8798

               Accuracy : 0.8313
                 95% CI : (0.8286, 0.834)
    No Information Rate : 0.7932
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4811

 Mcnemar's Test P-Value : 0.001137

            Sensitivity : 0.8968
            Specificity : 0.5801
         Pos Pred Value : 0.8912
         Neg Pred Value : 0.5943
             Prevalence : 0.7932
         Detection Rate : 0.7114
   Detection Prevalence : 0.7982
      Balanced Accuracy : 0.7384

       'Positive' Class : stable
```

Figure B.3: Performance metrics, total 80/20 split

```
          Reference
Prediction stable unstable
  stable    61839     3506
  unstable   3002     5010

               Accuracy : 0.9113
                 95% CI : (0.9092, 0.9133)
    No Information Rate : 0.8839
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5563

 Mcnemar's Test P-Value : 4.515e-10

            Sensitivity : 0.9537
            Specificity : 0.5883
         Pos Pred Value : 0.9463
         Neg Pred Value : 0.6253
             Prevalence : 0.8839
         Detection Rate : 0.8430
   Detection Prevalence : 0.8908
      Balanced Accuracy : 0.7710

       'Positive' Class : stable
```

Figure B.4: Performance metrics, 90/10 split

Boosting

This section contains the confusion matrices and their performance metrics for the boosting models.

```
          Reference
Prediction stable unstable
  stable    26887      242
  unstable   5539     1455

               Accuracy : 0.8306
                 95% CI : (0.8266, 0.8345)
    No Information Rate : 0.9503
    P-Value [Acc > NIR] : 1

                  Kappa : 0.277

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8292
            Specificity : 0.8574
         Pos Pred Value : 0.9911
         Neg Pred Value : 0.2080
             Prevalence : 0.9503
         Detection Rate : 0.7879
   Detection Prevalence : 0.7950
      Balanced Accuracy : 0.8433

       'Positive' Class : stable
```

Figure B.5: Performance metrics, low group

```
          Reference
Prediction stable unstable
  stable    32819     2168
  unstable   3380     3458

               Accuracy : 0.8674
                 95% CI : (0.8641, 0.8706)
    No Information Rate : 0.8655
    P-Value [Acc > NIR] : 0.1333

                  Kappa : 0.4778

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9066
            Specificity : 0.6146
         Pos Pred Value : 0.9380
         Neg Pred Value : 0.5057
             Prevalence : 0.8655
         Detection Rate : 0.7847
   Detection Prevalence : 0.8365
      Balanced Accuracy : 0.7606

       'Positive' Class : stable
```

Figure B.6: Performance metrics, high group

```
          Reference
Prediction stable unstable
  stable    53000     5553
  unstable   5616     9188

               Accuracy : 0.8477
                 95% CI : (0.8451, 0.8503)
    No Information Rate : 0.7991
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5266

 Mcnemar's Test P-Value : 0.5574

            Sensitivity : 0.9042
            Specificity : 0.6233
         Pos Pred Value : 0.9052
         Neg Pred Value : 0.6206
             Prevalence : 0.7991
         Detection Rate : 0.7225
   Detection Prevalence : 0.7982
      Balanced Accuracy : 0.7637

       'Positive' Class : stable
```

Figure B.7: Performance metrics, total 80/20 split

```
          Reference
Prediction stable unstable
  stable    61710     3635
  unstable   2751     5261

               Accuracy : 0.9129
                 95% CI : (0.9109, 0.915)
    No Information Rate : 0.8787
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5733

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9573
            Specificity : 0.5914
         Pos Pred Value : 0.9444
         Neg Pred Value : 0.6566
             Prevalence : 0.8787
         Detection Rate : 0.8412
   Detection Prevalence : 0.8908
      Balanced Accuracy : 0.7744

       'Positive' Class : stable
```

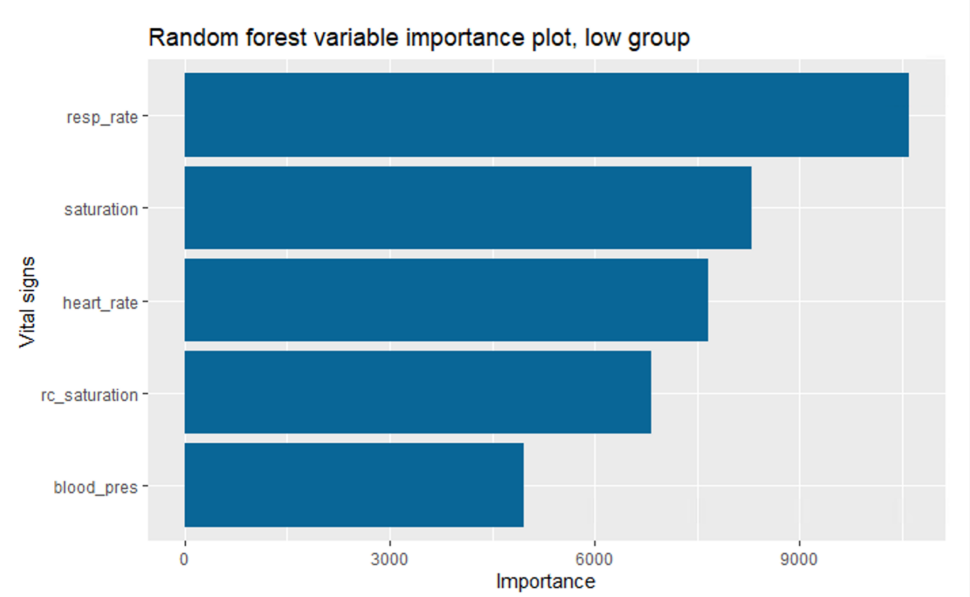
Figure B.8: Performance metrics, 90/10 split

C. Final model interpretation visualisations

This appendix visualises the variable importance and partial dependence plots of each model to create better insights in the models which improves interpretability.

Random forest

This section contains the variable importance plots (VIP) and partial dependence plots (PDP) for the random forest models.



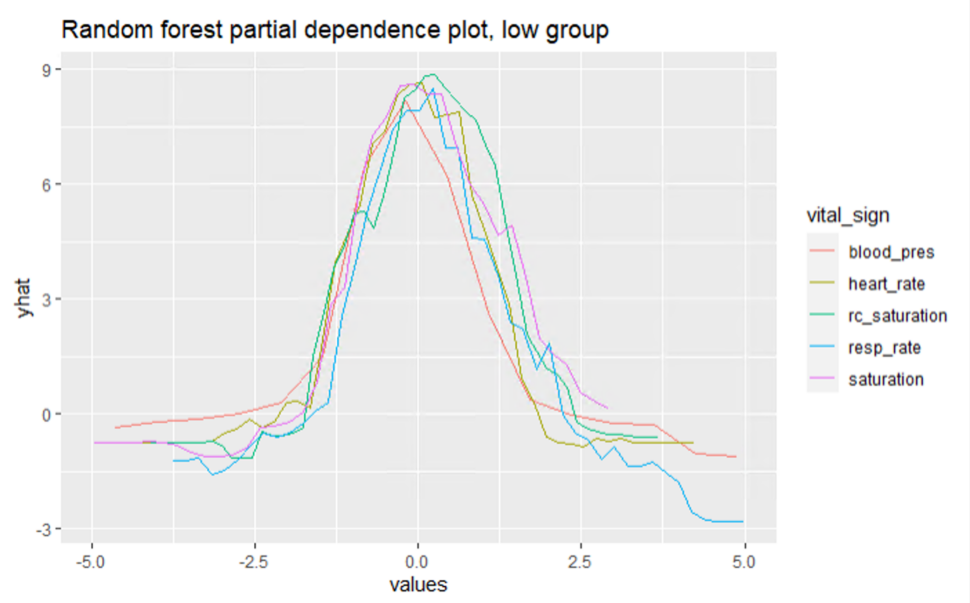
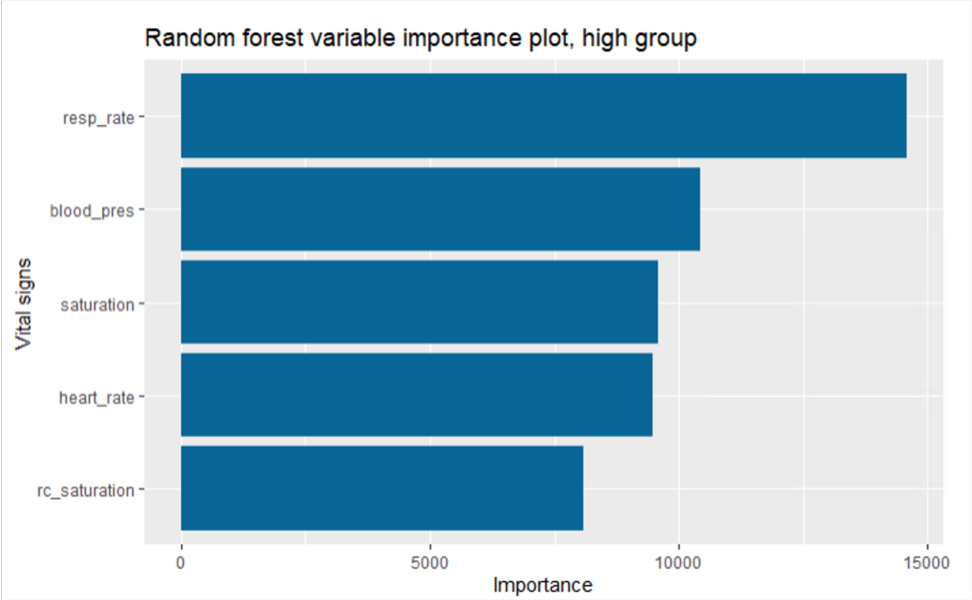


Figure C.1: VIP and PDP, low group



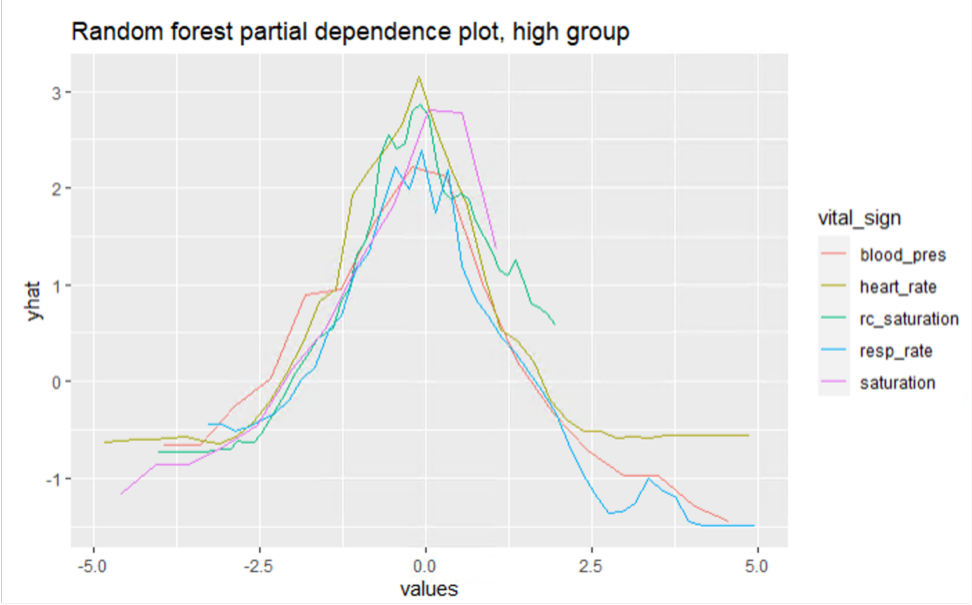
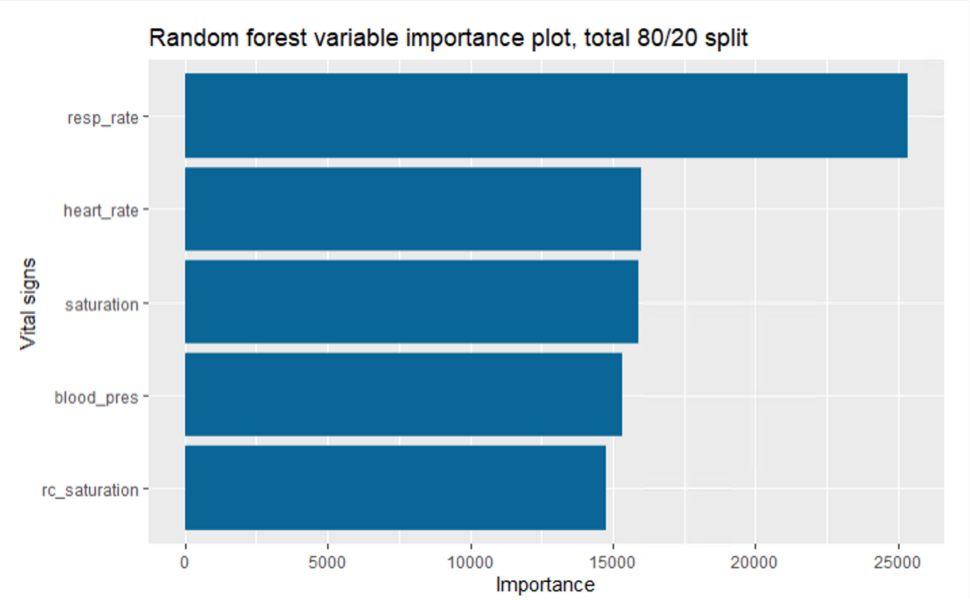


Figure C.2: VIP and PDP, high group



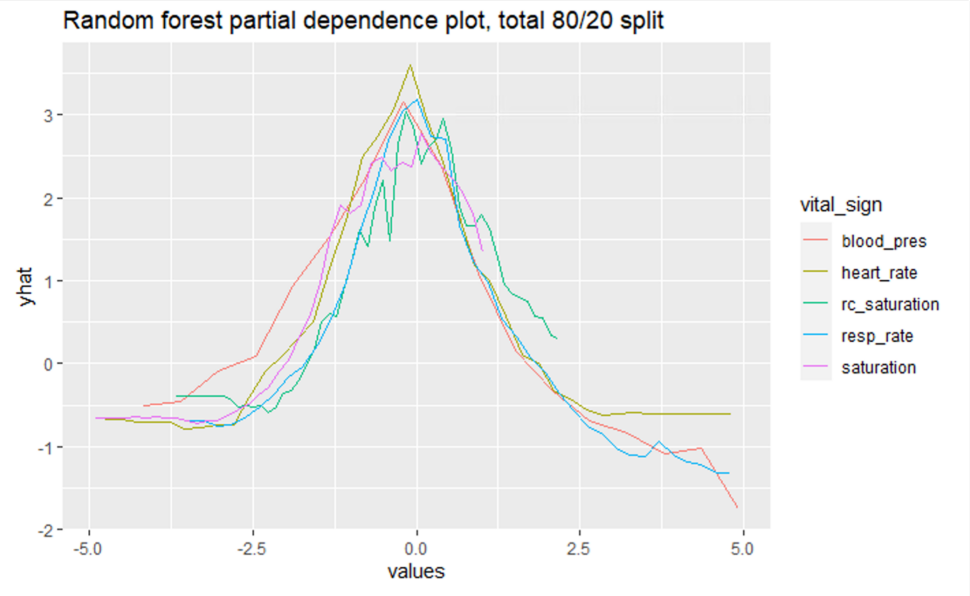
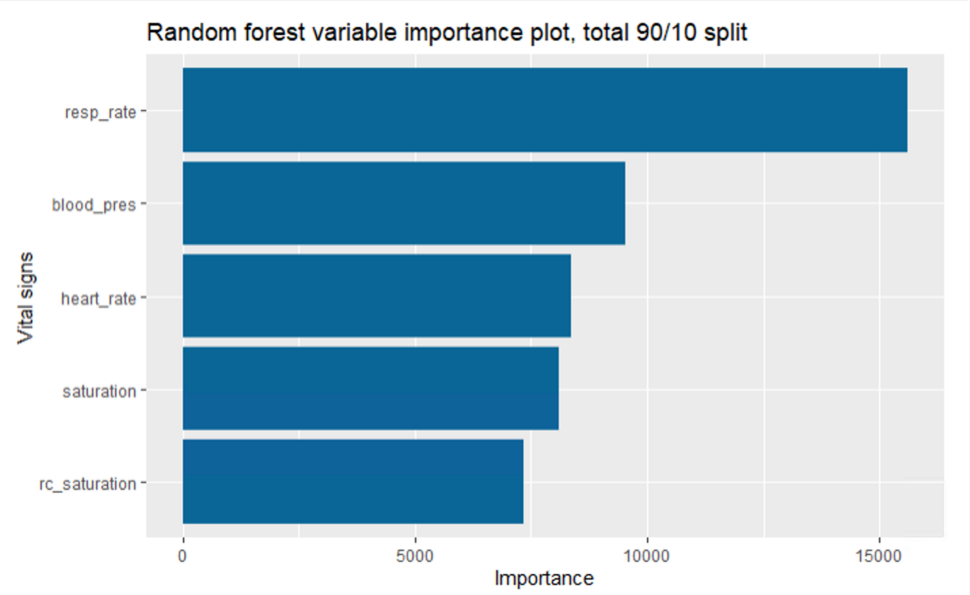


Figure C.3: VIP and PDP, total 80/20 split



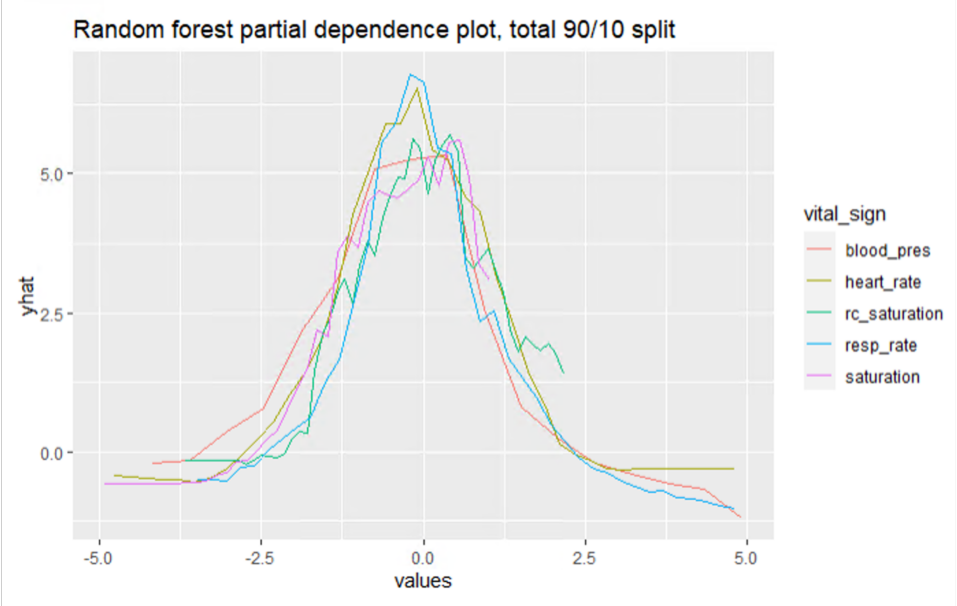
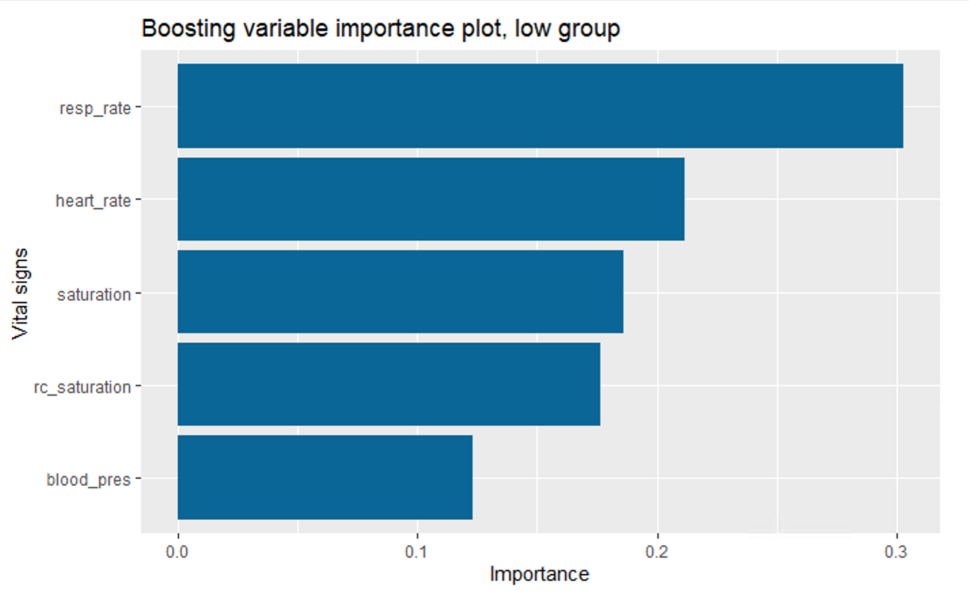


Figure C.4: VIP and PDP, total 90/10 split

Boosting

This section contains the variable importance plots (VIP) and partial dependence plots (PDP) for the boosting models.



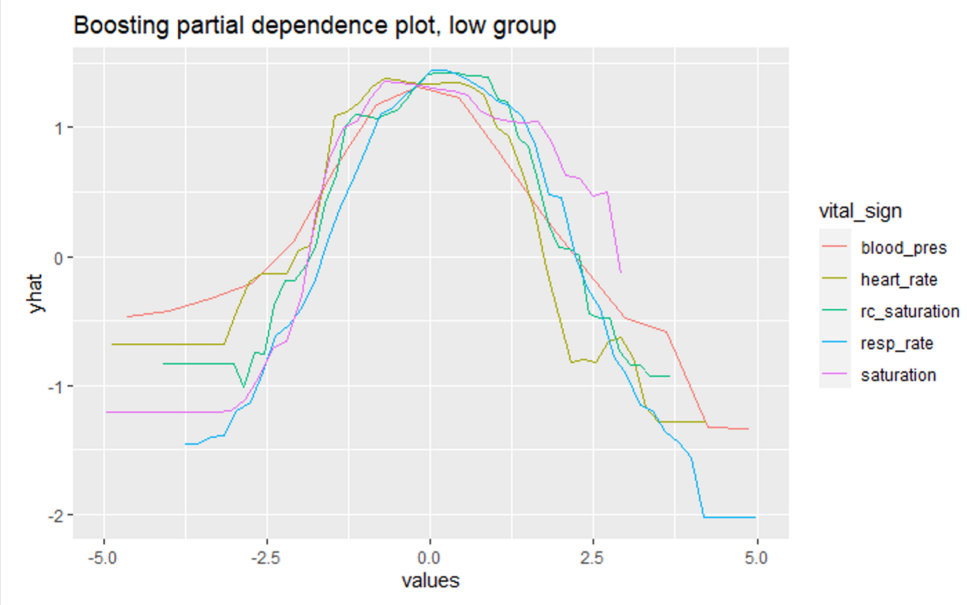
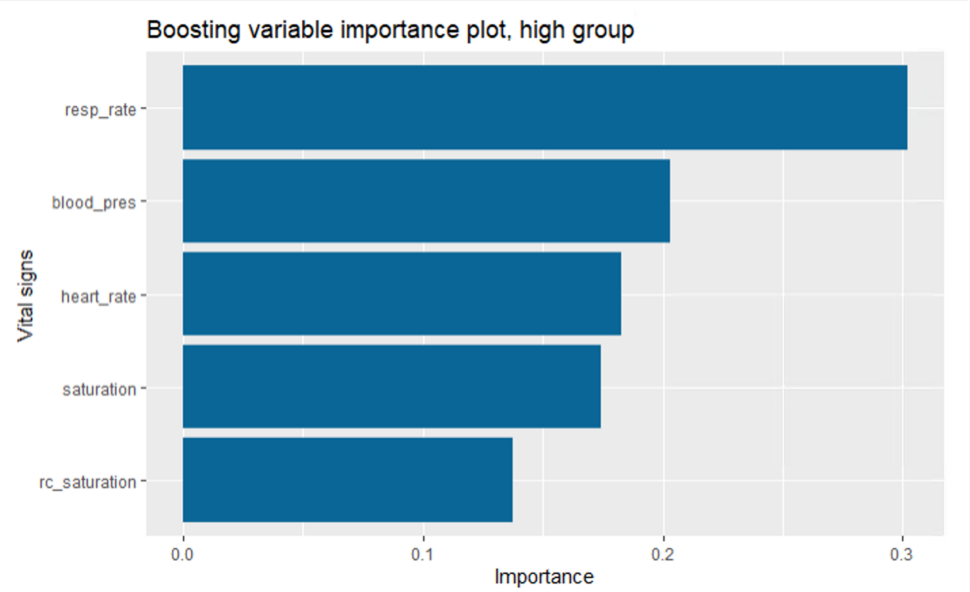


Figure C.5: VIP and PDP, low group



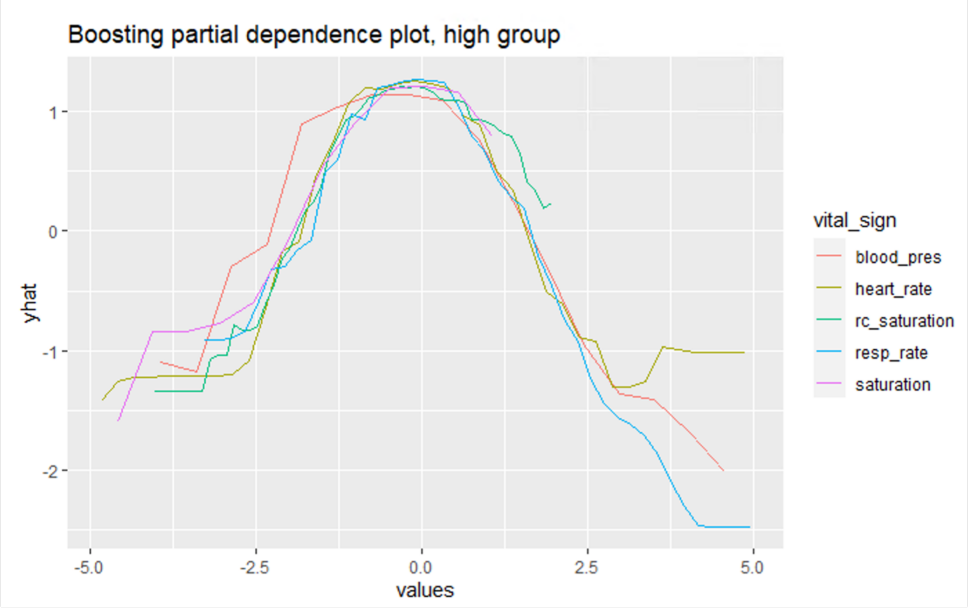
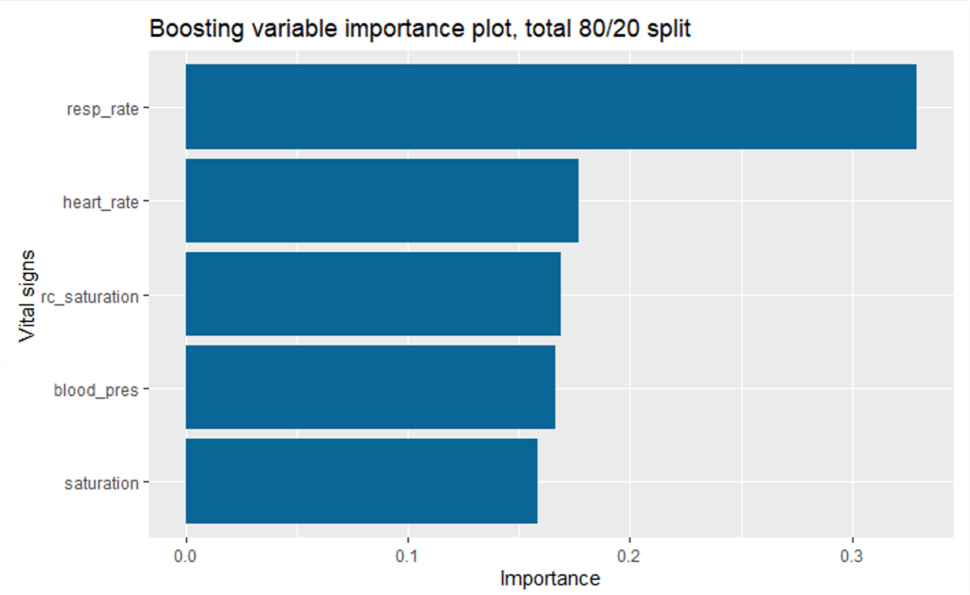


Figure C.6: VIP and PDP, high group



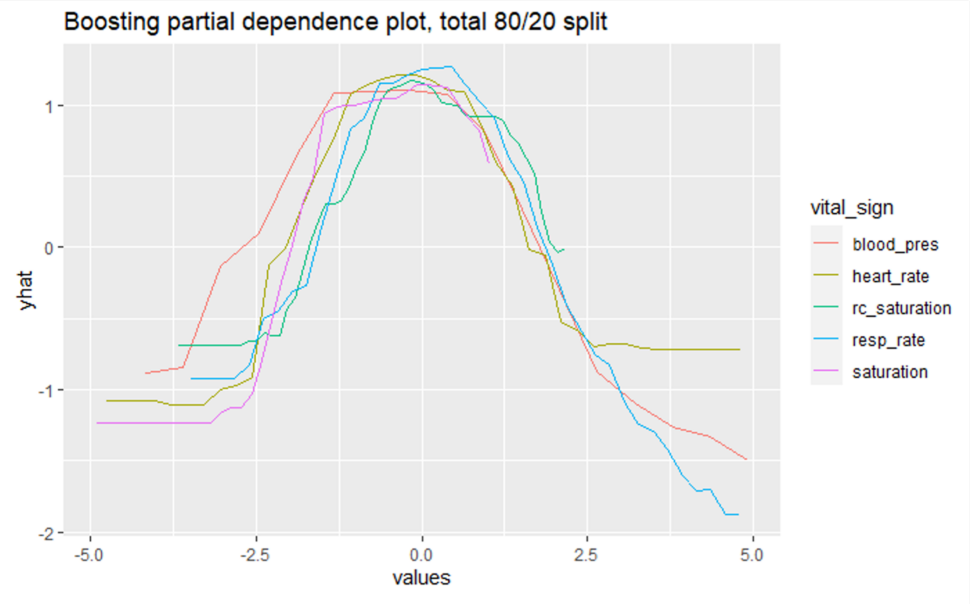
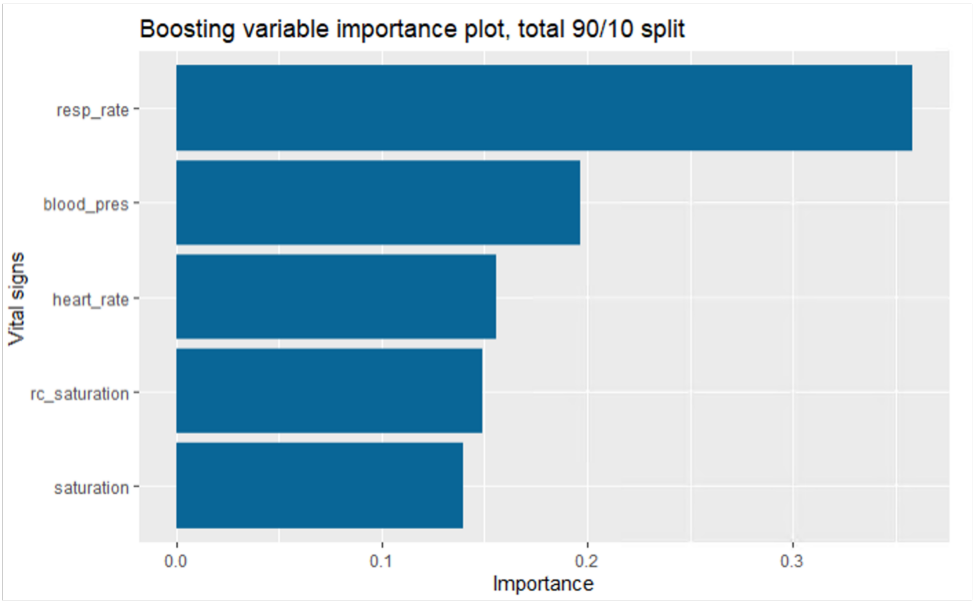


Figure C.7: VIP and PDP, total 80/20 split



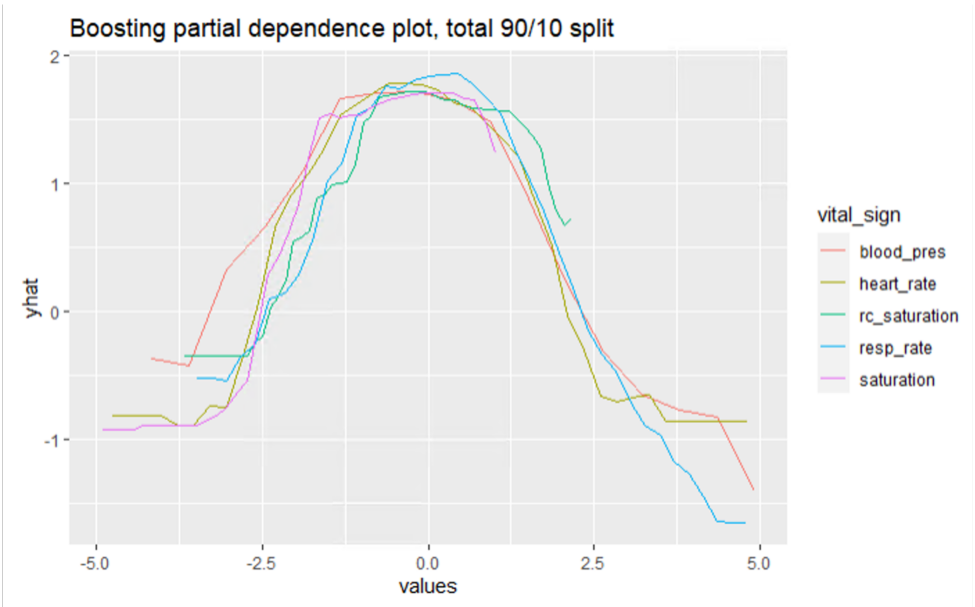


Figure C.8: VIP and PDP, total 90/10 split

D. Clinical performance

This appendix visualises the confusion matrices and their performance metrics for the clinical performance.

Total models

This section contains the confusion matrices and their performance metrics for the models trained and tested on all patients.

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     8007      710
  unstable   2337     1906

               Accuracy : 0.7649
                 95% CI : (0.7575, 0.7722)
    No Information Rate : 0.7981
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4079

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7741
            Specificity : 0.7286
         Pos Pred Value : 0.9185
         Neg Pred Value : 0.4492
             Prevalence : 0.7981
         Detection Rate : 0.6178
   Detection Prevalence : 0.6726
      Balanced Accuracy : 0.7513

       'Positive' Class : stable

       F1
0.8401448
```

Figure D.1: Performance metrics, random forest 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     9808     1354
  unstable    536     1262

               Accuracy : 0.8542
                 95% CI : (0.848, 0.8602)
    No Information Rate : 0.7981
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4875

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9482
            Specificity : 0.4824
         Pos Pred Value : 0.8787
         Neg Pred Value : 0.7019
             Prevalence : 0.7981
         Detection Rate : 0.7568
   Detection Prevalence : 0.8613
      Balanced Accuracy : 0.7153

       'Positive' Class : stable

        F1
0.9121175
```

Figure D.2: Performance metrics, random forest 90/10

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     8125      951
  unstable   2219     1665

               Accuracy : 0.7554
                 95% CI : (0.7479, 0.7628)
    No Information Rate : 0.7981
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3573

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7855
            Specificity : 0.6365
         Pos Pred Value : 0.8952
         Neg Pred Value : 0.4287
             Prevalence : 0.7981
         Detection Rate : 0.6269
   Detection Prevalence : 0.7003
      Balanced Accuracy : 0.7110

       'Positive' Class : stable

        F1
0.8367662
```

Figure D.3: Performance metrics, boosting 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     9926     1532
  unstable    418     1084

               Accuracy : 0.8495
                 95% CI : (0.8433, 0.8557)
    No Information Rate : 0.7981
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4447

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9596
            Specificity : 0.4144
         Pos Pred Value : 0.8663
         Neg Pred Value : 0.7217
             Prevalence : 0.7981
         Detection Rate : 0.7659
   Detection Prevalence : 0.8841
      Balanced Accuracy : 0.6870

       'Positive' Class : stable

       F1
0.9105587
```

Figure D.4: Performance metrics, boosting 90/10

Low group models

This section contains the confusion matrices and their performance metrics for the low group models. The first two models are trained and tested on low group patients where the last four are trained on all patients, but tested on low group patients.

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     4676      663
  unstable   2409      892

               Accuracy : 0.6444
                 95% CI : (0.6342, 0.6545)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1624

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6600
            Specificity : 0.5736
         Pos Pred Value : 0.8758
         Neg Pred Value : 0.2702
             Prevalence : 0.8200
         Detection Rate : 0.5412
   Detection Prevalence : 0.6179
      Balanced Accuracy : 0.6168

       'Positive' Class : stable

       F1
0.7527366
```

Figure D.5: Performance metrics, random forest

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     5862      665
  unstable   1223      890

               Accuracy : 0.7815
                 95% CI : (0.7726, 0.7902)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3506

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8274
            Specificity : 0.5723
         Pos Pred Value : 0.8981
         Neg Pred Value : 0.4212
             Prevalence : 0.8200
         Detection Rate : 0.6785
   Detection Prevalence : 0.7554
      Balanced Accuracy : 0.6999

       'Positive' Class : stable

        F1
0.8612989
```

Figure D.6: Performance metrics, boosting

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     5824      650
  unstable   1261      905

               Accuracy : 0.7788
                 95% CI : (0.7699, 0.7875)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3503

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8220
            Specificity : 0.5820
         Pos Pred Value : 0.8996
         Neg Pred Value : 0.4178
             Prevalence : 0.8200
         Detection Rate : 0.6741
   Detection Prevalence : 0.7493
      Balanced Accuracy : 0.7020

       'Positive' Class : stable

        F1
0.8590604
```

Figure D.7: Performance metrics, random forest 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     6874     1024
  unstable    211      531

               Accuracy : 0.8571
                 95% CI : (0.8495, 0.8644)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3916

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9702
            Specificity : 0.3415
         Pos Pred Value : 0.8703
         Neg Pred Value : 0.7156
             Prevalence : 0.8200
         Detection Rate : 0.7956
   Detection Prevalence : 0.9141
      Balanced Accuracy : 0.6558

       'Positive' Class : stable

       F1
0.9175732
```

Figure D.8: Performance metrics, random forest 90/10

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     5795      867
  unstable   1290      688

               Accuracy : 0.7503
                 95% CI : (0.7411, 0.7594)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2354

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8179
            Specificity : 0.4424
         Pos Pred Value : 0.8699
         Neg Pred Value : 0.3478
             Prevalence : 0.8200
         Detection Rate : 0.6707
   Detection Prevalence : 0.7711
      Balanced Accuracy : 0.6302

       'Positive' Class : stable

      F1
0.843093
```

Figure D.9: Performance metrics, boosting 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     6890     1196
  unstable    195      359

               Accuracy : 0.839
                 95% CI : (0.8311, 0.8467)
    No Information Rate : 0.82
    P-Value [Acc > NIR] : 1.73e-06

                  Kappa : 0.2716

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9725
            Specificity : 0.2309
         Pos Pred Value : 0.8521
         Neg Pred Value : 0.6480
             Prevalence : 0.8200
         Detection Rate : 0.7975
   Detection Prevalence : 0.9359
      Balanced Accuracy : 0.6017

       'Positive' Class : stable

        F1
0.9083119
```

Figure D.10: Performance metrics, boosting 90/10

High group models

This section contains the confusion matrices and their performance metrics for the high group models. The first two models are trained and tested on high group patients where the last four are trained on all patients, but tested on high group patients.

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     2317      134
  unstable    942      927

               Accuracy : 0.7509
                 95% CI : (0.7377, 0.7638)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : 0.7088

                  Kappa : 0.4652

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7110
            Specificity : 0.8737
         Pos Pred Value : 0.9453
         Neg Pred Value : 0.4960
             Prevalence : 0.7544
         Detection Rate : 0.5363
   Detection Prevalence : 0.5674
      Balanced Accuracy : 0.7923

       'Positive' Class : stable

        F1
0.8115587
```

Figure D.11: Performance metrics, random forest

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     2525      125
  unstable    734      936

               Accuracy : 0.8012
                 95% CI : (0.7889, 0.813)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : 1.574e-13

                  Kappa : 0.5504

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7748
            Specificity : 0.8822
         Pos Pred Value : 0.9528
         Neg Pred Value : 0.5605
             Prevalence : 0.7544
         Detection Rate : 0.5845
   Detection Prevalence : 0.6134
      Balanced Accuracy : 0.8285

       'Positive' Class : stable

        F1
0.8546285
```

Figure D.12: Performance metrics, boosting

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     2183       60
  unstable   1076     1001

               Accuracy : 0.737
                 95% CI : (0.7236, 0.7501)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : 0.996

                  Kappa : 0.4636

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6698
            Specificity : 0.9434
         Pos Pred Value : 0.9733
         Neg Pred Value : 0.4819
             Prevalence : 0.7544
         Detection Rate : 0.5053
   Detection Prevalence : 0.5192
      Balanced Accuracy : 0.8066

       'Positive' Class : stable

       F1
0.7935296
```

Figure D.13: Performance metrics, random forest 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     2934      330
  unstable    325      731

               Accuracy : 0.8484
                 95% CI : (0.8373, 0.859)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5902

 Mcnemar's Test P-Value : 0.8758

            Sensitivity : 0.9003
            Specificity : 0.6890
         Pos Pred Value : 0.8989
         Neg Pred Value : 0.6922
             Prevalence : 0.7544
         Detection Rate : 0.6792
   Detection Prevalence : 0.7556
      Balanced Accuracy : 0.7946

       'Positive' Class : stable

       F1
0.8995861
```

Figure D.14: Performance metrics, random forest 90/10

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     2330       84
  unstable    929      977

               Accuracy : 0.7655
                 95% CI : (0.7526, 0.7781)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : 0.04604

                  Kappa : 0.5012

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.7149
            Specificity : 0.9208
         Pos Pred Value : 0.9652
         Neg Pred Value : 0.5126
             Prevalence : 0.7544
         Detection Rate : 0.5394
   Detection Prevalence : 0.5588
      Balanced Accuracy : 0.8179

       'Positive' Class : stable

        F1
0.8214349
```

Figure D.15: Performance metrics, boosting 80/20

```
Confusion Matrix and Statistics

          Reference
Prediction stable unstable
  stable     3036      336
  unstable    223      725

               Accuracy : 0.8706
                 95% CI : (0.8602, 0.8805)
    No Information Rate : 0.7544
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6378

 Mcnemar's Test P-Value : 2.168e-06

            Sensitivity : 0.9316
            Specificity : 0.6833
         Pos Pred Value : 0.9004
         Neg Pred Value : 0.7648
             Prevalence : 0.7544
         Detection Rate : 0.7028
   Detection Prevalence : 0.7806
      Balanced Accuracy : 0.8074

       'Positive' Class : stable

      F1
0.915699
```

Figure D.16: Performance metrics, boosting 90/10